Representing Beliefs in the Fluent Calculus

Yi Jin and **Michael Thielscher** [1]

Abstract. Action formalisms like the fluent calculus have been developed to endow logic-based agents with the abilities to reason about the effects of actions, to execute high-level strategies, and to plan. In this paper we extend the fluent calculus by a method for belief change, which allows agents to revise their internal model upon making observations that contradict this model. Unlike the existing combination of the situation calculus with belief revision [16], our formalism satisfies all of the standard postulates for (iterated) belief change. Furthermore, we have extended the high-level action programming language FLUX by a computational approach to belief change which is provably equivalent to the axiomatic characterization in the fluent calculus.

1 INTRODUCTION

Logic-based agents and robots reason about actions for many purposes: to verify the executability of actions, to execute complex strategies, to plan ahead, etc. A variety of versatile theories of actions exist, among which are the situation calculus [11, 13] or the fluent calculus [18], which have recently evolved into the high-level, logic-based agent programming languages and systems GOLOG [10, 14] and FLUX [20], respectively. An important extension of basic action theories allows agents and robots to reason about their (incomplete) knowledge and knowledge-producing actions (i.e., sensing), e.g., [7, 2, 19, 15]. A crucial limitation of these approaches, however, is that they all assume agents and robots to have infallible knowledge. A sensing action can never result in an observation which contradicts the current world model, or else the whole theory collapses into an inconsistency. This does not allow for mistakes in the world model, e.g., due to unexpected changes in the environment. Under such circumstances agents should have (more or less strong) *beliefs* rather than (strict) knowledge.

A mostly independent branch of AI research is concerned with just these beliefs and how to revise them in the light of new, possibly conflicting information. While formalisms for belief revision tell agents how to adjust their beliefs given an observation, they do not deal with issues such as reasoning about preconditions of actions, high-level agent programming, or planning.

A first combination of belief change with reasoning about actions has been given in [16] as an extension of the situation calculus. The basic idea was to *rank* the set of possible situations. The agent believes what holds in all situations which are preferred according to the ranking. When a new observation contradicts the current beliefs, the preferred situations are rendered impossible, and so other, still possible situations can now become most preferred. This revision technique may nonetheless lead to an inconsistent belief state, namely, when there are no possible situations left: While an agent could initially believe in some property and then revise this belief, it cannot happen that later on it makes an observation which suggests that the property is true after all. From the perspective of belief revision, this violates the fundamental postulate which says that consistency is maintained unless the new information is self-contradictory [1]. The integration of belief change into GOLOG is also not considered in [16].

In this paper, we integrate belief change into the fluent calculus in a way which overcomes the limitations of [16]. Our axiomatization is based on rankings of possible *states*. Generalizing the concept of knowledge update axioms [19], the effects of actions are specified as a modification of the ranking. The extended action theory is justified in that it satisfies all of the standard postulates for (iterated) belief revision. Furthermore, we have extended FLUX by the concept of entrenchment bases [21] to encode belief states. Revision is then realized as a rewriting operation on these bases. It is shown that this computational approach is equivalent to the axiomatic approach taken in the fluent calculus.

2 FLUENT CALCULUS

The fluent calculus shares with the classical situation calculus [11] the basic notion of a *situation*. The initial situation is usually denoted by the constant S_0. The function $Do(a, s)$ denotes the situation which is reached by performing action a in situation s. In order to specify what holds in a situation, the expression $Holds(f, s)$ is used, where f is a *fluent* (i.e., term of sort FLUENT).

Throughout this paper, we will use the following simple scenario, which has been adopted from [16]: A robot can be in either of two rooms, and there is a light in each room which can be on or off. Let the fluent InR_1 denote that the robot is in room 1, while the fluents $Light_1$ and $Light_2$ shall denote whether the light is on in the respective room. The following axiom, e.g., says that initially the robot is not in room 1 (hence in room 2) and light is on in the first room:

$$\neg Holds(InR_1, S_0) \wedge Holds(Light_1, S_0) \tag{1}$$

The fluent calculus extends the situation calculus by the notion of a *state*. The term $State(s)$ denotes the state (of the environment of an agent) in situation s. By definition, every FLUENT term is a state (i.e., term of sort STATE), and if z_1 and z_2 are states then so is $z_1 \circ z_2$, where "$\circ$" is a binary function written in infix notation. The *foundational axioms* of the fluent calculus stipulate that this function shares essential properties with the union operation for sets (see, e.g., [20] for details). This allows to define the expression *Holds* as a mere macro by $Holds(f, s) \stackrel{\text{def}}{=} Holds(f, State(s))$ and $Holds(f, z) \stackrel{\text{def}}{=} (\exists z')\, z = f \circ z'$. With this, specification (1) entails the following equation for $State(S_0)$:

$$(\exists z)\,(State(S_0) = Light_1 \circ z \wedge \neg Holds(InR_1, z)) \tag{2}$$

[1] Dresden University of Technology, Dresden (Germany)
email: {yijin,mit}@inf.tu-dresden.de
The first author is funded by the *Deutsche Forschungsgemeinschaft* under grant no. Gr 334/3.

Based on the notion of a state, the frame problem is solved in the fluent calculus by *state update axioms*, which define the effects of an action A in situation s in terms of the difference between $State(s)$ and the successor $State(Do(A, s))$. Consider, for example, the action *Leave* of our robot to leave the current room and enter the adjacent one. This action has a conditional effect: If the robot starts in room 1, then it will no longer be there after the action. Conversely, if the robot starts in the other room, then it will end up in 1. This is expressed by the following state update axiom:

$$\begin{array}{l} Poss(Leave, s) \supset \\ \quad Holds(InR_1, s) \wedge State(Do(Leave, s)) = State(s) - InR_1 \\ \quad \vee \neg Holds(InR_1, s) \wedge State(Do(Leave, s)) = State(s) + InR_1 \end{array} \quad (3)$$

The standard predicate $Poss(a, s)$ means that action a is possible in situation s. The functions "$-$" and "$+$" denote, respectively, removal and addition of fluents to states. They have a purely axiomatic characterization in the fluent calculus (we again refer to [20] for details). For example, tacitly assuming $Poss(Leave, S_0)$ and uniqueness-of-names for the fluents InR_1 and $Light_1$, the instance $\{s/S_0\}$ of state update axiom (3) applied to equation (2) yields, with the help of the foundational axioms, $(\exists z)\, State(Do(Leave, S_0)) = InR_1 \circ Light_1 \circ z$.

Representing State Knowledge

The knowledge that an agent has of its environment can be represented in the fluent calculus via the notion of *possible states*. The predicate $KState(s, z)$ has been introduced in [19] with the intended meaning that, according to the knowledge of the agent, z is a possible state in situation s. The following axiom, for example, says implicitly that in the initial situation the robot knows that it is in room 1 and that the light in room 1 is off, but it does not know whether light is on in room 2:

$$(\forall z)\,(KState(S_0, z) \equiv Holds(InR_1, z) \wedge \neg Holds(Light_1, z)) \quad (4)$$

Formally, a property is defined to be known in a situation just in case it holds in all possible states:

$$Knows(\varphi, s) \stackrel{\text{def}}{=} (\forall z)(KState(s, z) \supset HOLDS(\varphi, z))$$

Here, φ is a *knowledge expression*, which is composed of fluents and the standard logical connectives. The macro $HOLDS(\varphi, z)$ stands for the fluent calculus formula which is obtained by replacing, in φ, every occurrence of a fluent f by $Holds(f, z)$. For example, (4) entails $Knows(InR_1 \wedge \neg Light_1, S_0)$ but not $Knows(Light_2, S_0) \vee Knows(\neg Light_2, S_0)$.

The effects of actions, including knowledge-producing actions, on the knowledge of an agent are specified by *knowledge update axioms*. These relate the possible states between successive situations. Consider, e.g., the action $Sense\_InR_1$ of our robot to sense whether it is in room 1:

$$\begin{array}{r} Poss(Sense\_InR_1, s) \supset (KState(Do(Sense\_InR_1, s), z) \equiv \\ KState(s, z) \wedge [Holds(InR_1, z) \equiv Holds(InR_1, s)]\,) \end{array} \quad (5)$$

Put in words, a state z is possible after $Sense\_InR_1$ just in case z was possible beforehand and InR_1 holds in z iff it actually holds in s.

The fluent calculus provides the formal underpinnings of FLUX, which is a method based on logic programming for the design of agents that reason about their actions and sensor information in the presence of incomplete knowledge [20]. Knowledge states of agents are represented in FLUX by *open-ended* lists of fluents along with *constraints*, as, e.g., in this encoding of (2):

```
Z0 = [light1|Z], not_holds(inR1,Z)
```

Agent programs in FLUX are constraint logic programs consisting of three components $P_{kernel} \cup P_{domain} \cup P_{strategy}$ providing, respectively, a domain-independent encoding of the foundational axioms and macros of the fluent calculus, an encoding of the domain axioms, and a specification of the task-oriented behavior of the agent, according to which it reasons, plans, and acts.

3 A FLUENT CALCULUS AXIOMATIZATION OF BELIEF CHANGE

The concept of knowledge in the fluent calculus presupposes that new information must be consistent with what is previously known. Recall, for example, axiom (4), which entails $Knows(InR_1, S_0)$. Tacitly assuming $Poss(Sense\_InR_1, S_0)$, the knowledge update axiom for $Sense\_InR_1$, (5), implies that if the robot were to sense that it is actually not in room 1 (i.e., $\neg Holds(InR_1, S_0)$), then no state z would satisfy $KState(Do(Sense\_InR_1, S_0), z)$. In other words, the robot would be left with an inconsistent knowledge state. In this section, we develop an axiomatic approach to the representation of belief rather than irrefutable knowledge in the fluent calculus.

3.1 State Ranking

To begin with, the underlying signature is modified by replacing predicate *KState* by the function

$$BState : \text{SIT} \times \text{STATE} \mapsto \mathbb{N}$$

Our intention is to represent the belief state of an agent in a situation by a *ranking* of states. Intuitively, a high value of $BState(s, z)$ indicates that state z is considered less plausible (in violating a strongly held belief) in situation s. The most plausible states are therefore those of rank 0, and the agent is said to *believe* a property φ just in case φ holds in all 0-states:

$$Believes(\varphi, s) \stackrel{\text{def}}{=} (\forall z)\,(BState(s, z) = 0 \supset HOLDS(\varphi, z))$$

For later purposes, we define a macro which can be used to determine the maximal rank of a state in situation s: $MaxRank(s) = n \stackrel{\text{def}}{=} (\exists z)\, BState(s, z) = n \;\wedge\; (\forall z)\, BState(s, z) \leq n$.[2] Consider, for example, the following specification of an initial belief state:

$$\begin{array}{l} BState(S_0, z) = 0 \equiv Holds(InR_1, z) \wedge Holds(Light_1, z) \\ BState(S_0, z) = 1 \equiv Holds(InR_1, z) \wedge \neg Holds(Light_1, z) \\ BState(S_0, z) = 2 \equiv \neg Holds(InR_1, z) \end{array} \quad (6)$$

Put in words, the robot believes that it is in room 1 and that light is on there. The belief in the former is stronger. This is indicated by the fact that every state which violates $Holds(InR_1, z)$ is of rank 2, while there are states for which $Holds(Light_1, z)$ is false whose rank is just 1. Axioms (6) entail, for example, $Believes(InR_1 \wedge Light_1, S_0)$ but not $Believes(Light_2, S_0) \vee Believes(\neg Light_2, S_0)$.

Given a specification of a belief state, agents and robots can use the standard features of the fluent calculus to reason about whether an action can be believed to be executable, whether they believe that a goal condition has been satisfied, etc.

3.2 Belief Change Axioms

Next, we introduce the central notion of *belief change axioms* as a means to specify the effects of actions on the belief state of an agent. We distinguish between sensing actions and actions with physical effects. In general, sensing actions require agents to revise their beliefs according to the newly acquired information.

[2] The use of this macro stipulates that all situations have a maximal rank.

Belief Revision in the Fluent Calculus

The axiomatization of belief change relies on the notion of the *rank* of a knowledge expression φ in a situation s. Intuitively, the *higher* the rank the stronger the belief in φ. Formally,

$$\begin{aligned} Rank(\varphi, s) = n \stackrel{\text{def}}{=} & (\forall z)\,(BState(s,z) < n \supset HOLDS(\varphi, z)) \\ & \wedge \\ & [(\exists z)\,(BState(s,z) = n \wedge \neg HOLDS(\varphi, z)) \\ & \vee (\forall z)\, HOLDS(\varphi, z) \wedge MaxRank(s) = n-1] \end{aligned}$$

Put in words, the rank of a formula is the lowest rank of a state which violates this formula. If no such state exists (that is, φ is a tautology), then the rank of φ is defined as the maximal state rank plus one. It is easy to verify that a formula has rank 0 iff it is not believed in s. For instance, belief state specification (6) from above entails that $Rank(InR_1) = 2$ and $Rank(InR_1 \wedge Light_1, S_0) = 1$ while $Rank(Light_2, S_0) = Rank(\neg Light_2, S_0) = 0$.

We are now prepared to define belief change axioms for sensing actions. From Spohn's ordinal conditional function [17], we have learned that a general theory of belief revision requires to supply a reliability degree to the new formula.[3] Let $Sense\_\varphi$ denote the action by which an agent learns with reliability e (of sort $\mathbb{N}$) whether or not knowledge expression φ holds. Prior to the formal definition, let us give an informal justification for the axiomatization of the effect of $Sense\_\varphi$: Suppose m is the rank of some state z in situation s, in which sensing takes place. Assume, for the sake of argument, that φ has been sensed to be true.

1. If $\neg HOLDS(\varphi, z)$, then z must be "upgraded" wrt. the reliability of the sensing, because it violates what has just been sensed. The revised rank of z is therefore $m + e$.
2. If $HOLDS(\varphi, z)$, then z must be "downgraded," because z complies with what has just been sensed. The revised rank of z is $m - Rank(\neg\varphi, s)$. Note that this ensures that the agent gives up a possible belief in $\neg\varphi$.

This intuition is axiomatized as follows:

$$\begin{aligned} & BState(Do(Sense\_\varphi, s), z) = n \equiv \\ & \quad (\exists m, r)\,(BState(s,z) = m \wedge Rank(\neg\varphi, s) = r \wedge \\ & \qquad [\neg HOLDS(\varphi, z) \supset n = m + e] \wedge \\ & \qquad [HOLDS(\varphi, z) \supset n = m - Rank(\neg\varphi, s)]) \end{aligned} \qquad (R_\varphi)$$

Let $R_{\neg\varphi}$ be the exact same formula but with φ replaced by $\neg\varphi$ (defining the case where φ is sensed to be false). The two cases are combined in this central definition of the *belief change axiom* for sensing actions:

$$Poss(Sense\_\varphi, s) \supset (\exists e)\,(R_\varphi \vee R_{\neg\varphi}) \qquad (7)$$

Put in words, the effect of sensing is that the agent obtains a reliability degree e and updates its belief state accordingly, depending on whether φ or $\neg\varphi$ has been sensed.

Theorem 1 *The axioms* $(7) \cup \{Poss(Sense\_\varphi, s)\}$ *for all knowledge expressions* φ *together are consistent with the foundational axioms.*

Agents can execute several sensing actions in sequence, which corresponds to *iterated* belief revision [4]. As the main result, it can be shown that our axiomatization in the fluent calculus satisfies the standard postulates.

Theorem 2 *The fluent calculus axiomatization of (iterated) belief revision satisfies the modified AGM postulates of [4] as well as the postulates of iterated belief revision of [4].*

[3] If this information is not available, it can be uniformly set to 1 as in [4].

Recall, for example, belief state specification (6) for our robot. Suppose $Sense\_InR_1$ results in the observation, with reliability 3, that the robot is in fact not in room 1. Then $R_{\neg InR_1}$ entails,

$$\begin{aligned} & BState(S_1, z) = 3 \equiv Holds(InR_1, z) \wedge Holds(Light_1, z) \\ & BState(S_1, z) = 4 \equiv Holds(InR_1, z) \wedge \neg Holds(Light_1, z) \\ & BState(S_1, z) = 0 \equiv \neg Holds(InR_1, z) \end{aligned} \qquad (8)$$

where $S_1 = Do(Sense\_InR_1, S_0)$. Hence, the robot now believes that it is not in room 1. Moreover, the belief in $Light_1$ is given up because it was weaker than the belief in InR_1.

Belief Update in the Fluent Calculus

Belief change as a consequence of a non-sensing action A is defined as the usual state update according to the effects of A. Since several states may lead to the same updated state, the rank of an updated state is, in general, defined as the *minimum* of the ranks of all states that map onto it. Moreover, some states may not be reachable at all, in which case their rank is defined as the maximum rank in the preceding situation plus one, thus indicating that they are highly implausible.

The axiomatization of actions with unconditional effects can be proved to satisfy the KM postulates for belief update [9]. Due to lack of space, here we just give the simple example of an action whose effect defines a one-to-one mapping on states. The action *Leave* of our robot (c.f. (3)) gives rise to this belief change axiom:

$$\begin{aligned} & Poss(Leave, s) \supset (\, BState(Do(Leave, s), z) = n \equiv \\ & \quad (\exists z')(BState(s, z') = n \wedge [\, Holds(InR_1, z') \wedge z = z' - InR_1 \\ & \qquad \vee \neg Holds(InR_1, z') \wedge z = z' + InR_1])) \end{aligned}$$

Recall, say, initial belief (6) of our robot and suppose that $Poss(Leave, S_0)$. Let $S_1 = Do(Leave, S_0)$, then

$$\begin{aligned} & BState(S_1, z) = 0 \equiv \neg Holds(InR_1, z) \wedge Holds(Light_1, z) \\ & BState(S_1, z) = 1 \equiv \neg Holds(InR_1, z) \wedge \neg Holds(Light_1, z) \\ & BState(S_1, z) = 2 \equiv Holds(InR_1, z) \end{aligned}$$

Hence, the robot now believes that it is no longer in room 1. Unlike in (8), however, the belief in $Light_1$ is still maintained.

4 BELIEF CHANGE IN FLUX

4.1 Computational Belief Revision

Extending FLUX for belief change requires a computational account of belief revision, which furthermore needs to be equivalent to the axiomatization in the fluent calculus. To this end, we adopt an approach originating in [6]. The basic idea is to consider some beliefs more important than others. We say that these beliefs have higher degree. When a belief change occurs, the agent prefers to give up beliefs with lower degree instead of those with higher degree.

Definition 4.1 A *belief set* is a deductively closed set of formulas. An *epistemic entrenchment (EE) relation* $\leq_K$ wrt. a belief set K is a total pre-order over all formulas, which obeys the postulates given in [6]. If $\alpha \leq_K \beta$, then β is *as epistemically entrenched as* α, and $\alpha <_K \beta$ means $\alpha \leq_K \beta$ and not $\beta \leq_K \alpha$.

Given an EE relation $\leq_K$, the result K^*_α of revising K with a formula α can be uniquely determined by the following condition:

$$(C*) \;\; \beta \in K^*_\alpha \text{ iff either } \models \neg\alpha \text{ or } \neg\alpha <_K \alpha \supset \beta \qquad (9)$$

The EE relation model is constructive in the sense that it uniquely determines belief change operations which satisfy all corresponding AGM postulates [1]. However, as it stands it is not suitable for computation, for two reasons.

1. An EE relation $\leq_K$ in general is infinite.
2. Condition $(C*)$ may have to be checked against infinite number of formulas.

To tackle the first problem, we adopt a model due to Wobcke [21], who has suggested to represent an EE relation by a finite base. The full EE relation can then be induced via the so-called most constructive entrenchment construction.

Definition 4.2 An *epistemic entrenchment base* B is a set $\{F_1 : e_1, \ldots, F_n : e_n\}$, where each F_i is a non-tautologous formula and $e_i \in \mathbb{N}$ is its (explicit) *belief degree*. B^m is the set of formulas of B which have at least belief degree m, that is,

$$B^m = \{F \mid F : e \in B \text{ and } e \geq m\}$$

Note that B^0 is the set of all formulas in B. The corresponding belief set *Bel*(B) of B is the deductive closure of B^0. A EE base is consistent iff its corresponding belief set is consistent. From now on, we only consider consistent EE bases. For any formula φ, its belief degree (also called *rank*) wrt. a given EE base B is defined as follows:

$$\mathit{Rank}(B, \varphi) = \begin{cases} 0 & \text{if } B^0 \not\models \varphi \\ n+1 & \text{if } \models \varphi \\ max(\{m \mid B^m \models \varphi\}) & \text{otherwise} \end{cases}$$

where n is the maximal belief degree in B. Any EE base B induces a binary relation $\leq_B$ over all formulas by letting $\alpha \leq_B \beta$ iff *Rank*$(B, \alpha) \leq$ *Rank*(B, β). The following result is due to [21]:

Theorem 3 *Given an EE base B, the induced binary relation $\leq_B$ is an EE relation wrt. Bel(B), that is, it satisfies all postulates for epistemic entrenchment relations.*

In addition to the remaining second problem, there is another well-known problem: Condition $(C*)$ only tells us what formulas are in the revised belief set. It does not impose any constraints on the posterior EE relation. This means that we lose extra-logical information by carrying out a belief contraction; hence, iterated belief change cannot be handled [4]. Since an EE relation conveys valuable information, we would like to keep as much EE information as possible. On the other hand, the change of the EE base should not be minimal in the sense of [3], in order not to have the undesired properties thereof (see [4]).

Motivated by the application of FLUX to the control of autonomous agents in dynamic environments, we consider the new formula φ and its supplied degree e as additional evidence. Hence, the revised rank of φ is the summation of its old rank and e. We assume that the revising formula is consistent. Algorithm 1 shows how we can do belief revision by modifying an EE base. For iterated belief revision, the algorithm is repeatedly applied.

Input : $B = [\beta_1 : e_1, \ldots, \beta_n : e_n], \varphi, e$
Output : B_1 such that $B_1 = B^*_{\varphi,e}$
begin
  $B_1 = [\,]$;
  $\overline{r} = \mathit{Rank}(B, \neg\varphi)$;
  for $i = 1 \ldots n$ **do**
    $B_1 = B_1 \cup \{\beta_i : e_i - min(e_i, \overline{r}), \beta_i \vee \varphi : e_i + e\}$;
  end
  $B_1 = B_1 \cup \{\varphi : e\}$
end

Algorithm 1: Algorithm of the EE base revision

The resulting EE base $B^*_{\varphi,e}$ may be redundant in the sense that some formula $\alpha : e$ in it has an induced rank which is greater than e. Such redundant formulas can be detected and removed.

The theorem below says that Algorithm 1 indeed defines a rational iterated belief revision operation (for arbitrary e and consistent B, φ).

Lemma 4.3 *Let $B_1 = B^*_{\varphi,e}$ and $\overline{r} = \mathit{Rank}(B, \neg\varphi)$, then for any formula β,*

$$\mathit{Rank}(B_1, \beta) = \begin{cases} t - \overline{r} & \text{if } t' = t \\ min(t' - \overline{r}, t + e) & \text{otherwise} \end{cases}$$

where $t = \mathit{Rank}(B, \beta)$ and $t' = \mathit{Rank}(B, \varphi \supset \beta)$.

Lemma 4.4 *For any formula β, we have $Rank(B^*_{\varphi,e}, \beta) > 0$ iff $\mathit{Rank}(B, \neg\varphi) < \mathit{Rank}(B, \varphi \supset \beta)$.*

Theorem 4 *The belief revision operation on EE bases satisfies all AGM postulates, provided that both the formula being revised and the original EE base alone are consistent.*

Belief Revision in FLUX

The integration of the computational approach to belief revision into FLUX requires a decision procedure for the underlying logical language. For the sake of efficiency, we restrict ourselves to propositional logic and employ an efficient decision procedure called non-clausal Davis-Putnam [12]. An EE base is encoded as `[F1@E1,...,Fn@En]`, where `Ei` $\in \mathbb{N}$. Here is an example of revision (where "`-`" denotes "$\neg$"):

```
?- B = [inR1 @ 2, light1 @ 1],
   revise(B, (-inR1) @ 3, B1).

B  = [inR1 @ 2, light1 @ 1]
B1 = [-(inR1) @ 3, -(inR1) v light1 @ 4]
```

4.2 EQUIVALENCE OF AXIOMATIC AND OPERATIONAL BELIEF REVISION

The definition of how to change an EE base in the presence of new information is essentially equivalent to the axiomatizations of belief revision in the fluent calculus. The formal proof is based on a mapping from EE bases onto axioms of the form *BState*$(s, z) \equiv \Psi(z)$: Let B $= \{F_1 : e_1, \ldots, F_n : e_n\}$ be an EE base, then Ψ defines each state which satisfies all formulas in B to have value 0. Each other state z gets the maximal degree e_i for which formula F_i does not hold in z. For example, the EE base $\{\mathit{InR}_1 : 2, \mathit{Light}_1 : 1\}$ maps onto belief state specification (6).

Due to lack of space we can only state the crucial intermediate results which lead the correctness theorem.

Lemma 4.5 *Let B be an EE base and Σ the fluent calculus axiomatization for belief including the belief state BState$(s, z) \equiv \Psi(z)$ determined from B. For any knowledge expression β and $n \in \mathbb{N}$,*

$$\Sigma \models \mathit{Rank}(\beta, s) = n \;\text{ iff }\; \mathit{Rank}(B, \beta) = n$$

What remains to be shown is that belief update axioms characterize exactly the way an EE relation is changed in FLUX. Since the condition in Lemma 4.3 determines uniquely the revised EE base, it suffices to show that the same condition holds in the fluent calculus.

Lemma 4.6 *Let s be a situation and φ a knowledge expression. Let $s' = Do(Sense\_\varphi, s)$ and $e \in \mathbb{N}$, then (R_φ) entails, for any β,*

$$Rank(\beta, s') = \begin{cases} t - \overline{r} & \text{if } t' = t \\ min(t' - \overline{r}, t + e) & \text{otherwise} \end{cases}$$

where $t = Rank(\beta, s)$, $t' = Rank(\varphi \supset \beta, s)$, and $\overline{r} = Rank(\neg\varphi, s)$.

Theorem 5 *Let B be an EE base, Σ the fluent calculus axiomatization for belief including the belief state $BState(s, z) \equiv \Psi(z)$ determined from B, and φ a knowledge expression being sensed to be true with degree e. For any β and $n \in \mathbb{N}$,*

$$\Sigma \models Rank(\beta, Do(Sense\_\varphi, s)) = n \;\; iff \;\; Rank(B^*_{\varphi,e}, \beta) = n$$

5 DISCUSSION

We have presented an integration of belief change into the fluent calculus. In contrast to the approach of [16], our axiomatization of sensing actions satisfies all standard postulates of (iterated) belief revision. Furthermore, the axiomatization of non-sensing (unconditional) actions satisfies all standard postulates of belief update. The underlying idea for our belief change axiom can be considered a generalization of Spohn's ordinal conditional functions [17]. There, the resulting rank of the revising formula φ is set to the reliability value e,[4] whereas with our belief change axiom φ will obtain the summation of its old rank and e. Another difference is that in general ordinal conditional functions do not satisfy the DP postulates in general. It is worth mentioning that our belief change axiom can be slightly modified (with a provably correct computational account) in such a way that φ obtains the maximum of its old rank and e and such that the AGM and DP postulates are still satisfied. If the reliability value is fixed to 1 (e.g., in cases where such reliability information is not available), then our approach is equivalent to the one proposed in [4]. Actually, the idea of our approach has already been informally hinted at in [4]. So we have not defined a completely new scheme of belief revision and have rather chosen an existing one which turned out to be suitable for integration into the fluent calculus and for extennding FLUX.

In the literature, belief change has been studied in mainly two ways. One approach is to define the class of so-called rational belief change operations and properties they should satisfy (that is, postulates), e.g., [1, 9]. The other approach is to give explicit constructions of belief change operations with desirable properties. Approaches of the latter kind can be further classified as either model-based or computational: In the former, a current belief state is represented using models or deductively closed sets of formulas, as in [6, 8]. This makes it easy to study formal properties of particular belief change operations, but is less suited for direct implementation. In computational approaches to belief change, therefore, a concrete belief state is represented by a finite *base* of formulas, and belief change is defined as rewriting this base, e.g., [5]. We have applied the computational approach to belief revision to develop an extension of FLUX and proved its equivalence to the axiomatic approach. As a by-product we have obtained a model-based characterization of a computational approach to belief change; or, the other way round, we have implemented efficiently a possible world-based approach of belief revision.

While the axiomatization in the fluent calculus allows belief states to be axiomatized using full first-order logic, our current extension of FLUX is restricted to propositional entrenchment bases. Future work will be to lift this restriction to cover the first-order features of the standard FLUX state representation [20].

[4] Hence, φ's prior rank is simply ignored. Furthermore, consecutive observations of φ do not reinforce the belief in φ.

REFERENCES

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *The Journal of Symbolic Logic 50(2)*, 510–530, (1985).

[2] C. Baral and T. C. Son, 'Approximate reasoning about actions in presence of sensing and incomplete information', in *Proc. of the International Logic Programming Symposium (ILPS)*, ed., J. Maluszynski, pp. 387–401, Port Jefferson, NY, (October 1997). MIT Press.

[3] C. Boutilier, 'Revision sequences and nested conditionals', in *Proc. of the Thirteenth International Joint Conference on Artifi cial Intelligence (IJCAI'93)*, pp. 519–525, (1993).

[4] A. Darwiche and J. Pearl, 'On the logic of iterated belief revision', *Artifi cial Intelligence 89*, **1–29**, (1997).

[5] S. E. Dixon, 'Belief revision: A computational approach'. PhD thesis, Basser Department of Computer Science, University of Sydney, Australia, 1994.

[6] P. Gärdenfors and D. Makinson, 'Revisions of knowledge systems using epistemic entrenchment', in *Proc. of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 83–95, Asilomar, CA, (1988).

[7] K. Golden and D. Weld, 'Representing sensing actions: The middle ground revisited', in *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, eds., L. C. Aiello, J. Doyle, and S. Shapiro, pp. 174–185, Cambridge, MA, (November 1996). Morgan Kaufmann.

[8] A. Grove, 'Two modellings for theory change', *Journal of Philosophical Logic, 17*, 157–180, (1988).

[9] H. Katsuno and A. Mendelzon, 'On the difference between updating a knowledge base and revising it', in *KR'91: Principles of Knowledge Representation and Reasoning*, eds., James F. Allen, Richard Fikes, and Erik Sandewall, 387–394, Morgan Kaufmann, San Mateo, California, (1991).

[10] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl, 'GOLOG: A logic programming language for dynamic domains', *Journal of Logic Programming*, **31**(1–3), 59–83, (1997).

[11] J. McCarthy, *Situations and Actions and Causal Laws*, Stanford Artifi cial Intelligence Project, Memo 2, Stanford University, CA, 1963.

[12] J. Otten, 'On the advantage of a non-clausal davis-putnam procedure', Technical Report AIDA-97-1, TH Darmstadt, Intellektik, (1997).

[13] R. Reiter, *Logic in Action*, MIT Press, 2001.

[14] R. Reiter, 'On knowledge-based programming with sensing in the situation calculus', *ACM Transactions on Computational Logic*, **2**(4), 433–457, (2001).

[15] R. Scherl and H. Levesque, 'Knowledge, action, and the frame problem', *Artifi cial Intelligence*, **144**(1), 1–39, (2003).

[16] S. Shapiro, M. Pagnucco, Y. Lespérance, and H. Levesque, 'Iterated belief change in the situation calculus', in *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, eds., A. Cohn, F. Giunchiglia, and B. Selman, pp. 527–538, Breckenridge, CO, (April 2000). Morgan Kaufmann.

[17] W. Spohn, 'Ordinal conditional functions: A dynamic theory of epistemic state', in *Causation in Decision: Belief, Change and Statistics: Proc. of the Irvine Conference on Probability and Causation*, eds., William L. Harper and Brian Skyrms, volume II, pp. 105–134, Dordrecht, (1988). Kluwer Academic Publisher.

[18] M. Thielscher, 'From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem', *Artifi cial Intelligence*, **111**(1–2), 277–299, (1999).

[19] M. Thielscher, 'Representing the knowledge of a robot', in *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, eds., A. Cohn, F. Giunchiglia, and B. Selman, pp. 109–120, Breckenridge, CO, (April 2000). Morgan Kaufmann.

[20] Michael Thielscher, 'FLUX: A logic programming method for reasoning agents', *Theory and Practice of Logic Programming*, (2004). Available for download at `www.fluxagent.org`.

[21] W. R. Wobcke, 'On the use of epistemic entrenchment in nonmonotonic reasoning', in *Proc. of the Tenth European Conference on Artifi cial Intelligence*, pp. 324–328, (1992).