Representing the Knowledge of a Robot

In A. Cohn and F. Giunchiglia and B. Selman, editors, Proc. of the International Conference on Principles of Knowledge Representation and Reasoning, pages 109–120, Breckenridge, CO, 2000. Morgan Kaufmann

Michael Thielscher
Department of Computer Science
Dresden University of Technology
01062 Dresden (Germany)
mit@inf.tu-dresden.de

Abstract

Acquiring information about its environment by sensing is a crucial ability of autonomous robots. Based on the established solution to the Frame Problem of the Fluent Calculus, we present a new, unifying formalism for representing and reasoning about sensing actions, knowledge preconditions, conditional actions, non-knowledge, and about what goals a robot can possibly achieve.

1 INTRODUCTION

Intelligent, autonomous robots choose most of their actions conditioned on the state of their environment. They are equipped with sensors for the purpose of acquiring information about the external world. Robots that perform reasoning about their goals and actions thus need an explicit representation of what they know of a state and how sensing affects their knowledge [Moore, 1985]. For example, the state of a door, that is, whether open or closed, should be known to a robot before it tries to enter the room behind. If the robot does not know then it should plan ahead both the appropriate sensing action and the possibility of having to open the door. This requires the robot to reason about what it currently knows or does not know, and what it will know after sensing.

A formal account of robot knowledge is also needed to prove the ability of a robot to achieve certain goals [Lin and Levesque, 1997]. For example, suppose that the door in question is closed but that its state can be altered by pressing a button next to it. Nonetheless, a robot, call it *Blindie*, who is unable to sense the state of the door will not be able, without assistance, to achieve the goal of entering the room. For it can never know whether it should press the button or not. Likewise unable to achieve the goal will be a robot, call it *Dumbie*, who can see but does not know how the button is causally related to the door. For this robot cannot conclude that it simply must press the button. The latter conclusion shows that a general account of the knowledge of robots must allow for distinguishing between the actual effects of actions and what a robot knows about these effects—an aspect which is not covered by existing accounts of knowledge in action formalisms, such as [Scherl and Levesque, 1993; Lobo *et al.*, 1997; Son and Baral, 1998].

The contribution of the present paper to the research on knowledge in cognitive robotics is manifold:

1. We extend the Fluent Calculus solution to the Frame Problem of [Thielscher, 1999], that is, the concept of state update axioms, to representing and reasoning about the knowledge of a robot.
2. Our theory provides a solution not only to the representational but also the inferential ([Bibel, 1998]) Frame Problem for knowledge.
3. Our theory enables the definition of conditional actions which can be inserted into a plan by a deductive planner.
4. By following a simple axiomatization scheme, our theory can readily be used to reason about what a robot does *not* know; unlike the approach of [Lakemeyer and Levesque, 1998], there is no need for a non-standard semantics or complex second-order axioms to this end.
5. Distinguishing between the actual effects of actions and what a robot knows about them, our theory supports an account of goal achievability *within* the corresponding Fluent Calculus axiomatization, as opposed to the meta-notion of achievability of [Lin and Levesque, 1997].

The key to these achievements lies in the explicit notion of a state offered by the Fluent Calculus aside from the standard notion of a situation. Thus the distinguishing feature of our theory is to formalize the knowledge of a robot in terms of possible *states* rather than possible *situations*.

The organization of the paper is as follows. In the next section, we give an informal overview of our approach to formalizing the knowledge of a robot. In Section 3, we briefly repeat the formal notions and notations of the basic Fluent Calculus. In Section 4, we extend this calculus to the representation of knowledge of states. We formally introduce the concept of knowledge update axioms and prove some crucial properties of our approach. In Section 5, we show how our approach provides a simple and elegant way of reasoning about what a robot does *not* know. In Section 6, we show how our approach can be used to reason about the ability of robots. We summarize in Section 7.

2 KNOWLEDGE IN THE FLUENT CALCULUS: THE BASIC IDEA

The knowledge a robot has of its environment shall be represented using a new binary predicate denoted by $KState(s, z)$. This relation is meant to hold iff in situation s the robot considers z a possible world state.[1] To specify what a robot knows of a particular situation S, an axiom of the form $(\forall z)\,(KState(S, z) \supset \Phi(z))$ is used, where sub-formula Φ constrains z according to the knowledge the robot has about z. Let $Holds(f, z)$ denote that fluent f holds in state z, then an example is,[2]

$$\begin{aligned}&KState(S_0, z) \supset\\ &\quad (\exists x)\,(Door(x, Room411) \wedge \neg Holds(Closed(x), z))\end{aligned}$$

where $Door(x, y)$ is a static (non-fluent) predicate saying that door x leads to room y, and $Closed(x)$ denotes the fluent of door x being closed. Thus the implication says that in all states which are considered possible in situation S_0, there is a door to $Room411$ which is not closed. On this basis, we can say that a robot *knows* that a fluent holds or does not hold in a situation iff the fluent either holds or does not hold in all possible states:[3]

$$\begin{aligned}Knows(f, s) &\stackrel{\text{def}}{=} (\forall z)\,(KState(s, z) \supset Holds(f, z))\\ Knows(\neg f, s) &\stackrel{\text{def}}{=} (\forall z)\,(KState(s, z) \supset \neg Holds(f, z))\end{aligned}$$

These abbreviations generalize to the knowledge of compound formulas in a natural way.

The macro $Knows$ can be used to specify knowledge preconditions for actions. For instance, let $Poss(a, s)$ denote that a is possible in situation s, then according to the following axiom a robot must know of some open door if it wants to enter a room:

$$\begin{aligned}&Poss(Enter(y), s) \equiv\\ &\quad (\exists x)\,(Door(x, y) \wedge Knows(\neg Closed(x), s))\end{aligned}$$

The representational Frame Problem for knowledge is solved by axioms that specify the relation between the possible states before and after an action: The effect of an action $A(\vec{x})$, be it sensing or not, on the knowledge of a robot is specified by a *knowledge update axiom*,

$$\begin{aligned}\Delta \supset (\forall z)\,(&KState(Do(A(\vec{x}), s), z) \equiv\\ &(\exists z')(KState(s, z') \wedge \Psi(\vec{x}, z, z', s)))\end{aligned} \tag{1}$$

where $Do(a, s)$ denotes the situation reached by performing a in s and sub-formula Ψ defines how, under condition Δ, the states considered possible after the action, z, relate to the states considered possible beforehand, z'. Let, for example, $SenseDoor(x)$ denote the action of sensing whether a door x is closed, then this is a suitable knowledge update axiom:[4]

$$\begin{aligned}&Poss(SenseDoor(x), s) \supset\\ &\quad (\forall z)\,(\,KState(Do(SenseDoor(x), s), z) \equiv\\ &\qquad KState(s, z) \wedge\\ &\quad [Holds(Closed(x), z) \equiv Holds(Closed(x), s)])\end{aligned} \tag{2}$$

where $Holds(f, s)$ means that fluent f actually holds in situation s. Hence, the axiom says that among the states possible in s only those are still considered possible after sensing which agree with the actual state of the world as far as the status of the sensed door is concerned. A crucial immediate consequence of (2) in the light of the above definition of $Knows$ is that a

[1] For the reader who is unfamiliar with the Fluent Calculus we note that states are reified, i.e., first-class citizens; see Section 3 for the details.

[2] A word on the notation: Predicate and function symbols, including constants, start with a capital letter whereas variables are in lower case, sometimes with sub- or superscripts. Free variables in formulas are assumed universally quantified. Throughout the paper, action variables are denoted by the letter a, situation variables by the letter s, fluent variables by the letter f, and state variables by the letter z, all possibly with sub- or superscript.

[3] Throughout the paper we assume that the knowledge of a robot—though incomplete—is correct, that is, the actual world state is always among the states which are considered possible. (Foundational axiom (11) in Section 4.2 below expresses this assumption formally.)

[4] The knowledge update axiom below is of the form $\Delta \supset (\forall z)\,(KState(Do(a, s), z) \equiv KState(s, z) \wedge \Psi)$, whose equivalent correct form is $\Delta \supset (\forall z)\,(KState(Do(a, s), z) \equiv (\exists z')\,(KState(s, z') \wedge \Psi \wedge z = z'))$.

robot capable of sensing knows afterwards whether the door is open:

$$
\begin{aligned}
&Poss(SenseDoor(x), s) \supset \\
&\quad Knows(Closed(x), Do(SenseDoor(x), s)) \vee \\
&\quad Knows(\neg Closed(x), Do(SenseDoor(x), s))
\end{aligned}
$$

(This follows since only one of $Holds(Closed(x), s)$ and $\neg Holds(Closed(x), s)$ can be true.)

Providing a solution to the representational aspect of the Frame Problem for knowledge, knowledge update axioms lay the foundations for overcoming the inferential aspect, too. The inference scheme employed to this end takes as input an implication of the form $KState(\sigma, z) \supset \Phi(z)$, which specifies the knowledge of a situation σ, along with the consequence of an aplicable knowledge update axiom of the form (1) for an action $A(\vec{\tau})$. Then an immediate logical consequence of the two input formulas is,

$$
KState(Do(A(\vec{\tau}), \sigma), z) \supset (\exists z')(\Phi(z') \wedge \Psi(\vec{x}, z, z', s))
$$

which provides a full specification of what is known about the successor situation $Do(A(\vec{\tau}), \sigma)$. There is no need to carry over to the new situation all pieces of knowledge one-by-one and using separate instances of axioms, which shows why and how the inferential Frame Problem can be solved on the basis of knowledge update axioms.

In case of non-sensing actions, knowledge update axioms describe what a robot knows of their effect. Examples are given below, in Sections 4 and 6. Since these specifications are independent of state update axioms, which describe the actual effect of actions, it is possible to formally represent and reason about limitations of a robot as regards its knowledge of the dynamics of the environment.

3 THE SIMPLE FLUENT CALCULUS: STATE UPDATE AXIOMS

In the following we provide a brief description of the fundamentals of the Fluent Calculus; for a complete introduction see [Thielscher, 1999] or the electronically available, archived reference article [Thielscher, 1998].[5] The Fluent Calculus, which roots in the logic programming formalism of [Hölldobler and Schneeberger, 1990], is an order-sorted second order language with equality, which includes the sorts *action*, *sit*, *fluent*, and *state*, with *fluent* being a sub-sort of *state*. Fluents are reified propositions. That is to say, terms like, for instance, $Closed(x)$ denote fluents, where $Closed$ is a unary function symbol. Fluents can be joined together by the binary function symbol "$\circ$" to make up states. We write this symbol in infix notation. The function shall satisfy the laws of associativity and commutativity and admit a unit element, denoted by $\emptyset$. Associativity allows us to omit parentheses in nested applications of $\circ$.

A function $State : sit \mapsto state$ relates a situation to the state of the world in that situation. As an example, consider three doors $Door1$, $Door2$, and $Door3$, the first of which is initially closed while the second one is open. Moreover, the robot is currently not in front of $Door1$ or $Door3$ but in front of $Door2$. Let the fluents $Closed(x)$ and $InFrontOf(x)$ denote, resp., that door x is closed and that the robot is in front of door x. Then the given incomplete specification (nothing is said about $Door3$ being open or not) of the initial situation, S_0, can be axiomatized in the Fluent Calculus as follows:

$$
\begin{aligned}
(\exists z)\,[\; State(S_0) = {} & Closed(Door1) \\
& \circ InFrontOf(Door2) \circ z \\
\wedge\ (\forall z')\,(\, & z \neq Closed(Door2) \circ z' \wedge \\
& z \neq InFrontOf(Door1) \circ z' \wedge \\
& z \neq InFrontOf(Door3) \circ z')\,]
\end{aligned}
\tag{3}
$$

That is, of the initial state $State(S_0)$ it is known that both $Closed(Door1)$ and $InFrontOf(Door2)$ are true and that possibly some other fluents z hold except for each of $Closed(Door2)$, $InFrontOf(Door1)$, and $InFrontOf(Door3)$, of which we know they are not true in S_0.

Fundamental for any Fluent Calculus axiomatization is the axiom set $EUNA$, which extends given unique name-assumptions by the axioms AC1 (i.e., associativity, commutativity, and unit element):

$$
\begin{aligned}
(z_1 \circ z_2) \circ z_3 &= z_1 \circ (z_2 \circ z_3) \\
z_1 \circ z_2 &= z_2 \circ z_1 \\
z \circ \emptyset &= z
\end{aligned}
$$

along with the following axioms, which entail inequality of state terms (as used, e.g., in (3)) if some fluent occurs in one but not in the other state:[6]

$$
\begin{aligned}
& z = f \;\supset\; z \neq \emptyset \wedge [\, z = z' \circ z'' \;\supset\; z' = \emptyset \vee z'' = \emptyset \,] \\
& z_1 \circ z_2 = z_3 \circ z_4 \;\supset \\
& \quad (\exists z_a, z_b, z_c, z_d)\,[\, z_1 = z_a \circ z_b \,\wedge\, z_2 = z_c \circ z_d \,\wedge \\
& \qquad\qquad z_3 = z_a \circ z_c \,\wedge\, z_4 = z_b \circ z_d \,]
\end{aligned}
$$

[5] or see the online tutorial at `http://pikas.inf.tu-dresden.de/~mit/FC/Tutorial/index.htm`

[6] Unlike the definition of $EUNA$ in terms of unification completeness wrt. AC1, as used in earlier versions of the Fluent Calculus [Hölldobler and Thielscher, 1995; Thielscher, 1999], the new axioms allow to incorporate domain dependent assumptions of unique names [Störr and Thielscher, 2000].

For computing with state terms, two immediate consequences of these axioms are of importance, the following rules of cancellation and distribution.

Proposition 1 **[Störr and Thielscher, 2000]** *Axioms EUNA entail:*

1. *If* $z_1 \circ f = z_2 \circ f$, *then* $z_1 = z_2$.
2. *If* $f_1 \neq f_2$ *and* $z_1 \circ f_1 = z_2 \circ f_2$, *then* $(\exists z')\, z_2 = f_1 \circ z'$ *and* $(\exists z')\, z_1 = f_2 \circ z'$.

In addition, we have the foundational axiom

$$State(s) \neq f \circ f \circ z \tag{4}$$

by which double occurrences of fluents are prohibited in any state which is associated with a situation. (It will be explained shortly why "$\circ$" is not required to be idempotent to this end.) Finally, the Fluent Calculus uses the expressions $Holds(f,z)$—denoting that f holds in state z—and the common $Holds(f,s)$—stating that fluent f holds in situation s—, though not as part of the signature but as mere abbreviations of equality sentences:

$$\begin{array}{lcl} Holds(f,z) & \stackrel{\text{def}}{=} & (\exists z')\; z = f \circ z' \\ Holds(f,s) & \stackrel{\text{def}}{=} & Holds(f, State(s)) \end{array} \tag{5}$$

So-called state update axioms specify the entire relation between the states at two consecutive situations. Deterministic actions with only direct and closed effects[7] give rise to the simplest form of state update axioms, where a mere equation relates a successor state $State(Do(A,s))$ to the preceding state $State(s)$:

$$\begin{array}{l} Poss(A(\vec{x}),s) \wedge \Delta(\vec{x},s) \supset \\ \quad (\exists \vec{y})\, State(Do(A(\vec{x}),s)) \circ \vartheta^- = State(s) \circ \vartheta^+ \end{array} \tag{6}$$

where ϑ^- are the negative effects and ϑ^+ the positive effects, resp., of action $A(\vec{x})$ under condition $\Delta(\vec{x},s)$ (and where $\vec{y}$ are the variables in ϑ^-, ϑ^+ which are not among $\vec{x}$).[8] While actions may have conditional effects, and hence multiple state update axioms, there is an assumptions generally made, namely, that a set of state update axioms be consistent and complete, in the following sense. Let Ax be a domain axiomatization, then for any action $A(\vec{x})$ the following holds: First, for any two different axioms (6) in Ax for action $A(\vec{x})$ and with conditions $\Delta_1(\vec{x},s)$ and $\Delta_2(\vec{x},s)$, we have

$$EUNA \models \neg[Poss(A(\vec{x}),s) \wedge \Delta_1(\vec{x},s) \wedge \Delta_2(\vec{x},s)] \tag{7}$$

(That is, no two different state update axioms for one action apply in the same situation.) Second, if $\Delta_1(\vec{x},s), \ldots, \Delta_n(\vec{x},s)$ are the conditions of all state update axioms (6) in Ax for action $A(\vec{x})$, then

$$Ax \models Poss(A(\vec{x}),s) \supset \Delta_1(\vec{x},s) \vee \ldots \vee \Delta_n(\vec{x},s) \tag{8}$$

(That is, there is always an applicable state update axiom for an action that is possible.)

Moreover, we assume that each action $A(\vec{x})$ is accompanied by a precondition axiom of the form,

$$Poss(A(\vec{x}),s) \;\equiv\; \pi(\vec{x},s) \tag{9}$$

where π is a first-order formula without predicate $Poss$ and with free variables among $\vec{x}, s$ and in which s is the only term of sort sit.

As an example, let $Press(Button(x))$ denote the action of pressing the button next to door x, by which the door opens if it is closed and closes if it is open. This is a suitable pair of state update axioms for this action:

$$\begin{array}{l} Poss(Press(Button(x)),s) \wedge Holds(Closed(x),s) \\ \quad \supset State(Do(Press(Button(x)),s)) \circ Closed(x) \\ \qquad\qquad\qquad\qquad\qquad\qquad = State(s) \\ Poss(Press(Button(x)),s) \wedge \neg Holds(Closed(x),s) \\ \quad \supset State(Do(Press(Button(x)),s)) \\ \qquad\qquad\qquad\qquad\qquad = State(s) \circ Closed(x) \end{array} \tag{10}$$

That is to say, if x is currently closed then $Closed(x)$ becomes false whereas it becomes true if x is currently open. Let the precondition of $Press(Button(x))$ be given by the axiom $Poss(Press(Button(x)),s) \equiv Holds(InFrontOf(x),s)$. Suppose a scenario where the initial situation is described by formula (3), and consider the action of pressing the button next to $Door2$. The result can be inferred using the instance $\{x/Door2, s/S_0\}$ of the second one of our state update axioms (10): After verifying that $Poss(Press(Button(Door2)),S_0)$ and $\neg Holds(Closed(Door2),S_0)$, we can replace the expression $State(S_0)$ in the entailed equation by the term which equals $State(S_0)$ according to (3). So doing yields—setting $S_1 = Do(Press(Button(Door2)),S_0)$ and repeating the relevant additional information in (3) about z—,

$$\begin{array}{l} (\exists z)\,[\, State(S_1) = Closed(Door1) \circ InFrontOf(Door2) \\ \qquad\qquad\qquad \circ\, z \circ Closed(Door2) \\ \qquad \wedge\; (\forall y, z')\,(z \neq InFrontOf(Door1) \circ z' \wedge \\ \qquad\qquad\qquad z \neq InFrontOf(Door3) \circ z')\,] \end{array}$$

[7] By closed effects we mean that an action does not have an unbounded number of direct effects.

[8] This scheme is the reason for not stipulating that "$\circ$" be idempotent, contrary to what one might intuitively expect. For if the function were idempotent, then the equation would not imply that $State(Do(A,s))$ does not include ϑ^-.

We thus obtain from an incomplete initial specification a still partial description of the successor state, which in particular includes the unaffected fluents $Closed(Door1)$ and $InFrontOf(Door2)$ and the still valid information about the robot not being in front of the two doors $Door1$ and $Door3$. Hence, all these pieces of information survived the computation of the effect of the action and so need not be carried over by separate application of axioms. This illustrates why and how state update axioms provide a solution not only to the representational but also the inferential Frame Problem.

Under the provision that actions have only closed effects, state update axioms of the form (6) can be fully mechanically generated from a set of simple Situation Calculus-style effect axioms if the latter can be assumed to provide a complete account of the relevant effects of an action. It has been proved that a collection of thus generated state update axioms correctly reflects the fundamental assumption of persistence. This is the primary theorem of the simple Fluent Calculus [Thielscher, 1999].

4 AXIOMATIZING KNOWLEDGE UPDATE

4.1 Extending the signature

The only addition to the signature of the basic Fluent Calculus required to represent knowledge, is the predicate

$$KState : \ sit \times state$$

with the intended meaning that according to the robot's knowledge the second argument is a possible state in the situation denoted by the first argument. On this basis, the fact that some property of a situation is known to the robot is specified using the macro $Knows$, which is defined as follows:

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

where

$$\begin{aligned} HOLDS(f,z) &\stackrel{\text{def}}{=} Holds(f,z) \\ HOLDS(\neg\varphi, z) &\stackrel{\text{def}}{=} \neg HOLDS(\varphi, z) \\ HOLDS(\varphi \land \psi, z) &\stackrel{\text{def}}{=} HOLDS(\varphi, z) \land HOLDS(\psi, z) \\ HOLDS((\forall x)\,\varphi, z) &\stackrel{\text{def}}{=} (\forall x)\,HOLDS(\varphi, z) \end{aligned}$$

4.2 Foundational axioms

The Fluent Calculus with knowledge requires the addition of two foundational axioms, which characterize properties of the knowledge predicate. First, the knowledge of a robot is correct:

$$KState(s, State(s)) \tag{11}$$

(That is, the actual state of the world is always among the states considered possible.) Second, no possible state contains multiple occurrences of fluents (c.f. foundational axiom (4) of the simple Fluent Calculus):

$$KState(s,z) \supset (\forall f, z')\, z \neq f \circ f \circ z' \tag{12}$$

4.3 Knowledge update axioms

While state update axioms specify the effect of actions on the external world, update axioms for knowledge specify their effect on what the robot knows about the world.

Definition 2 A *knowledge update axiom* for an action $A(\vec{x})$ takes the form

$$\begin{aligned} &\Delta(\vec{x}, s) \supset \\ &\quad (\forall z)\,(KState(Do(A(\vec{x}), s), z) \equiv \\ &\qquad\qquad (\exists z')(KState(s,z') \land \Psi(\vec{x}, z, z', s))\,) \end{aligned}$$

where Δ and Ψ are first-order formulas with free variables among $\vec{x}, s$ and $\vec{x}, z, z', s$, resp.

Example 1 Consider a Fluent Calculus signature with fluents $InFrontOf(x)$, $Closed(x)$ and actions $SenseDoor(x)$, $Press(Button(x))$ with the obvious meaning. The robot may sense a door or press the button next to it iff it knows that it is in front of the door:

$$\begin{aligned} &Poss(SenseDoor(x), s) \equiv \\ &\qquad Knows(InFrontOf(x), s) \\ &Poss(Press(Button(x)), s) \equiv \\ &\qquad Knows(InFrontOf(x), s) \end{aligned} \tag{13}$$

Being a sensing action, $SenseDoor$ has no effect on the external world, hence the simple state update axiom

$$\begin{aligned} &Poss(SenseDoor(x), s) \supset \\ &\quad State(Do(SenseDoor(x), s)) = State(s) \end{aligned}$$

The action does, however, affect the knowledge of the robot in that the actual status of the door becomes known, as specified by knowledge update axiom (2) of Section 2.

Pressing the button next to a door causes a closed door to open and an open door to close; hence the state update axiom (10) of Section 3. If the robot knows about this effect, then the appropriate knowledge update axiom mirrors the actual update:

$$\begin{aligned} &Poss(Press(Button(x)), s) \supset \\ &\quad (\forall z)\,(KState(Do(Press(Button(x)), s), z) \equiv \\ &\qquad\quad (\exists z')\,(KState(s,z') \land \\ &\quad [Holds(Closed(x), z') \supset z \circ Closed(x) = z'] \land \\ &\quad [\neg Holds(Closed(x), z') \supset z = z' \circ Closed(x)])) \end{aligned} \tag{14}$$

Put in words, any previously possible state z' in which door x is closed is considered possible after pressing the button if z' is modified by making false $Closed(x)$, and any previously possible state z' in which door x is open is considered possible after pressing the button if z' is modified by adding $Closed(x)$. ∎

Our example illustrates two kinds of actions with special properties as regards knowledge, namely, pure sensing actions and actions of whose effects the agent has accurate knowledge.

Definition 3 Consider a set of state and knowledge update axioms Ax.

1. An action $A(\vec{x})$ is *pure sensing* in Ax if it has a single state update axiom and if this axiom is of the form

$$Poss(A(\vec{x}), s) \supset State(Do(A(\vec{x}), s)) = State(s)$$

2. Let

$$\begin{array}{l} Poss(A(\vec{x}), s) \wedge \Delta_1(\vec{x}, s) \supset \\ \quad (\exists \vec{y}_1)\, State(Do(A(\vec{x}), s)) \circ \vartheta_1^- = State(s) \circ \vartheta_1^+ \\ \qquad \vdots \\ Poss(A(\vec{x}), s) \wedge \Delta_n(\vec{x}, s) \supset \\ \quad (\exists \vec{y}_n)\, State(Do(A(\vec{x}), s)) \circ \vartheta_n^- = State(s) \circ \vartheta_n^+ \end{array}$$

be all state update axioms for an action $A(\vec{x})$, then $A(\vec{x})$ is *accurately known* in Ax if the following is the unique knowledge update axiom for this action:[9]

$$\begin{array}{l} Poss(A(\vec{x}), s) \supset \\ \quad (\forall z)\, (KState(Do(A(\vec{x}), s), z) \equiv \\ \qquad (\exists z')(KState(s, z') \wedge Poss(A(\vec{x}), z') \wedge \\ \qquad\quad [\, \Delta_1(\vec{x}, z') \supset (\exists \vec{y}_1)\, z \circ \vartheta_1^- = z' \circ \vartheta_1^+ \,] \wedge \\ \qquad\qquad \vdots \\ \qquad\quad [\, \Delta_n(\vec{x}, z') \supset (\exists \vec{y}_n)\, z \circ \vartheta_n^- = z' \circ \vartheta_n^+ \,])) \end{array}$$

Our approach to robot knowledge enjoys two important properties as regards these two kinds of actions.

Theorem 4

1. *Pure sensing does not affect the world state.*

[9]Below, the expression $\Delta_i(\vec{x}, z)$ stands for $\Delta_i(\vec{x}, s)$ with each $Holds(f, s)$ replaced by $Holds(f, z)$; and the expression $Poss(A(\vec{x}), z)$ stands for $\pi(\vec{x}, s)$ with each $Holds(f, s)$ replaced by $Holds(f, z)$], where $\pi(\vec{x}, s)$ is the specification for $Poss(A(\vec{x}), s)$ (c.f. precondition schema (9)).

2. *For any situation s, any accurately known action a which is known to be possible in s, and any fluent f not affected by a, f is known to hold after performing a in s iff it is known to hold in s.*

Proof: The first claim follows immediately by definition of pure sensing actions.

For the second claim, observe first that if a is accurately known and possible in s, then the knowledge update axiom implies,

$$\begin{array}{l} (\forall z)\, (KState(Do(A(\vec{x}), s), z) \equiv \\ \quad (\exists z')(KState(s, z') \wedge Poss(A(\vec{x}), z') \wedge \\ \qquad [\, \Delta_1(\vec{x}, z') \supset (\exists \vec{y}_1)\, z \circ \vartheta_1^- = z' \circ \vartheta_1^+ \,] \wedge \\ \qquad\quad \vdots \\ \qquad [\, \Delta_n(\vec{x}, z') \supset (\exists \vec{y}_n)\, z \circ \vartheta_n^- = z' \circ \vartheta_n^+ \,])) \end{array}$$

Since a is known to be possible in s, the assumptions of consistency and completeness (c.f. (7),(8)) imply that for each state z' satisfying $KState(s, z')$, there is a unique $i = 1, \ldots, n$ such that $\Delta_i(\vec{x}_i, z')$ is true. Let

$$(\exists \vec{y}_i)\, z \circ \vartheta_i^- = z' \circ \vartheta_i^+$$

be the equation implied by this $\Delta_i(\vec{x}_i, z')$. In turn, this equation implies $Holds(f, z)$ iff $Holds(f, z')$ according to the rule of distribution (Proposition 1). For f is not amongst the fluents in $\vartheta_i^-, \vartheta_i^+$ following the assumption that f is not affected by a. Hence, if f is true in all states possible in s, then f is true in all states possible in $Do(a, s)$. Conversely, if f is false in some state possible in s, then f is false in some state possible in $Do(a, s)$. ∎

The first one of these two fundamental results coincides with a property of the Situation Calculus-based approach to sensing actions of [Scherl and Levesque, 1993], while the second one generalizes a property from [Scherl and Levesque, 1993] in that we additionally distinguish between the effect of the action itself and the robot's awareness of it. Note the necessity of the condition in Item 2 which requires the robot to know that the action is possible. For otherwise the mere executability of the action may provide the robot with new knowledge.

Item 2 shows that knowledge update axioms provide a solution to the representational Frame Problem for knowledge: Everything that is known before an action is performed is still known afterwards, provided it is known to being unaffected by the action. Knowledge update axioms moreover lay the foundations for overcoming the inferential aspect of the Frame Problem for knowledge, too. The following simple inference scheme can be employed to this end. Suppose that the knowledge of the robot about a situation σ is given by $KState(\sigma, z) \supset \Phi(z)$. Suppose further an action $A(\vec{\tau})$, sensing or not, with knowledge update axiom (1) is performed in σ. Then an immediate logical consequence of the instance $\{\vec{x}/\vec{\tau}, s/\sigma\}$ of (1) and the implication just mentioned, is

$$KState(Do(A(\vec{\tau}), \sigma), z) \supset (\exists z')(\Phi(z') \land \Psi(\vec{\tau}, z, z', s))$$

(assuming successful evaluation of Δ), which provides a specification of what is known about the successor situation $Do(A(\vec{\tau}), \sigma)$. There is no need to carry over to the new situation all pieces of knowledge one-by-one and using separate instances of axioms.

Example 1 (continued) Suppose that of the initial situation the robot knows that it is in front of *Door1* and not in front of *Door2* and that *Door2* is closed, that is,

$$\begin{aligned} & KState(S_0, z) \supset \\ & \quad Holds(InFrontOf(Door1), z) \land \\ & \quad \neg Holds(InFrontOf(Door2), z) \land \\ & \quad Holds(Closed(Door2), z) \end{aligned} \tag{15}$$

Let $S_1 = Do(SenseDoor(Door1), S_0)$ and consider the instance $\{x/Door1, s/S_0\}$ of knowledge update axiom (2). Then the sub-formula $KState(S_0, z)$ of this instance can be replaced by its consequence as given in (15), which yields, after evaluating the antecedent against the precondition axiom in (13),

$$\begin{aligned} & KState(S_1, z) \supset \\ & \quad Holds(InFrontOf(Door1), z) \land \\ & \quad \neg Holds(InFrontOf(Door2), z) \land \\ & \quad Holds(Closed(Door2), z) \land \\ & \quad (Holds(Closed(Door1), z) \equiv \\ & \qquad\qquad Holds(Closed(Door1), S_0)) \end{aligned}$$

We thus obtain a description of what is known about the successor state, which in particular includes the unaffected knowledge about $InFrontOf(Door1)$, $InFrontOf(Door2)$, and $Closed(Door2)$. Hence, all this knowledge is still readily available. ■

4.4 Conditional actions

Employing a theory of sensing actions for robot planning is known to require more complex a notion of a plan than given by the classical view of plans as mere sequences of elementary actions [Levesque, 1996]. At the very least, a robot must be able to condition its course of actions on the result of a sensing action. This minimal requirement can be satisfied in our approach by introducing the concept of a *conditional* action, based on the function

$$If : fluent \times action \mapsto action$$

An instance $If(f, a)$ shall be interpreted as denoting action a if condition f is known to hold, otherwise as denoting the 'action' of doing nothing. A conditional action is defined as possible iff the truth value of the condition is known to the robot and if, provided the condition is true, the respective action is possible:

$$\begin{aligned} Poss(If(f, a), s) \equiv & [Knows(f, s) \lor Knows(\neg f, s)] \\ & \land [Knows(f, s) \supset Poss(a, s)] \end{aligned} \tag{16}$$

The effect of a conditional action on the world state and on the robot's knowledge state is identical to the effect of the action if it applies; otherwise, the conditional has no effect:

$$\begin{aligned} & Poss(If(f, a), s) \supset \\ & \quad [Knows(f, s) \supset \\ & \qquad State(Do(If(f, a), s)) = State(Do(a, s)) \land \\ & \qquad (\forall z)\,(KState(Do(If(f, a), s), z) \equiv \\ & \qquad\qquad\qquad KState(Do(a, s), z))\,] \land \\ & \quad [Knows(\neg f, s) \supset \\ & \qquad State(Do(If(f, a), s)) = State(s) \land \\ & \qquad (\forall z)\,(KState(Do(If(f, a), s), z) \equiv \\ & \qquad\qquad\qquad KState(s, z))\,] \end{aligned} \tag{17}$$

Example 2 Suppose that the robot knows it is in front of *Door1*. It is not given whether the door is open. Formally,

$$KState(S_0, z) \supset Holds(InFrontOf(Door1), z) \tag{18}$$

Let Ax denote the conjunction of this axiom, the precondition and the state and knowledge update axioms for $SenseDoor(x)$ and $Press(Button(x))$ from above, axioms (16) and (17) defining If, and the foundational axioms of the Fluent Calculus. The task shall be to find a plan after whose execution the robot knows that *Door1* is open. A solution is the term

$$\begin{aligned} \sigma = & Do(If(Closed(Door1), Press(Button(Door1))), \\ & \quad Do(SenseDoor(Door1), S_0)) \end{aligned}$$

which satisfies $Ax \models Knows(\neg Closed(Door1), \sigma)$:

From (18) and (13), $Poss(SenseDoor(Door1), S_0)$. Let $S_1 = Do(SenseDoor(Door1), S_0)$, then from (2),

$$\begin{aligned} KState(S_1, z) \supset & KState(S_0, z) \land \\ & [Holds(Closed(Door1), z) \equiv \\ & \ Holds(Closed(Door1), S_0)] \end{aligned}$$

which implies both that $Knows(Closed(Door1), S_1) \lor Knows(\neg Closed(Door1), S_1)$ and, according to (18), $Knows(InFrontOf(Door1), S_1)$. By (16) and (13), $Poss(If(Closed(Door1), Press(Button(Door1))), S_1)$. Hence, from (17), (14), and (12) it follows that

$$\begin{aligned} & KState(Do(If(Closed(Door1), \\ & \qquad\qquad Press(Button(Door1))), S_1), z) \\ & \quad \supset \neg Holds(Closed(Door1), z) \end{aligned}$$

which implies $Ax \models Knows(\neg Closed(Door1), \sigma)$. ■

5 WHAT DOES A ROBOT NOT KNOW?

The explicit notion of a state in the Fluent Calculus for the representation of state knowledge offers an intriguingly simple and elegant way of reasoning about what a robot does *not* know, following the motivation of [Lakemeyer and Levesque, 1998]. To specify that a robot knows and only knows certain facts about the state of the world in a situation S, one employs an axiom of the form $KState(S, z) \equiv \Phi(z)$, where Φ describes all that is known about z. On this basis, it is straightforward to prove that, say, the truth value of some fluent f is *not* known in S by proving validity of $(\exists z)\,(\Phi(z) \wedge Holds(f, z)) \wedge (\exists z')\,(\Phi(z') \wedge \neg Holds(f, z'))$.

Example 3 Suppose the robot knows that in situation S_0 it is in front of two doors *Door1* and *Door2* and that it knows that at least one of them is not closed, but it does not know which one.[10] This combination of knowledge with ignorance is formally specified by,

$$\begin{aligned} & KState(S_0, z) \equiv \\ & \quad Holds(InFrontOf(Door1), z) \wedge \\ & \quad Holds(InFrontOf(Door2), z) \wedge \\ & \quad [\neg Holds(Closed(Door1), z) \vee \\ & \qquad \neg Holds(Closed(Door2), z)\,] \wedge \\ & \quad (\forall f, z')\, z \neq f \circ f \circ z' \end{aligned} \tag{19}$$

Note that the last conjunct is necessary in order not to produce a logical contradiction to foundational axiom (12). Let Ax denote the conjunction of this axiom, the precondition and the state and knowledge update axioms for $SenseDoor(x)$ from above, and the foundational axioms of the Fluent Calculus. Then we can draw the following conclusions, which exemplify the interaction between knowing, not knowing, and sensing.

1. The robot knows that some door is open initially, that is,

   $$Ax \models Knows((\exists x)\, \neg Closed(x), S_0)$$

   This can be easily seen from the equivalent proposition

   $$\begin{aligned} Ax \models (\forall z)\, (KState(S_0, z) \supset \\ (\exists x)\, \neg Holds(Closed(x), z)) \end{aligned}$$

   which follows directly from (19).

2. However, the robot does not know of any particular open door initially, that is,

   $$Ax \models \neg(\exists x)\, Knows(\neg Closed(x), S_0)$$

   This follows from the equivalent proposition

   $$\begin{aligned} Ax \models (\forall x)(\exists z)\, (KState(S_0, z) \wedge \\ Holds(Closed(x), S_0)) \end{aligned}$$

   For in case $x = Door1$ the state

   $$\begin{aligned} z = & InFrontOf(Door1) \circ InFrontOf(Door2) \circ \\ & Closed(Door1) \end{aligned}$$

   satisfies the conjunct; in case $x = Door2$ the state

   $$\begin{aligned} z = & InFrontOf(Door1) \circ InFrontOf(Door2) \circ \\ & Closed(Door2) \end{aligned}$$

   satisfies the conjunct; and in case $x \neq Door1$ and $x \neq Door2$ the state

   $$\begin{aligned} z = & InFrontOf(Door1) \circ InFrontOf(Door2) \circ \\ & Closed(Door1) \circ Closed(x) \end{aligned}$$

   satisfies the conjunct.

3. After sensing the state of *Door1*, the robot will know of a particular open door (although it is not known in advance which one), that is,

   $$Ax \models (\exists x)\, Knows(\neg Closed(x), S_1)$$

   where $S_1 = Do(SenseDoor(Door1), S_0)$. This follows from the equivalent proposition

   $$\begin{aligned} Ax \models (\exists x)(\forall z)\, (KState(S_1, z) \supset \\ \neg Holds(Closed(x), z)) \end{aligned}$$

   For, we have $Poss(SenseDoor(Door1), S_0)$ from (13) and (19). Hence, from (2) it follows that

   $$\begin{aligned} (\forall z)\, (KState(S_1, z) \equiv KState(S_0, z) \wedge \\ \neg Holds(Closed(Door1), z)) \end{aligned}$$

   in case $\neg Holds(Closed(Door1), S_0)$, and

   $$\begin{aligned} (\forall z)\, (KState(S_1, z) \equiv KState(S_0, z) \wedge \\ Holds(Closed(Door1), z)) \end{aligned}$$

   in case $Holds(Closed(Door1), S_0)$. Furthermore, from (19) we have

   $$\begin{aligned} (\forall z)\, (KState(S_0, z) \wedge Holds(Closed(Door1), z) \\ \supset \neg Holds(Closed(Door2), z)) \end{aligned}$$

   Altogether, both if $\neg Holds(Closed(Door1), S_0)$ and if $Holds(Closed(Door1), S_0)$ there is some x such that for all z, $KState(S_1, z)$ implies $\neg Holds(Closed(x), z)$. ■

[10] This is an adaptation of the example of [Lakemeyer and Levesque, 1999].

With knowledge update axioms the inferential Frame Problem for 'only knowing' can be solved by employing an inference scheme analogous to the one presented in Section 4.3. Suppose that the given complete knowledge of the robot about a situation σ is specified by $KState(\sigma, z) \equiv \Phi(z)$, as described above. Suppose further an action $A(\vec{\tau})$, sensing or not, with update axiom (1) is performed in σ. Then an immediate logical consequence of the instance $\{\vec{x}/\vec{\tau}, s/\sigma\}$ of (1) and the equivalence just mentioned, is

$$KState(Do(A(\vec{\tau}), \sigma), z) \equiv (\exists z')(\Phi(z') \wedge \Psi(\vec{x}, z, z', s))$$

(assuming successful evaluation of Δ), which provides a complete specification of what is known and what is not known about the successor situation $Do(A(\vec{\tau}), \sigma)$. Again, there is no need to carry over to the new situation all pieces of knowledge and non-knowledge one-by-one and using separate instances of axioms.

6 REASONING ABOUT ABILITY

The independence of state update specifications from knowledge update specifications enables the ready usage of our formalism for the purpose of reasoning about possibly restricted subjective achievability. Two kinds of 'mental' limitations may prevent a robot from reaching a goal although it would be physically able to do so: The robot may lack crucial state knowledge without having at hand the appropriate sensing action, or it may lack complete knowledge of the effect of its actions.

Formal proofs of non-achievability rely on induction over situations along the line of [Reiter, 1993], where the following second-order axiom has been introduced:

$$(\forall \Pi)\,(\Pi(S_0) \wedge (\forall a, s)\,(\Pi(s) \supset \Pi(Do(a, s))) \\ \supset (\forall s)\,\Pi(s))$$

That is, a property Π holds for all situations if it holds initially and if all actions preserve it. Usually, one is interested in proving properties only for those situations that are reachable by an executable sequence of actions. To this end, [Reiter, 1993] adds the following foundational axioms:

$$(\forall s)\,\neg s < S_0$$
$$(\forall a, s, s')\,(s < Do(a, s') \equiv Poss(a, s') \wedge s \leq s')$$

where $s \leq s'$ abbreviates $s < s' \vee s = s'$.

Using induction in the Fluent Calculus requires a domain closure axiom for actions since the induction step can never be proved if there are actions without state update axioms, in which case successor situation may enjoy arbitrary properties. Let $A_1(\vec{x}_1), \ldots, A_n(\vec{x}_n)$ be the actions available to a robot, then this is the corresponding closure axiom:

$$(\forall a)\,(\exists \vec{x}_1)\,a = A_1(\vec{x}_1) \vee \ldots \vee (\exists \vec{x}_n)\,a = A_n(\vec{x}_n)$$

Example 4 Recall robot *Blindie* from the introduction, who may be in front of an open door without knowing the state of that door:

$$\neg Holds(Closed(Door1), S_0) \qquad (20)$$
$$Knows(InFrontOf(Door1), S_0) \qquad (21)$$
$$\neg Knows(\neg Closed(Door1), S_0) \qquad (22)$$

The goal of entering the room cannot be achieved by this robot without assistance. Although it could move into the room through the open door, the robot has no way of arriving at this conclusion if it is not able to sense the states of doors: Let Ax denote the conjunction of the axioms just mentioned, the precondition and the state and knowledge update axioms for $Press(Button(x))$ from above, the foundational axioms including the induction and accompanying axioms, and the following closure axiom, stating that pressing buttons is the only action available to the robot:

$$(\forall a)\,(\exists x)\,a = Press(Button(x))$$

Then the robot can never know whether *Door1* is open or not:

$$Ax \models (\forall s)\,(S_0 \leq s \supset \neg Knows(Closed(Door1), s) \\ \wedge \neg Knows(\neg Closed(Door1), s)$$

This follows from the induction axiom instantiated by $\{\Pi/\lambda s.\,F\}$ where F denotes the entire formula in the range of the quantification. The base case, S_0, is given by (22) and by (20) in conjunction with (11). For the induction step, consider the only action $a = Press(Button(x))$ in conjunction with knowledge update axiom (14) and foundational axiom (12). Then Ax entails,

$$Poss(Press(Button(x)), s) \supset \\ x \neq Door1 \supset \\ (\forall z)\,(KState(Do(Press(Button(x)), s), z) \supset \\ [\,Holds(Closed(Door1), z) \equiv \\ (\exists z')\,KState(s, z') \wedge Holds(Closed(Door1), z')\,])$$

and

$$Poss(Press(Button(x)), s) \supset \\ x = Door1 \supset \\ (\forall z)\,(KState(Do(Press(Button(x)), s), z) \supset \\ [\,Holds(Closed(Door1), z) \equiv \\ (\exists z')\,KState(s, z') \wedge \neg Holds(Closed(Door1), z')\,])$$

The induction step now follows from the induction hypothesis that the robot does not know the status of *Door1* in s. ■

Example 5 Recall robot *Dumbie* from the introduction, who is aware of the fact that a door is somehow under the control of a button next to it without knowing the precise causal relation, namely, that pressing the button always alters the state of the door. Hence, while the state update itself is still suitably described by the two axioms of (10), this is the knowledge update axiom characterizing *Dumbie*:

$$\begin{array}{l} Poss(Press(Button(x)), s) \supset \\ \quad (\forall z)\,(KState(Do(Press(Button(x)), s), z) \equiv \\ \qquad (\exists z')\,(KState(s, z') \;\wedge \\ [Holds(Closed(x), z') \supset z = z' \vee z \circ Closed(x) = z'] \wedge \\ [\neg Holds(Closed(x), z') \supset z = z' \vee z = z' \circ Closed(x)])) \end{array}$$

(Compare this to knowledge update axiom (14), which encodes accurate effect knowledge.) Although it could open a closed door by pressing the button next to it, the robot does not know this: Let Ax denote the knowledge update axiom just mentioned along with the precondition and state update axioms for $Press(Button(x))$ from above, the foundational axioms including including the induction and accompanying axioms, the axiom

$$Knows(Closed(Door1), S_0) \tag{23}$$

and the closure axiom

$$(\forall a)\,(\exists x)\, a = Press(Button(x))$$

Then the robot can never know that it is possible to open the door:

$$Ax \models (\forall s)\,(S_0 \leq s \supset \neg Knows(\neg Closed(Door1), s)$$

This follows from the induction axiom instantiated by $\{\Pi/\lambda s.\, F\}$ where F denotes the entire formula in the range of the quantification. The base case, S_0, is given by (23) in conjunction with (11). For the induction step, consider the only action $a = Press(Button(x))$ in conjunction with the knowledge update axiom of *Dumbie* for this action and foundational axiom (12). Then Ax entails,

$$\begin{array}{l} Poss(Press(Button(x)), s) \supset \\ \quad x \neq Door1 \supset \\ \qquad (\forall z)\,(KState(Do(Press(Button(x)), s), z) \supset \\ \qquad\qquad [\,Holds(Closed(Door1), z) \equiv \\ \quad (\exists z')\, KState(s, z') \wedge Holds(Closed(Door1), z')\,]) \end{array}$$

and

$$\begin{array}{l} Poss(Press(Button(x)), s) \supset \\ \quad x = Door1 \supset \\ \qquad (\exists z)\,(KState(Do(Press(Button(x)), s), z) \supset \\ \qquad\qquad\qquad Holds(Closed(Door1), z)) \end{array}$$

The induction step now follows from the induction hypothesis. ■

7 DISCUSSION

We have developed a formal account of a robot's changing knowledge about the state of its environment. Based on the established predicate calculus formalism for reasoning about actions of the Fluent Calculus, our approach is kept representationally and inferentially simple in that it avoids non-classical extensions to standard predicate logic. Our formalism accounts for both knowledge preconditions of actions and information gathering actions which enhance the state knowledge of a robot. Our theory also provides simple and elegant means to reason about what a robot does *not* know and about goal achievability.

The effect of actions on state knowledge is specified by so-called knowledge update axioms, by which is solved the representational Frame Problem for knowledge according to the main theorem of the Fluent Calculus for sensing (Item 2 of Theorem 4). Moreover, knowledge update axioms have been shown to lay the foundations for overcoming the inferential aspect of this Frame Problem, too.

We have axiomatically introduced the concept of a conditional action, by which a robot may condition its intended course of actions on the result of a sensing action included in its plan. Our *If*-construct uses only atomic conditions and only allows for a single conditionally executed action. The integration of more expressive notions, say arbitrarily complex conditions and unbounded iteration [Levesque, 1996], is considered an important direction of future research. Following [Levesque *et al.*, 1997], such complex actions can be dealt with in two fundamentally different ways: They can be defined via macros or be integrated into the language. The former approach, actually taken in [Levesque *et al.*, 1997], has the drawback that non-sequential plans (that is, which include conditional and loop statements) cannot be planned by deduction as they are not part of the language. The latter approach, on the other hand, requires complete reification of arbitrary formulas. Inasmuch as reification plays an important role in the simple Fluent Calculus anyway, it is this second alternative approach which seems most promising a route to take towards an extension of our formalism.

Knowledge and sensing actions were first investigated in [Moore, 1985] in the context of the Situation Calculus [McCarthy, 1963], and in [Scherl and Levesque, 1993] this approach was combined with the solution to the Frame Problem provided by so-called successor state axioms [Reiter, 1991]. The basic idea of this approach is to represent state knowledge by a binary situation-situation relation $K(s, s')$, meaning that as far as the robot knows in situation s it could as well be in situation s'. Hence, every given fact about any such s' is considered possible by the robot. Having readily available the explicit notion of a state in the Fluent Calculus, our formalization avoids this indirect encoding of state knowledge, which is intuitively less appealing because it seems that a robot should always know exactly which *situation* it is in—after all, situations in the Situation Calculus are merely sequences of actions that have been or will be taken by the robot [Levesque *et al.*, 1998].

Apart from this clash of intuitions, there is a more crucial difference between our approach and that of [Moore, 1985; Scherl and Levesque, 1993]: The latter defines the effect of a non-sensing action a on the robot's state knowledge via the equivalence relation $K(Do(a, s), s'') \equiv (\exists s')\,(K(s, s') \wedge s'' = Do(a, s'))$. Hence, the very same successor state axioms apply to both the state update (when moving from s to $Do(a, s)$) and the knowledge update (when moving from s' to $s'' = Do(a, s')$). In contrast, with independent specifications of state and knowledge update, our formalism furnishes a ready approach for representing and reasoning about goal achievability that is possibly restricted due to limited knowledge of the effects of actions. This separating what a user knows from what a robot knows distinguishes our theory from other existing accounts of sensing action and knowledge, too, such as [Lobo *et al.*, 1997; Son and Baral, 1998], where also non-sensing actions have identical effect on the external and internal states.

Representing and reasoning about non-knowledge has previously been realized in the context of the Situation Calculus [Lakemeyer and Levesque, 1998; Lakemeyer and Levesque, 1999]. Two approaches to 'only knowing' have been offered, one of which is by a non-standard semantics while the other one is an axiomatization in classical logic but with two complex second-order axioms involved. Exploiting the reification of fluents and states, knowledge and non-knowledge can be expressed in our approach by mere first-order sentences along with the standard semantics of classical predicate logic.

Goal achievability has been analyzed previously in [Lin and Levesque, 1997], independently of the second author's approach to knowledge and sensing. The result of the present paper can thus be viewed as a unifying theory for representing and reasoning about knowledge, sensing, and mental ability of achieving goals.

Our further plans for future work include the integration of the formalism for reasoning about a robot's knowledge into existing extensions of the basic Fluent Calculus. A particularly interesting combination is that of ramifications and knowledge. Just like it may have only restricted knowledge of the direct effects of actions, a robot may lack knowledge of state constraints and of indirect effects. The solution to the Ramification Problem of [Thielscher, 1997] is readily available for a combination with knowledge update axioms to allow a robot to reason about the indirect effects it is aware of.

References

[Bibel, 1998] Wolfgang Bibel. Let's plan it deductively! *Artificial Intelligence*, 103(1–2):183–208, 1998.

[Hölldobler and Schneeberger, 1990] Steffen Hölldobler and Josef Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.

[Hölldobler and Thielscher, 1995] Steffen Hölldobler and Michael Thielscher. Computing change and specificity with equational logic programs. *Annals of Mathematics and Artificial Intelligence*, 14(1):99–133, 1995.

[Lakemeyer and Levesque, 1998] Gerhard Lakemeyer and Hector J. Levesque. $\mathcal{AOL}$: A logic of acting, sensing, knowing, and only knowing. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 316–327, Trento, Italy, 1998.

[Lakemeyer and Levesque, 1999] Gerhard Lakemeyer and Hector J. Levesque. Query evaluation and progression in $\mathcal{AOL}$ knowledge bases. In T. Dean, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 124–131, Stockholm, Sweden, 1999.

[Levesque *et al.*, 1997] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31(1–3):59–83, 1997.

[Levesque *et al.*, 1998] Hector Levesque, Fiora Pirri, and Ray Reiter. Foundations for a calculus of situations. *Linköping Electronic Articles in Computer and Information Science*, 3(18), 1998. URL: http://www.ep.liu.se/ea/cis/1998/018/.

[Levesque, 1996] Hector J. Levesque. What is planning in the presence of sensing? In B. Clancey and D. Weld, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 1139–1146, Portland, OR, August 1996. MIT Press.

[Lin and Levesque, 1997] Fangzhen Lin and Hector Levesque. What robots can do: Robot programs and effective achievability, 1997. (Manuscript).

[Lobo *et al.*, 1997] Jorge Lobo, Gisela Mendez, and Stuart R. Taylor. Adding knowledge to the action description language $\mathcal{A}$. In B. Kuipers and B. Webber, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 454–459, Providence, RI, July 1997. MIT Press.

[McCarthy, 1963] John McCarthy. *Situations and Actions and Causal Laws.* Stanford Artificial Intelligence Project, Memo 2, 1963.

[Moore, 1985] Robert Moore. A formal theory of knowledge and action. In J. R. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, 1985.

[Reiter, 1991] Ray Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[Reiter, 1993] Ray Reiter. Proving properties of states in the situation calculus. *Artificial Intelligence*, 64:337–351, 1993.

[Scherl and Levesque, 1993] Richard Scherl and Hector Levesque. The frame problem and knowledge-producing actions. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 689–695, Washington, DC, July 1993.

[Son and Baral, 1998] Tran Cao Son and Chitta Baral. Formalizing sensing actions—a transition function based approach, 1998. (Manuscript).

[Störr and Thielscher, 2000] Hans-Peter Störr and Michael Thielscher. A new equational foundation for the fluent calculus, 2000. (Manuscript.) URL: http://pikas.inf.tu-dresden.de/~mit/publications/conferences/FCeq.ps.

[Thielscher, 1997] Michael Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.

[Thielscher, 1998] Michael Thielscher. Introduction to the Fluent Calculus. *Electronic Transactions on Artificial Intelligence*, 2(3–4):179–192, 1998. URL: http://www.ep.liu.se/ea/cis/1998/014/.

[Thielscher, 1999] Michael Thielscher. From Situation Calculus to Fluent Calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111(1–2):277–299, 1999.