$\mathcal{AOL}$: a logic of acting, sensing, knowing, and only knowing

Gerhard Lakemeyer
Department of Computer Science
Aachen University of Technology
D-52056 Aachen
Germany
gerhard@cs.rwth-aachen.de

Hector J. Levesque
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 3A6
hector@cs.toronto.edu

Abstract

This work is motivated by the existence of two useful but quite different knowledge representation formalisms, the situation calculus due to McCarthy, and the logic $\mathcal{OL}$ of only knowing due to Levesque. In this paper, we propose the logic $\mathcal{AOL}$, which combines both approaches in a clean and natural way. We present a semantics for $\mathcal{AOL}$ which generalizes the semantics of $\mathcal{OL}$ to account for actions, and a sound and complete set of axioms for $\mathcal{AOL}$ which generalizes the Lin and Reiter foundational axioms of the situation calculus to account for only knowing. The logic is compatible with earlier work on knowledge and action in that the solution to the frame problem for knowledge proposed by Scherl and Levesque becomes now a theorem of $\mathcal{AOL}$. We also demonstrate that the logic avoids certain anomalies present in related work by Lakemeyer. Finally we provide a mapping from $\mathcal{OL}$ into $\mathcal{AOL}$ such that a sentence of $\mathcal{OL}$ is valid iff its mapping is a theorem of $\mathcal{AOL}$, thus providing, for the first time, a complete axiomatic characterization of $\mathcal{OL}$.

1 Introduction

This work is motivated by the existence of two useful but quite different knowledge representation formalisms:

- the *situation calculus* [McC63] is a dialect of first order logic for representing and reasoning about the preconditions and effects of actions. A recent second-order refinement explored by Lin and Reiter [LR94] has been shown to be useful for high-level robot and agent control [LRL97], exploiting a simple solution to the frame problem presented in [Rei91].
- the logic of *only knowing* $\mathcal{OL}$ [Lev90] is a quantified modal logic for representing and reasoning about the *de dicto* and *de re* knowledge[1] of an agent, and all that that agent knows.[2] This language has been found useful for capturing autoepistemic reasoning [Moo85b] within a purely monotonic logic, as well as certain forms of relevance [Lak95].

In this paper, we propose the logic $\mathcal{AOL}$, which includes in a clean and natural way both the situation calculus noted above and an embedding of the logic of only knowing. The amalgamation is carried out at both the semantic and the syntactic levels. While the semantics naturally extends the model theory of $\mathcal{OL}$ to account for actions, the axioms, which are sound and complete for the semantics, can be seen as a version of the foundational axioms of the situation calculus proposed in [LR94], generalized to deal with the much richer ontology required for only knowing.

The motivation for the amalgamation is perhaps best seen in the following example. Suppose we have a robot that knows nothing about the initial state of the environment, but that there is a sensing action, reading a sonar, which tells the robot when it is getting close to a wall. Then we would like to prove the following:

1. in the situation that results from reading the sonar, the robot knows whether the wall is close;
2. assuming the robot knows the sonar is working, it also knows in the initial state that it will know whether the wall is close after checking its sonar;
3. suppose the robot checks its sonar and discovers that the wall is not close. If it now moves towards the wall, it no longer knows whether or not the wall is close; if

[1] *De dicto* and *de re* knowledge refers to the distinction between *knowing that* and *knowing who* [Kap71].

[2] Actually, it is belief that is dealt with in $\mathcal{OL}$, but we will use the two terms interchangeably, unless noted otherwise.

it moves away from the wall instead, it continues to know whether or not it is close.

These are all simple and reasonable properties involving knowledge and action. Observe that to get them right, it is necessary to reason about the effects of sensing actions, knowledge about knowledge and action, and all that is known. The latter is necessary, in particular, if an agent wants to reason about its own ignorance without having to be told explicitly what it does not know. Furthermore, knowledge about ignorance plays an important role in guiding an agent's actions such as deciding whether it is necessary to use a sensor.

Ours is certainly not the first approach combining knowledge and action (see, for example, [Moo85a, SL93, Lak96]). In fact, $\mathcal{AOL}$ is compatible with this line of work in that the solution to the frame problem for knowledge proposed by Scherl and Levesque [SL93], which builds on [Moo85a], becomes a logical consequence of the axioms. So far, only Lakemeyer [Lak96] has proposed an amalgamation of only knowing and action in a quantified logic.[3] However, he provides only a semantics, but no axioms. In this paper we also point to certain anomalies in his logic and demonstrate how they are avoided in $\mathcal{AOL}$. Finally we provide a mapping from $\mathcal{OL}$ into $\mathcal{AOL}$ such that a sentence in $\mathcal{OL}$ is valid iff its mapping into $\mathcal{AOL}$ is a logical consequence of the axioms. This provides, for the first time, a complete axiomatic characterization of $\mathcal{OL}$.

The rest of the paper is organized as follows. In Sections 2 and 3, we briefly review the situation calculus and the logic $\mathcal{OL}$, respectively. In Section 4, we present the new logic $\mathcal{AOL}$, both semantically and axiomatically. The properties of knowledge and action are studied in more detail in Section 5. Section 6 formalizes the robot example above. In Section 7, we show an embedding of $\mathcal{OL}$ into $\mathcal{AOL}$. In Section 8, we compare our approach with the amalgamation presented in [Lak96]. Section 9 presents a brief summary and suggests areas of future work.

2 Situation Calculus

One increasingly popular language for representing and reasoning about the preconditions and effects of actions is the situation calculus [McC63]. We will only go over the language briefly here noting the following features: all terms in the language are one of three sorts, ordinary objects, actions or situations; there is a special constant S_0 used to denote the *initial situation*, namely that situation in which no actions have yet occurred; there is a distinguished binary function symbol *do* where $do(a, s)$ denotes the successor situation to s resulting from performing the action a; relations whose truth values vary from situation to situation, are called relational *fluents*, and are denoted by predicate symbols taking a situation term as their last argument; similarly, functions varying across situations are called functional fluents and are denoted analogously; finally, there is a special predicate $Poss(a, s)$ used to state that action a is executable in situation s.

Within this language, we can formulate theories which describe how the world changes as the result of the available actions. One possibility is a *basic action theory* of the following form [Rei91]:

- Axioms describing the initial situation, S_0.
- Action precondition axioms, one for each primitive action a, characterizing $Poss(a, s)$.
- Successor state axioms, one for each fluent F, stating under what conditions $F(\vec{x}, do(a, s))$ holds as a function of what holds in situation s. These take the place of the so-called effect axioms, but also provide a solution to the frame problem [Rei91].
- Domain closure and unique names axioms for the primitive actions.
- A collection of foundational, domain independent axioms.

In [LR94] the following foundational axioms are considered:[4]

1. $\forall s \forall a. S_0 \neq do(a, s)$.
2. $\forall a_1, a_2, s_1, s_2.\ do(a_1, s_1) = do(a_2, s_2) \supset (a_1 = a_2 \wedge s_1 = s_2)$.
3. $\forall P.\ P(S_0) \wedge [\forall s \forall a. (P(s) \supset P(do(a, s)))] \supset \forall s P(s)$.
4. $\forall s.\ \neg(s < S_0)$.
5. $\forall s, s', a.\ (s < do(a, s') \equiv (Poss(a, s') \wedge s \leq s'))$, where $s \leq s'$ is an abbreviation for $s < s' \vee s = s'$.

The first three axioms serve to characterize the space of all situations, making it isomorphic to the set of ground terms of the form $do(a_1, \cdots, do(a_n, S_0) \cdots)$. The third of these is a second-order induction axiom that ensures that there are no situations other than those accessible using *do* from S_0. The final two axioms serve to characterize a $<$ relation between situations. Later, we will be introducing abbreviations of various sorts into our representation language,

[3] In the context of modeling belief revision, [dVS94] provide axioms for only knowing in the situation calculus, but these are limited to the propositional case.

[4] In addition to the standard axioms of equality

and we can in fact do so here, defining the $<$ relation as an abbreviation for a certain second-order formula:

$$s < s' \doteq \forall P[\ldots \supset P(s, s')]$$

where the ellipsis stands for the conjunction of

$$\forall a, s.Poss(a, s) \supset P(s, do(a, s))$$
$$\forall s_1, s_2, s_3.P(s_1, s_2) \wedge P(s_2, s_3) \supset P(s_1, s_3)$$

It is not hard to show that with this definition, the final two axioms do not have to be postulated, and are in fact logical theorems.

3 The Logic $\mathcal{OL}$

The language of $\mathcal{OL}$ is a modal first-order dialect with equality and function symbols plus a countably infinite set of standard names $\mathcal{N} = \{^{\#}0, ^{\#}1, ^{\#}2, \ldots\}$ which will serve as our universe of discourse. As discussed in more detail in [Lev84], standard names allow interesting distinctions between *de dicto* and *de re* beliefs (see below for an example). Terms and atomic formulas are defined as usual. A term is called *primitive* if it consists of a function symbol followed by standard names as argument. Similarly a formula is called primitive if it consists of a predicate symbol followed by standard names. Arbitrary formulas of $\mathcal{OL}$ are constructed in the usual way from the atomic formulas, equality, the connectives $\neg$ and $\vee$, the quantifier $\forall$,[5] and the modal operators $\boldsymbol{K}$ and $\boldsymbol{O}$, where $\boldsymbol{K}\alpha$ should be read as "the agent knows α" and $\boldsymbol{O}\alpha$ as "the agent only knows α." *Sentences* are formulas without free variables. A formula is called *objective* if it does not contain any modal operators. Vector notation will be used freely for sequences, for example, $\forall\vec{x}$ for $\forall x_1 \cdots \forall x_k$.

The semantics of $\mathcal{OL}$ is based on the familiar notion of a world, which assigns meaning to the nonlogical symbols of the language. $\mathcal{OL}$ makes the assumption that all worlds have as their universe of discourse the same set which is isomorphic to the standard names, that is, an individual is identified with a unique name. A world is then completely specified by providing the meaning of every primitive term and formula:

Definition 3.1: A *world* w is a function from primitive expressions into $\{0, 1\} \cup \mathcal{N}$, where $w[p] \in \{0, 1\}$ for primitive formulas p, and $w[t] \in \mathcal{N}$ for primitive terms t.

Given a world w, the denotation of an arbitrary ground term t is defined recursively as

$|n|_w = n$, where n is a standard name.

$|f(t_1, \ldots, t_k)|_w = w[f(n_1, \ldots, n_k)]$, where $n_i = |t_i|_w$.

We often write $|\vec{t}|_w$ instead of $\langle |t_1|_w, \ldots, |t_k|_w \rangle$.

While the truth of objective sentences is determined by a single world w, the meaning of sentences of the form $\boldsymbol{K}\alpha$ and $\boldsymbol{O}\alpha$ is defined relative to a *set* of worlds e. $\boldsymbol{K}\alpha$ holds if α is true in all worlds of e. $\boldsymbol{O}\alpha$ holds if $\boldsymbol{K}\alpha$ holds and, in addition, every world that satisfies α is also a member of e. This way e minimizes what is known besides α. e is also called an *epistemic state* and a pair (e, w) is sometimes referred to as a an $\mathcal{OL}$ model.

The semantic rules which determine the truth of a sentence α at a given world w and epistemic state e (denoted as $e, w \models \alpha$) are defined as follows:

$e, w \models F(\vec{t})$	iff	$w[F(\vec{n})] = 1$ and $\vec{n} = \lvert\vec{t}\rvert_w$, where $F(\vec{t})$ is atomic.
$e, w \models t_1 = t_2$	iff	$\lvert t_1\rvert_w = \lvert t_2\rvert_w$
$e, w \models \neg\alpha$	iff	$e, w \not\models \alpha$
$e, w \models \alpha \vee \beta$	iff	$e, w \models \alpha$ or $e, w \models \beta$
$e, w \models \forall x\alpha$	iff	$e, w \models \alpha^x_n$ for all $n \in \mathcal{N}$
$e, w \models \boldsymbol{K}\alpha$	iff	for all $w' \in e$, $e, w' \models \alpha$
$e, w \models \boldsymbol{O}\alpha$	iff	for all w', $w' \in e$ iff $e, w' \models \alpha$

A formula α is valid ($\models_{\text{OL}} \alpha$) iff $e, w \models \alpha$ for all worlds w and all sets of worlds e.[6] We sometimes write $w \models \alpha$ if α is an objective sentence.

Here we only briefly discuss the operators $\boldsymbol{K}$ and $\boldsymbol{O}$. For a detailed discussion of $\mathcal{OL}$, we refer the reader to [Lev90] and [LL9x]. $\boldsymbol{K}$ has the usual properties of the logic *K45* or *weak S5* [HC68] whose characteristic axioms are:

K: $\boldsymbol{K}(\alpha \supset \beta) \supset (\boldsymbol{K}\alpha \supset \boldsymbol{K}\beta)$
4: $\boldsymbol{K}\alpha \supset \boldsymbol{K}\boldsymbol{K}\alpha$
5: $\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\boldsymbol{K}\alpha$

The Barcan formula $(\forall x\boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x\alpha)$ is also valid since we are assuming a fixed universe of discourse [HC68]. While $\boldsymbol{K}$ is very well understood, this is less the case for $\boldsymbol{O}$ except perhaps when $\boldsymbol{O}$ is applied to an objective sentence. Consider an atomic sentence p. It is easy to see that the only epistemic state e where $\boldsymbol{O}p$ is satisfied is $e = \{w \mid w \models p\}$, that is, the set of all worlds where p is true. It is the "iff" in the semantic rule of $\boldsymbol{O}$ which has the effect of maximizing e. As a result, the objective sentences known at e are exactly the logical consequences of p, which captures the idea that p is all that is known. Note that the meaning of $\boldsymbol{O}$ crucially depends on e as well as on the complement of e. In particular, for $\boldsymbol{O}\alpha$ to be true, α has to be known and all worlds not in e have to falsify α. The story becomes much more complicated with arbitrary formulas in the scope of

[5] Other logical connectives like $\wedge$, $\supset$, and $\equiv$ and the quantifier $\exists$ are used freely and are defined in the usual way.

[6] Levesque used so-called *maximal sets* to define validity, a complication we ignore here for simplicity.

$\boldsymbol{O}$. For example, as shown in [Lev90], using $\boldsymbol{O}$ it is possible to fully reconstruct and extend Moore's Autoepistemic Logic [Moo85b]. However, in this paper we will not be concerned with such issues, and all the examples used here consider only knowing applied to objective sentences.

[Lev90] presents an axiomatization of $\mathcal{OL}$, which is complete for the propositional case, but was recently shown to be incomplete for the full first-order language [HL95]. An interesting by-product of our work is that, by appealing to second-order logic, a complete axiom system for $\mathcal{OL}$ obtains.

To see the utility of $\boldsymbol{O}$, consider an agent who knows the standard name for the current temperature, which could be a particular value on some temperature scale. If this is all she knows then it should follow that she knows what the temperature is and that she does not know what the barometric pressure is. In $\mathcal{OL}$ this can be expressed by the following valid sentence, where n is a standard name and both *temperature* and *pressure* are ordinary constants:

$$\boldsymbol{O}(\mathit{temperature} = n) \supset \\ \exists x \boldsymbol{K}(\mathit{temperature} = x) \wedge \neg \exists y \boldsymbol{K}(\mathit{pressure} = y).$$

Note that the implication would not go through if we replaced $\boldsymbol{O}$ by $\boldsymbol{K}$: knowing the temperature does not rule out knowing the barometric pressure as well. We feel that being able to derive facts about what is *not* known without having to state them as premises is a useful feature of the language that becomes even more important when reasoning about knowledge and action.

4 The Logic $\mathcal{AOL}$

There are two ways of understanding what is required to amalgamate the situation calculus and the logic $\mathcal{OL}$: we can view it in terms of extensions to $\mathcal{OL}$, and how the semantics there needs to be modified, or we can view it in terms of extensions to the situation calculus, and how the foundational axioms need to change. Our approach can be seen as taking both views. We begin by proposing a semantics which extends that of $\mathcal{OL}$ by adding actions and applying it to a slightly extended language of the situation calculus to account for knowledge and standard names. We then provide a small set of axioms which are sound and complete for the semantics and which, taken by themselves, can be thought of as foundational axioms for an extended situation calculus.

The language of $\mathcal{AOL}$ will be a dialect of the second-order predicate calculus, like the situation calculus introduced in Section 2. Again we have three sorts: ordinary objects, actions and situations. The constant S_0, the function *do*, and special predicate $\mathit{Poss}(a, s)$ are exactly as before. We will however require two new special predicates, $\mathit{SF}(a, s)$ and $K_0(s)$, a new constant $\mathtt{0}$ of sort ordinary object, and a new function symbol $\mathit{succ}(x)$, which maps ordinary objects to ordinary objects. The set of (relational and functional) fluents as well as the set of action function symbols is assumed to be finite.

For simplicity, we also make the following restrictions: there are no constants or functions of the situation sort other than S_0 and *do*; action functions do not take situations as arguments; all ordinary object functions other than $\mathtt{0}$ and *succ* are fluents; and all predicates other than those mentioned above are fluents. Finally, we assume that fluents only have situation terms as arguments in the final position.

Recall that $\mathcal{OL}$ assumes a fixed countably infinite domain of ordinary objects, isomorphic to the set of standard names. We want to use standard names as objects in $\mathcal{AOL}$ as well. However, in order to facilitate the axiomatization later on, we will not represent them using infinitely many distinct predicate calculus constants $^{\#}0$,$^{\#}1$, ..., but instead construct them using $\mathtt{0}$ and *succ*. The idea is that $\mathtt{0}$ will play the role of $^{\#}0$, $\mathit{succ}(\mathtt{0})$ the role of $^{\#}1$, *etc.* (similar to the way numerals are represented in number theory). As before, we refer to the set of standard names (now ground terms) as $\mathcal{N}$.

To deal with knowledge in $\mathcal{AOL}$, the biggest change is that we imagine that in addition to S_0 and its successors, there are an uncountable number of other initial and non-initial situations considered as possible epistemic alternatives. To state what is known in S_0, we use K_0. Informally, taking S_0 to be the situation counterpart to the given world w in $\mathcal{OL}$, K_0 is the counterpart to the given epistemic state e in $\mathcal{OL}$. In other words, $K_0(s)$ is intended to hold if s is a situation considered by the agent in S_0 to be possible. How knowledge changes when performing an action a in situation s is governed by $\mathit{SF}(a, s)$ and $\mathit{Poss}(a, s)$ and will be discussed later in Section 5.

4.1 Semantics

Recall that in $\mathcal{OL}$, there are worlds corresponding to all possible interpretations of the predicate and function symbols (over the domain of standard names). Different applications, of course, will use different subsets as part of the given e, but the complement of e is still relevant because of only knowing. We need the same in $\mathcal{AOL}$ with respect to K_0, but more: we need to allow for all possible interpretations of the predicate and function symbols *after all possible sequences of actions*. That is, to ensure that it is possible to know the initial value of a term or formula without also necessarily knowing its value in successor situations, it is necessary that there be initial situations that agree on the values of all terms and formulas but that have

successors that disagree on these values.[7] Thus instead of defining a world as a function from primitive expressions to suitable values as we did in $\mathcal{OL}$, we define a world in $\mathcal{AOL}$ as a function from primitive expressions and sequences of actions to these values. We then define a situation as a pair consisting of a world and a sequence of actions.

Worlds and situations

More precisely, the standard names for objects, as already mentioned, are ground terms involving just $\mathtt{0}$ and *succ*; the standard names for actions are terms of the form $A(t_1, \ldots, t_n)$ where A is an action function and each t_i is a standard name; there are no standard names for situations, since there will be more situations than expressions in the language. The primitive terms $\mathcal{T}$ are object terms of the form $f(t_1, \ldots, t_n)$ where each t_i is a standard name, and $f(x_1, \ldots, x_n, s)$ is a functional fluent. The primitive formulas $\mathcal{P}$ are atoms of the form $F(t_1, \ldots, t_n)$ where each t_i is a standard name, and $F(x_1, \ldots, x_n, s)$ is a relational fluent, or F is one of *Poss* or *SF*. Note that except for *Poss* and *SF*, primitive expressions are all fluents with the situation argument suppressed.

Let Act^* be the set of all sequences of standard names for actions including the empty sequence ϵ.

Definition 4.1: An $\mathcal{AOL}$ *world* w is a function:

$$w : (\mathcal{P} \times Act^*) \ \cup \ (\mathcal{T} \times Act^*) \longrightarrow \{0, 1\} \cup \mathcal{N}$$

such that

$w[p, \vec{a}] \in \{0, 1\}$ for all $p \in \mathcal{P}$.
$w[t, \vec{a}] \in \mathcal{N}$ for all $t \in \mathcal{T}$.

Let $\mathcal{W}$ denote the set of all $\mathcal{AOL}$ worlds.

Definition 4.2: An $\mathcal{AOL}$ *situation* is a pair $(w, \vec{a})$, where $w \in \mathcal{W}$ and $\vec{a} \in Act^*$. An *initial* situation is one where $\vec{a} = \epsilon$.

Definition 4.3: An action model M is a pair $\langle e, w \rangle$, where $w \in \mathcal{W}$ and $e \subseteq \mathcal{W}$.

As in $\mathcal{OL}$, w is taken to specify the actual world, and e specifies the epistemic state as those worlds an agent has not yet ruled out as being the actual one. As we will see below, a situation term s will be interpreted semantically as an $\mathcal{AOL}$ situation $(w, \vec{a})$, consisting of a world and a sequence of actions that have happened so far. A fluent $p(s)$ will be considered true if $w[p, \vec{a}] = 1$.

Because situations cannot have standard names, to interpret formulas with variables, we need to use variable maps. A variable map ν maps object, action, and situation variables into standard names for objects and actions, and into $\mathcal{AOL}$ situations, respectively. In addition, ν assigns relations of the appropriate type[8] to relational variables. For a given ν, ν_o^x denotes the variable map which is like ν except that x is mapped into o.

The meaning of terms

We write $|\cdot|_{M,\nu}$ for the denotation of terms with respect to an action model $M = \langle e, w \rangle$ and a variable map ν. Then

$|\mathtt{0}|_{M,\nu} = \mathtt{0}$;
$|succ(t)|_{M,\nu} = succ(|t|_{M,\nu})$;
$|f(\vec{t}, t_s)|_{M,\nu} = w'[f(|\vec{t}|_{M,\nu}), \vec{a}]$, where $f(\vec{t}, t_s)$ is a functional fluent, and $|t_s|_{M,\nu} = (w', \vec{a})$;
$|A(\vec{t})|_{M,\nu} = A(|\vec{t}|_{M,\nu})$, where $A(\vec{t})$ is an action term;
$|S_0|_{M,\nu} = (w, \epsilon)$;
$|do(t_a, t_s)|_{M,\nu} = (w', \vec{a} \cdot a)$, where $|t_s|_{M,\nu} = (w', \vec{a})$, and $|t_a|_{M,\nu} = a$;
$|x|_{M,\nu} = \nu(x)$, where x is any variable, including predicate variables.

Observe that in a model $M = \langle e, w \rangle$, the only way to refer to a situation that does not use the given world w is to use a situation variable.

The meaning of formulas

We write $M, \nu \models \alpha$ to mean formula α comes out true in action model M and variable map ν:

$M, \nu \models F(\vec{t}, t_s)$ iff $w'[F(|\vec{t}|_{M,\nu}), \vec{a}] = 1$, where $F(\vec{t}, t_s)$ is a relational fluent, and $|t_s|_{M,\nu} = (w', \vec{a})$;
$M, \nu \models X(\vec{t})$ iff $|\vec{t}|_{M,\nu} \in \nu(X)$, where X is a relational variable;
$M, \nu \models Poss(t_a, t_s)$ iff $w'[Poss(|t_a|_{M,\nu}), \vec{a}] = 1$, where $|t_s|_{M,\nu} = (w', \vec{a})$;
$M, \nu \models SF(t_a, t_s)$ iff $w'[SF(|t_a|_{M,\nu}), \vec{a}] = 1$, where $|t_s|_{M,\nu} = (w', \vec{a})$;
$M, \nu \models K_0(t_s)$ iff $|t_s|_{M,\nu} = (w', \epsilon)$ and $w' \in e$;
$M, \nu \models t_1 = t_2$ iff $|t_1|_{M,\nu} = |t_2|_{M,\nu}$;
$M, \nu \models \neg\alpha$ iff $M, \nu \not\models \alpha$;
$M, \nu \models \alpha \vee \beta$ iff $M, \nu \models \alpha$ or $M, \nu \models \beta$;
$M, \nu \models \forall x.\alpha$ iff $M, \nu_o^x \models \alpha$ for all o of the appropriate sort (object, action, situation, relation).

[7] In some applications this generality will not be required.

[8] The type determines the arity and the sort of each argument of the relations the variable ranges over. Since, in our examples, the type will always be obvious from the context, we leave this information implicit.

For sentences α we sometimes write $M \models \alpha$ instead of $M, \nu \models \alpha$.

Validity is defined in the usual way as truth in all models, that is, a formula α is valid in $\mathcal{AOL}$ ($\models_{\text{AOL}} \alpha$) iff for all action models $M = \langle e, w \rangle$ and variable maps ν, $M, \nu \models \alpha$.

4.2 An axiomatization

The first three axioms tell us that the set of objects is isomorphic to the set $\mathcal{N}$ of standard object names. Indeed the formulation resembles the usual second-order definition of the natural numbers, that is, the following three axioms do no more than give us domain closure and unique names axioms for objects.

F1: $\forall x.\ succ(x) \neq \mathsf{0}$
F2: $\forall x, y.\ succ(x) = succ(y) \supset x = y$
F3: $\forall P.\ [P(\mathsf{0}) \wedge \forall x(P(x) \supset P(succ(x)))] \supset \forall x.P(x)$

Next we need to say that the actions consist precisely of the primitive actions and that they are all distinct. This can be done in the usual way.

F4: Domain closure and unique names axioms for actions.

Neither *SF* nor *Poss* need special axioms since their meaning is left completely user-defined. The only foundational axiom concerning K_0 is one saying that it only applies to initial situations. To be precise, let

$$Init(s') \doteq \neg\exists a, s.s' = do(a, s).$$

Then we have

F5: $Init(S_0) \wedge \forall s.K_0(s) \supset Init(s)$

Finally we have the job of characterizing the set of situations. As in the original dialect of the situation calculus, we first want to say that any non-initial situation is the result of applying *do* to an initial situation. We use variants of the previous axioms:

F6: $\forall a_1, a_2, s_1, s_2.\ do(a_1, s_1) = do(a_2, s_2) \supset (a_1 = a_2 \wedge s_1 = s_2)$.
F7: $\forall P.\ (\forall s, s'.Init(s) \wedge s \preceq s' \supset P(s')) \supset \forall s.P(s)$

where we have

$$s \preceq s' \doteq \forall R[\ldots \supset R(s, s')]$$

with the ellipsis standing for the conjunction of

$$\forall s_1.\ R(s_1, s_1)$$
$$\forall a, s_1.\ R(s_1, do(a, s_1))$$
$$\forall s_1, s_2, s_3.\ R(s_1, s_2) \wedge R(s_2, s_3) \supset R(s_1, s_3)$$

Then the only remaining job is to characterize the set of initial situations. Looking back at the semantics of $\mathcal{AOL}$, recall that for a correct interpretation of only knowing, we had to insist that there be an initial situation corresponding to any conceivable outcome of the fluents initially and after any sequence of actions. Given the power of second-order logic, it is possible to precisely capture this property axiomatically.[9]

To handle sequences of actions, we begin by introducing an abbreviation $C(s', s)$ intended to say that s' and s involve the same sequence of actions from perhaps different initial states:

$$C(s', s) \doteq \forall R[\ldots \supset R(s', s)]$$

where the ellipsis stands for the conjunction of

$$\forall s_1, s_2.\ Init(s_1) \wedge Init(s_2) \supset R(s_1, s_2)$$
$$\forall a, s_1, s_2.\ R(s_1, s_2) \supset R(do(a, s_1), do(a, s_2)).$$

With this in place, we can use situations s where $S_0 \preceq s$ as a canonical way of talking about sequences of actions.

Suppose that our language contains relational fluents $F_1, \ldots, F_n$.[10] Then we can write our final axiom as:

F8: $\forall s.Init(s) \equiv \forall Q[\ldots \supset Q(s)]$

where the ellipsis stands for

$$\begin{array}{l}
\forall P_1, \ldots, P_{n+2}, \exists s'.\ Q(s') \wedge Init(s') \wedge \\
\quad \forall \vec{x}_1, \ldots, \vec{x}_n, t, u, a. \\
\quad\quad s' \preceq t \wedge S_0 \preceq u \wedge C(t, u) \supset \\
\quad\quad\quad F_1(\vec{x}_1, t) \equiv P_1(\vec{x}_1, u) \wedge \ldots \wedge \\
\quad\quad\quad F_n(\vec{x}_n, t) \equiv P_n(\vec{x}_n, u) \wedge \\
\quad\quad\quad Poss(a, t) \equiv P_{n+1}(a, u) \wedge \\
\quad\quad\quad SF(a, t) \equiv P_{n+2}(a, u)
\end{array}$$

To see how this axiom works, imagine that $n = 1$, $F_1(x, s)$ is a unary fluent, and ignore *Poss* and *SF*. Then, the set of initial situations is the least set such that for every mapping from sequences of actions (here represented by the u) to sets of objects, there is an initial situation s' such that F_1 holds on exactly that set of objects in the situation t that results from doing those actions starting in s'.

[9] We are indebted to Fangzhen Lin who pointed this out to us.

[10] Recall that apart from the special predicates K_0, *Poss* and *SF*, our language has only finitely many relational and functional fluents. To make things simple, we omit functional fluents completely from this axiom. They require functional variables in the language.

These are all the axioms we need, and we will refer to them collectively as AX from now on, and we let $\mathrm{AX} \models \alpha$ stand for "α is logically implied by AX" in ordinary (second-order) logic.

It is not hard to show that the axioms are sound with respect to the semantics of $\mathcal{AOL}$. Moreover, action models are, in a sense, the only models of the axioms. More precisely, one can show that for any arbitrary Tarskian model I of the axioms there is an action model M such that I and M agree on all sentences. The key property is that the objects, actions, and situations of an arbitrary model of AX are isomorphic to the objects, actions, and situations, respectively, of action models. With that we obtain the main result.

Theorem 4.4: *For any* α, $\models_{AOL} \alpha$ *iff* $\mathrm{AX} \models \alpha$.[11]

Given this result one may wonder what the point of introducing a non-standard semantics for $\mathcal{AOL}$ is in the first place, that is, why not just use the axioms? Perhaps the strongest argument in favor of the semantics is that it lends independent support to the claim that the axioms are indeed reasonable. The fact that the semantics generalizes that of $\mathcal{OL}$ in a natural way adds further credence to that claim.

5 Knowledge and Action

Before going into details about the connection between knowledge and action in $\mathcal{AOL}$, a few words are in order about how we should envisage using $\mathcal{AOL}$ to model a particular domain of interest.

Instead of simply writing a basic action theory as presented in Section 2 describing what holds in S_0 and after performing actions, the user must now worry about the other initial situations.

Consider, for example, a precondition axiom for an action A. In the ordinary situation calculus, the user would write an axiom of the form

$$\forall s.\ \mathit{Poss}(A, s) \equiv \phi_a(s).$$

where ϕ_a is some formula does not mention *Poss*. The intent of the quantification (given the previous foundational axioms) was for this to hold in S_0 and all its successors. But in $\mathcal{AOL}$, the quantification over *all* situations is much too strong. By virtue of **F8** there will be initial situations where *Poss* and ϕ_a have different truth values, and so the axiom as it stands is false! To achieve the desired effect, the user should write instead

$$\forall s'.\ S_0 \preceq s' \supset [\mathit{Poss}(A, s') \equiv \phi_a(s')].$$

[11]For reasons of space proofs are generally omitted and deferred to a longer version of this paper [LL98].

to ensure that the precondition applies to S_0 and its successors. It is then a separate step to assert that the precondition is known, if desired. To do so, the user would write

$$\forall s, s'.\ K_0(s) \wedge s \preceq s' \supset [\mathit{Poss}(A, s') \equiv \phi_a(s')].$$

This ensures that the axiom is considered to hold in any situation initially considered possible and all of its successors. Similar considerations apply to writing successor state axioms. In general, we write initial state axioms, precondition axioms, and successor state axioms all parameterized by the initial situation we wish to consider, and only quantify over successors of that initial situation. We will see an example shortly.

Given a specification of what is known in S_0, the predicates *SF* and *Poss* are then used to characterize what is known in successor situations. Note that the logic itself imposes no constraints on either *SF* or *Poss*; it is up to the user in an application to write appropriate axioms. For *Poss*, these are the precondition axioms; for *SF*, the user must write *sensed fluent axioms*, one for each action type, as discussed in [Lev96]. The idea is that $\mathit{SF}(a, s)$ gives the condition sensed by action a in situation s. So we might have, for example,

$$\mathit{SF}(\mathsf{sonar}, s) \equiv (\mathit{wdist}(s) < 10)$$

as a way of saying that the sonar sensing action in situation s tells the robot whether or not the distance to the wall in s is less than 10 units. In case the action a has no sensing component (as in simple physical actions, like moving), the axiom should state that $\mathit{SF}(a, s)$ is identically TRUE. Having defined *SF* as a predicate, we essentially confine ourselves to sensing truth values. If we want the result of a sense action to be the value of a term such as a sonar measuring the actual distance to the wall, we can do so by simply redefining *SF* as a function and treating TRUE and FALSE as special values returned by *SF*. To keep the presentation simple, however, we ignore this issue here.

With these terms, we can now define $K(s', s)$ as an abbreviation for a formula that characterizes when a situation s' is accessible from an arbitrary situation s:[12]

$$K(s', s) \doteq \forall R[\ldots \supset R(s', s)]$$

where the ellipsis stands for the conjunction of

$$\begin{array}{l} \forall s_1, s_2.\ Init(s_1) \wedge Init(s_2) \wedge K_0(s_2) \supset R(s_2, s_1) \\ \forall a, s_1, s_2.\ R(s_2, s_1) \ \wedge\ (\mathit{SF}(a, s_2) \equiv \mathit{SF}(a, s_1)) \ \wedge \\ \qquad (\mathit{Poss}(a, s_2) \equiv \mathit{Poss}(a, s_1)) \ \supset \\ \qquad\quad R(do(a, s_2), do(a, s_1)). \end{array}$$

[12]We could have defined K, as well as $\preceq$ and C, as a predicate in the language as is usually done, but we have chosen not to simply because we wanted to keep the formal apparatus as small as possible.

Space precludes a detailed analysis of this definition, except to claim that it satisfies the successor state axiom for a predicate K proposed in [SL93] as a solution to the frame problem for knowledge and later reformulated in [Lev96], whose notation we follow here:

Theorem 5.1: *The following is a theorem of $\mathcal{AOL}$:*

$$\begin{aligned}\forall a, s, s'.\ \mathit{Poss}(a, s) \supset K(s', do(a, s)) \equiv \\ \exists s''.\ s' = do(a, s'') \wedge K(s'', s) \wedge \mathit{Poss}(a, s'') \\ \wedge\ [\mathit{SF}(a, s) \equiv \mathit{SF}(a, s'')].\end{aligned}$$

Given K, knowledge can then be defined in a way similar to possible-world semantics [Kri63, Hin62, Moo85a] as truth in all accessible situations. Similarly, only knowing a sentence α at a situation s means that all and only those situations with the same action history as s are accessible. We denote the two forms of knowledge using the following macros, where α may contain the special situation symbol now. Let α_s^{now} refer to α with all occurrences of now replaced by s. Then

$$\begin{aligned}\texttt{Knows}(\alpha, s) &\doteq \forall s' K(s', s) \supset \alpha_{s'}^{now} \\ \texttt{OKnows}(\alpha, s) &\doteq \forall s' C(s', s) \supset (K(s', s) \equiv \alpha_{s'}^{now}).\end{aligned}$$

For example, $\texttt{Knows}(Broken(x, now), S_0)$ stands for $\forall s K(s, S_0) \supset Broken(x, s)$ and should be read as "the agent knows in S_0 that x is (now) broken."

6 An Example

We now turn to an example showing how knowing and only knowing can be combined with actions in $\mathcal{AOL}$.

Imagine a robot that lives in a 1-dimensional world, and that can move towards or away from a fixed wall. The robot also has a sonar sensor that tells it when it gets too close to the wall, say, less than 10 units away. So we might imagine three actions, adv and rev which move the robot one unit towards and away from the wall, and a sonar sensing action. We have a single fluent, $\mathit{wdist}(s)$, which gives the actual distance from the robot to the wall in situation s.

We begin by defining precondition axioms, sensed fluent axioms and successor state axioms, all parameterized by some initial situation s (as discussed in Section 5). Let $\mathit{ALL}(s)$ stand for the conjunction of these formulas:

$$\begin{aligned}&\forall s'.s \preceq s' \supset \mathit{Poss}(\mathsf{adv}, s') \equiv \mathit{wdist}(s') > 0 \\ &\forall s'.s \preceq s' \supset \mathit{Poss}(\mathsf{rev}, s') \equiv \text{TRUE} \\ &\forall s'.s \preceq s' \supset \mathit{Poss}(\mathsf{sonar}, s') \equiv \text{TRUE} \\ &\forall s'.s \preceq s' \supset \mathit{SF}(\mathsf{adv}, s') \equiv \text{TRUE} \\ &\forall s'.s \preceq s' \supset \mathit{SF}(\mathsf{rev}, s') \equiv \text{TRUE} \\ &\forall s'.s \preceq s' \supset \mathit{SF}(\mathsf{sonar}, s') \equiv \mathit{Close}(s')\end{aligned}$$

$$\begin{aligned}&\forall s'.s \preceq s' \supset \\ &\quad \forall a.\ \mathit{wdist}(do(a, s')) = z \equiv \\ &\qquad a = \mathsf{adv} \wedge z = \mathit{wdist}(s') - 1 \\ &\qquad \vee\ \ a = \mathsf{rev} \wedge z = \mathit{wdist}(s') + 1 \\ &\qquad \vee\ \ z = \mathit{wdist}(s') \wedge a \neq \mathsf{adv} \wedge a \neq \mathsf{rev}\end{aligned}$$

The formula $\mathit{Close}(s)$ in the above is an abbreviation:

$$\mathit{Close}(s) \doteq \mathit{wdist}(s) < 10.$$

Now we are ready to consider some specifics having to do with what is true initially. Assume the robot is located initially 6 units away from the wall in S_0. For simplicity, we also assume that all of the axioms above are true in S_0, that they are also known in S_0, and this is all that is known. So let Γ_0 be the conjunction of the axioms in AX and:

$$\mathit{wdist}(S_0) = 6 \wedge \mathit{ALL}(S_0) \wedge \texttt{OKnows}(\mathit{ALL}(now), S_0)$$

With these in hand,[13] we are ready to establish some properties of the relationship between knowledge and action.

1. After reading its sonar sensor, the robot knows that it is close to the wall:

   $$\Gamma_0 \models \texttt{Knows}(\mathit{Close}, do(\mathsf{sonar}, S_0))$$

   The proof is as follows: Suppose $M \models \Gamma_0$. Further suppose that $M, \nu \models K(s, do(\mathsf{sonar}, S_0))$. By Theorem 5.1, we get that

   $$\begin{aligned}M, \nu \models \exists s'.s = do(\mathsf{sonar}, s') \\ \wedge\ K_0(s') \wedge [\mathit{Close}(s') \equiv \mathit{Close}(S_0)].\end{aligned}$$

   Since $M \models \mathit{Close}(S_0)$ and

   $$M \models \forall s.K_0(s) \supset (\mathit{wdist}(do(\mathsf{sonar}, s)) = \mathit{wdist}(s))$$

   by virtue of Γ_0, we get that $M, \nu \models \mathit{Close}(s)$. Thus, we have that

   $$M \models \forall s.K(s, do(\mathsf{sonar}, S_0)) \supset \mathit{Close}(s).$$

2. Before reading its sensor, however, the robot does not know if it is close to the wall:

   $$\Gamma_0 \models \neg\texttt{Knows}(\mathit{Close}, S_0)$$

   The proof is as follows: Suppose $M \models \Gamma_0$. Let w' be the element of $\mathcal{W}$ which is just like w except that $w'[\mathit{wdist}, \epsilon] = 10$. Then, when $\nu(s) = (w', \epsilon)$, we have that $M, \nu \models \neg\mathit{Close}(s)$ and $M, \nu \models \mathit{ALL}(s)$. From the latter and the fact that

   $$M \models \forall s.Init(s) \supset (K_0(s) \equiv \mathit{ALL}(s)),$$

   it follows that $M, \nu \models K_0(s)$. Consequently,

   $$M \models \neg\forall s.K(s, S_0) \supset \mathit{Close}(s).$$

[13] Strictly speaking, we also need axioms for basic arithmetic as needed by the example. For simplicity we ignore this complication here.

3. After reading its sensor and moving closer to the wall, the robot continues to know that the wall is close:

$$\Gamma_0 \models \texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{adv}, \mathit{do}(\mathsf{sonar}, S_0)))$$

The proof is as follows: Suppose $M \models \Gamma_0$. Further suppose that $M, \nu \models K(s, \mathit{do}(\mathsf{adv}, \mathit{do}(\mathsf{sonar}, S_0)))$. By Theorem 5.1, we get that

$$M, \nu \models \exists s'.s = \mathit{do}(\mathsf{adv}, \mathit{do}(\mathsf{sonar}, s')) \wedge K_0(s') \\ \wedge [\mathit{Close}(s') \equiv \mathit{Close}(S_0)].$$

Since $M \models \mathit{Close}(S_0)$ and

$$M \models \forall s.K_0(s) \supset \\ (\mathit{wdist}(\mathit{do}(\mathsf{adv}, \mathit{do}(\mathsf{sonar}, s))) = \mathit{wdist}(s) - 1)$$

by virtue of Γ_0, we get that $M, \nu \models \mathit{Close}(s)$. Thus, we have that

$$M \models \forall s.K(s, \mathit{do}(\mathsf{adv}, \mathit{do}(\mathsf{sonar}, S_0))) \supset \\ \mathit{Close}(s).$$

4. After reading its sensor and moving away from the wall, the robot is still close to the wall, but no longer knows it:

$$\Gamma_0 \models \mathit{Close}(\mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0))) \wedge \\ \neg\texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0))).$$

The proof is as follows: Suppose $M \models \Gamma_0$. To show that $M \models \mathit{Close}(\mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0)))$, we need only observe that because of Γ_0, we have that

$$M \models \mathit{wdist}(\mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0))) = 7.$$

To show that

$$M \models \neg\texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0))),$$

we begin by letting w' be the element of $\mathcal{W}$ this is just like w except that $w'[\mathit{wdist}, \epsilon] = 9$. Then, when $\nu(s) = (w', \epsilon)$, we have that $M, \nu \models \mathit{Close}(s)$, and so because of Γ_0, we get that $M, \nu \models K_0(s)$. It follows that

$$M, \nu \models K(\mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, s)), \\ \mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0)))$$

and moreover, that

$$M, \nu \models \mathit{wdist}(\mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, s))) = 10.$$

Thus, we have that

$$M, \nu \models \exists s.K(s, \mathit{do}(\mathsf{rev}, \mathit{do}(\mathsf{sonar}, S_0))) \wedge \\ \neg\mathit{Close}(s).$$

5. As for knowledge of the future, we have that the robot knows initially that after it reads its sonar, it will know whether or not it is close to the wall:

$$\Gamma_0 \models \texttt{Knows}([\texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{sonar}, \mathit{now})) \vee \\ \texttt{Knows}(\neg\mathit{Close}, \mathit{do}(\mathsf{sonar}, \mathit{now}))], S_0)$$

The proof is as follows: Suppose $M \models \Gamma_0$. Suppose further that $M, \nu \models K_0(s)$. There are two cases to consider: suppose that $M, \nu \models \mathit{Close}(s)$. Then, by an argument similar to the one for (1) above, we get

$$M, \nu \models \texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{sonar}, s));$$

if on the other hand, $M, \nu \models \neg\mathit{Close}(s)$, by a similar argument we get that

$$M, \nu \models \texttt{Knows}(\neg\mathit{Close}, \mathit{do}(\mathsf{sonar}, s)).$$

Either way, we have that

$$M \models \forall s.\ K_0(s) \supset [\texttt{Knows}(\mathit{Close}, \mathit{do}(\mathsf{sonar}, s)) \vee \\ \texttt{Knows}(\neg\mathit{Close}, \mathit{do}(\mathsf{sonar}, s))]$$

6. As for knowledge of the past, we have for example that after moving closer to the wall, the robot knows that it was at least 1 unit away just before doing that action:

$$\Gamma_0 \models \texttt{Knows}(\exists s'[\mathit{now} = \mathit{do}(\mathsf{adv}, s') \wedge \\ \mathit{wdist}(s') > 0], \mathit{do}(\mathsf{adv}, S_0))$$

The proof is as follows: Suppose $M \models \Gamma_0$. Suppose further that $M, \nu \models K(s, \mathit{do}(\mathsf{adv}, S_0))$. Then by Theorem 5.1, we have that

$$M, \nu \models \exists s'.s = \mathit{do}(\mathsf{adv}, s') \wedge K_0(s') \wedge \mathit{Poss}(s').$$

From Γ_0, we get that

$$M \models \forall s.K_0(s) \supset [\mathit{Poss}(\mathsf{adv}, s) \equiv (\mathit{wdist}(s) > 0)].$$

Thus, $M, \nu \models \exists s'.s = \mathit{do}(\mathsf{adv}, s') \wedge (\mathit{wdist}(s') > 0)$. So we have that

$$M \models \forall s.K(s, \mathit{do}(\mathsf{adv}, S_0)) \supset \\ [\exists s'.\ s = \mathit{do}(\mathsf{adv}, s') \wedge (\mathit{wdist}(s') > 0)].$$

7 Embedding $\mathcal{OL}$ in $\mathcal{AOL}$

In this section we show that $\mathcal{AOL}$ is a faithful extension of $\mathcal{OL}$ in the following sense. It is possible to translate every sentence α of $\mathcal{OL}$ into a sentence $\alpha[s]$ of $\mathcal{AOL}$, where s is any situation, such that α is valid in $\mathcal{OL}$ iff $\alpha[S_0]$ is a logical consequence of the axioms. The following translation is essentially the same as the one proposed in [Lak96].

Definition 7.1: Given any term or formula ϕ in $\mathcal{OL}$, the corresponding term or formula $\phi[s]$ in $\mathcal{AOL}$, where s is any situation term, is defined as follows.

First, we let * denote the obvious translation from the standard names of $\mathcal{OL}$ into those of $\mathcal{AOL}$ involving $\mathtt{0}$ and *succ*. For example, $^{\#}3^* = succ(succ(succ(\mathtt{0})))$.

$$
\begin{array}{l}
x[s] = x \text{ if } x \text{ is a variable} \\
n[s] = n^* \text{ if } n \text{ is a standard name} \\
f(t_1, \ldots, t_n)[s] = f(t_1[s], \ldots, t_n[s], s) \\
\quad \text{if } f(\vec{t}) \text{ is a term in } \mathcal{OL} \\
F(t_1, \ldots, t_n)[s] = F(t_1[s], \ldots, t_n[s], s) \\
\quad \text{if } P(\vec{t}) \text{ is an atomic formula in } \mathcal{OL} \\
(t_1 = t_2)[s] = (t_1[s] = t_2[s]) \\
(\neg\alpha)[s] = \neg\alpha[s] \\
(\alpha \vee \beta)[s] = \alpha[s] \vee \beta[s] \\
(\forall x\alpha)[s] = \forall x\alpha[s] \\
(\boldsymbol{K}\alpha)[s] = \mathtt{Knows}(\alpha[now], s) \\
(\boldsymbol{O}\alpha)[s] = \mathtt{OKnows}(\alpha[now], s)
\end{array}
$$

For example, let $\alpha = \boldsymbol{O}P(a) \supset \neg\exists x\boldsymbol{K}P(x)$. Then

$$
\alpha[S_0] = [\forall s C(s, S_0) \supset (K(s, S_0) \equiv P(a(s), s))] \supset \neg\exists x(\forall s K(s, S_0) \supset P(x, s)).
$$

Note that we tacitly assume that for each predicate and function symbol in α there is a corresponding fluent of the same name in $\mathcal{AOL}$. Since we are applying the translation only to sentences, one at a time, there is no problem in making this assumption[14]

The embedding of $\mathcal{OL}$ into $\mathcal{AOL}$ is established, roughly, by proving that for every $\mathcal{OL}$ model M there is an "equivalent" action model M', and vice versa. Here "equivalent" means that M satisfies α iff M' satisfies $\alpha[S_0]$ for any α mentioning only fluents in $\mathcal{AOL}$. We denote the set of all fluents of $\mathcal{AOL}$ as $\mathcal{F}$. (Note that *Poss* is not part of $\mathcal{F}$.)

Let us consider informally how to construct, given an $\mathcal{OL}$ model $M = (e, w)$, an equivalent action model $M' = \langle e', w' \rangle$. First, notice that the truth value of $\alpha[S_0]$ is determined by initial situations only. This means that, when mapping M into M', we can simply ignore all actions and non-initial situations. The mapping from M to M' then, roughly, amounts to the following. Let $\overline{e}$ denote the complement of e. Then for each $w^* \in e$ ($\overline{e}$) make sure that e' ($\overline{e'}$) contains all those $\mathcal{AOL}$ worlds w' whose initial situations agree with w^* on $\mathcal{F}$. There is only one complication. It may be the case that there are $\mathcal{OL}$ worlds w_1 and w_2 such that $w_1 \in e$ and $w_2 \in \overline{e}$ and both agree on $\mathcal{F}$. In this case we need to split the $\mathcal{AOL}$ worlds whose initial situations agree with w_1 and w_2 on $\mathcal{F}$ into two nonempty sets and assign them to e' and $\overline{e'}$, respectively. Such a split is always possible and it can easily be arranged based on the truth value of a predicate not occurring in $\mathcal{F}$ (such as *Poss*).

Similarly, one can also show the converse, namely that for every action model M there is an equivalent $\mathcal{OL}$ model M' such that M' satisfies α iff M satisfies $\alpha[S_0]$.

This construction, together with the fact that validity in $\mathcal{AOL}$ is completely characterized by the axioms AX (Theorem 4.4), then leads to the desired embedding of $\mathcal{OL}$.

Theorem 7.2: *Let α be a sentence in $\mathcal{OL}$. Then α is valid in $\mathcal{OL}$ iff $\alpha[S_0]$ is a logical consequence of* AX.

This result then provides us, for the first time, with an axiomatic characterization of the valid sentences of $\mathcal{OL}$.

The careful reader will have noticed that Axiom **F8** needs to vary depending on α, since **F8** must mention explicitly at least all those fluents occurring in α. In the full paper we will show that it is possible to have a fixed axiom system for all sentences of $\mathcal{OL}$ by encoding the infinitely many predicate and function symbols of $\mathcal{OL}$ using only finitely many fluents in $\mathcal{AOL}$.

Apart from this technical issue, it should be noted that the price of the axiomatization of $\mathcal{OL}$ is high in that we need to appeal to second-order logic. Whether there is a first-order axiom system for $\mathcal{OL}$ remains an open question. Note, however, that Halpern and Lakemeyer [HL95] have recently shown that, even if one exists, it cannot be recursive.[15] So chances are that we may have to settle for second-order.

8 Comparison with Lakemeyer's $\mathcal{OLS}$:

As already mentioned, Lakemeyer [Lak96] has also proposed an amalgamation of only knowing and the situation calculus. There are obvious differences between his logic $\mathcal{OLS}$ and $\mathcal{AOL}$. For one, $\mathcal{OLS}$ considers real knowledge rather than belief, that is, whatever is believed in $\mathcal{OLS}$ is also true in S_0. For another, while $\mathcal{OLS}$ has a formal semantics, there is no axiomatization. In addition, there are deeper differences as well, which give rise to anomalies when reasoning about only knowing in $\mathcal{OLS}$ which are not present in $\mathcal{AOL}$.

To start with, only knowing in $\mathcal{OLS}$ does have reasonable properties quite similar to $\mathcal{AOL}$ if it is confined to sentences of the form $\mathtt{OKnows}(\alpha(now), t_s)$, where α contains

[14] There would be a problem if we were to apply the translation to infinite sets of sentences, since $\mathcal{AOL}$ is restricted to finitely many fluents, while $\mathcal{OL}$ is not. We will have a bit more to say about how to deal with this mismatch at the end of this section.

[15] There is a trivial nonrecursive "axiom system", which is simply the set of all valid sentences of $\mathcal{OL}$. The interesting question, of course, is whether there is a system with a finite set of axiom schemas.

no situation terms other than now and t_s is a closed situation term. Intuitively, here we are looking only at what is known about one particular situation, which is very similar to $\mathcal{OL}$ except that we can now ask queries about successor situations as well.

Problems arise when we allow arbitrary sentences as arguments of only knowing in $\mathcal{OLS}$, in particular those where we talk about more than one situation as in precondition axioms. The difficulty is that $\mathcal{OLS}$ models have, in a sense, far fewer situations than there are in $\mathcal{AOL}$. In particular, in $\mathcal{OLS}$ there are only as many initial situations as there are different valuations of the fluents (with the situation argument suppressed). This means that any two distinct initial situations in $\mathcal{OLS}$ must differ in the value of at least one primitive expression. In other words, it is impossible to represent within an $\mathcal{OLS}$ model two different courses of events which have completely identical initial situations and only diverge after some actions have been performed. In addition, *Poss* is handled as a function from situations and actions into $\{0, 1\}$, which further restricts the range of possibilities. For example, there are $\mathcal{OLS}$ models where no actions are possible anywhere.

Without going into any further detail of the formalism, let us consider again the example of Section 6. It is possible to construct an $\mathcal{OLS}$ model M such that M satisfies $\mathit{ALL}(S_0) \wedge \mathit{wdist}(S_0) = 6$, M does not satisfy $\mathit{ALL}(s)$ for any other initial situation s (for example, by choosing $\mathit{Poss}(\mathsf{rev}, s)$ to be false), and S_0 is the only situation epistemically accessible from S_0. Then M satisfies $\mathtt{OKnows}(\mathit{ALL}(now), S_0)$ since S_0 is the only initial situation where *ALL* holds. In addition, M also satisfies $\mathtt{Knows}(\mathit{Close}, S_0)$ since the distance to the wall is 6 at S_0 and there are no other epistemic alternatives. This is clearly unintuitive since the robot has neither been told what its position is nor has it read its sensors. Moreover, the problem arises precisely because there are not enough situations. In particular, there are no initial situations where *ALL* holds and the distance to the wall is greater than 9.

We believe that $\mathcal{AOL}$ fixes the problems of $\mathcal{OLS}$ in just the right way.

9 Conclusion

In summary, we have introduced the logic $\mathcal{AOL}$ which amalgamates both the situation calculus and the logic of only knowing $\mathcal{OL}$. Besides a semantics we have provided a sound and complete set of axioms. $\mathcal{AOL}$ is compatible with earlier work on knowledge and action and improves on a previous approach to only knowing in the situation calculus. By way of examples we demonstrated that $\mathcal{AOL}$ allows us to make distinctions which are intuitive and, as far as we know, cannot be handled by other formalisms. Finally, as a side-benefit we obtained a complete axiomatization of $\mathcal{OL}$.

$\mathcal{AOL}$ should be understood as a specification of only knowing within a theory of action. We do not expect $\mathcal{AOL}$ to be implemented in its full generality. On the other hand, note that the second-order situation calculus forms the basis of the high-level control language GOLOG [LRL97], which is used in real robots as in [BCF98]. We believe that $\mathcal{AOL}$ has a role to play in future extensions of GOLOG. Other future work includes extensions of $\mathcal{AOL}$ itself such as a generalization to the multi-agent case.

Acknowledgements

We are indebted to Fangzhen Lin for his idea how to axiomatically characterize the set of all situations in second-order logic.

References

[BCF98] Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., The Interactive Museum Tour-Guide Robot, to appear: *AAAI-98*.

[dVS94] del Val, A. and Shoham, Y., A Unified View of Belief Revision and Update. *Journal of Logic and Computation* Special Issue on Actions and Processes, **4**, 1994, pp. 797–810.

[HL95] Halpern, J. Y. and Lakemeyer, G., Levesque's Axiomatization of Only Knowing is Incomplete. *Artificial Intelligence* **74**(2), 1995, pp. 381–387.

[HM92] Halpern, J. Y. and Moses, Y. O., A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief. *Artificial Intelligence*, **54**, 1992, pp. 319–379.

[Hin62] Hintikka, J., *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.

[HC68] Hughes, G. E. and Cresswell, M. J., *An Introduction to Modal Logic*, Methuen and Company Ltd., London, England, 1968.

[Kap71] Kaplan, D., Quantifying In, in L. Linsky (ed.), *Reference and Modality*, Oxford University Press, Oxford, 1971.

[Kri63] Kripke, S. A., Semantical considerations on modal logic. *Acta Philosophica Fennica* **16**, 1963, pp. 83–94.

[Lak95] A Logical Account of Relevance, *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Morgan Kaufmann, 1995, pp. 853–859.

[Lak96] Lakemeyer, G., Only Knowing in the Situation Calculus, *Proc. of the Fifth International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Francisco, 1996, pp. 14–25.

[LL98] Lakemeyer, G. and Levesque, H. J. $\mathcal{AOL}$: a logic of acting, sensing, knowing, and only knowing. In preparation.

[Lev84] Levesque, H. J., Foundations of a Functional Approach to Knowledge Representation, *Artificial Intelligence*, **23**, 1984, pp. 155-212.

[Lev90] Levesque, H. J., All I Know: A Study in Autoepistemic Logic. *Artificial Intelligence*, North Holland, **42**, 1990, pp. 263–309.

[Lev96] Levesque, H. J., What is Planning in the Presence of Sensing. AAAI-96, AAAI Press, 1996.

[LL9x] Levesque, H. J. and Lakemeyer, G., *The Logic of Knowledge Bases*, Monograph, forthcoming.

[LRL97] 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**, 59-84, 1997.

[LR94] Lin, F. and Reiter, R., State constraints revisited. *J. of Logic and Computation*, *special issue on actions and processes*, **4**, 1994, pp. 665–678.

[McC63] McCarthy, J., *Situations, Actions and Causal Laws.* Technical Report, Stanford University, 1963. Also in M. Minsky (ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, 1968, pp. 410–417.

[Moo85a] Moore, R. C., A Formal Theory of Knowledge and Action. In J. R. Hobbs and R. C. Moore (eds.), *Formal Theories of the Commonsense World*, Ablex, Norwood, NJ, 1985, pp. 319–358.

[Moo85b] Moore, R. C., Semantical Considerations on Nonmonotonic Logic. *Artificial Intelligence* **25**, 1985, pp. 75–94.

[Rei91] Reiter, R., The Frame Problem in the Situation Calculus: A simple Solution (sometimes) and a Completeness Result for Goal Regression. In V. Lifshitz (ed.), *Artificial Intelligence and Mathematical Theory of Computation*, Academic Press, 1991, pp. 359–380.

[Rei93] Reiter, R., Proving Properties of States in the Situation Calculus. *Artificial Intelligence*, **64**, 1993, pp. 337-351.

[SL93] Scherl, R. and Levesque, H. J., The Frame Problem and Knowledge Producing Actions. in *Proc. of the National Conference on Artificial Intelligence (AAAI-93)*, AAAI Press, 1993, 689–695.