A Situation-Calculus Based Theory of Justified Knowledge and Action

Richard Scherl
Department of Computer Science & Software Engineering
Monmouth University
West Long Branch, NJ 07764
rscherl@monmouth.edu

Abstract

This paper proposes an integration of the situation calculus with justification logic. Justification logic can be seen as a refinement of a modal logic of knowledge and belief to one in which knowledge not only is something that holds in all possible worlds, but also is justified. The work is an extension of that of Scherl and Levesque's integration of the situation calculus with a modal logic of knowledge. We show that the solution developed here retains all of the desirable properties of the earlier solution while incorporating the enhanced expressibility of having justifications.

Introduction

The situation calculus is at the core of one major approach to *cognitive robotics* as it enables the representation and reasoning about the relationship between knowledge, perception, and action of an agent (Levesque and Lakemeyer 2007; Reiter 2001). Axioms are used to specify the prerequisites of actions as well as their effects, that is, the fluents that they change (McCarthy 1968). By using successor state axioms (Reiter 1991), one can avoid the need to provide frame axioms (McCarthy and Hayes 1969) to specify what particular actions do not change. This approach to dealing with the frame problem and the resulting style of axiomatization has proven useful as the foundation for the high-level robot programming language GoLog (Levesque et al. 1997; Giacomo, Lespérance, and Levesque 2000).

Knowledge and knowledge-producing actions have been incorporated into the situation calculus (Moore 1985; Scherl and Levesque 2003) by treating knowledge as a fluent that can be affected by actions. Situations from the situation calculus are identified with possible worlds from the semantics of modal logics of knowledge. It has been shown that knowledge-producing actions can be handled in a way that avoids the frame problem: knowledge-producing actions do not affect fluents other than the knowledge fluent, and that actions that are not knowledge-producing only affect the knowledge fluent as appropriate.

Within epistemology, the traditional analysis of knowledge (dating back to Plato) is tripartite (Ichikawa and Steup 2014). An agent, S knows that p iff (1) p is true; (2) S believes that p; (3) S is justified in believing that p. There has been much discussion of counterexamples to the sufficiency of this tripartite analysis (Lehrer 1990; Hendricks 2007; Nozick 1981; Goldman 1967).

The possible-world analysis of knowledge only handles the first two elements of the tripartite analysis; p is known if it is believed (i.e., true in all accessible worlds) and if it is true in the actual world. The component of justification has recently been added with the development of justification logic (Artemov 2008; 2001; Artemov and Fitting 2012). In justification logic, there is in addition to formulas, a category of terms called *justifications*. If t is a justification term and X is a formula, then $t{:}X$ is a formula which is read as "t is a justification for X." If the formula X is also true and believed to be true, one can then write $[t]:X$ for X is known with justification t.

One of the examples used in the philosophical literature mentioned above is the Red Barn Example (Luper 2012; Artemov and Fitting 2012; Kripke 2011; Nozick 1981; Dretske 2005). *Henry is driving through the countryside and perceptually identifies an object as a barn. Normally, one would then say that Henry knows that it is a barn. But Henry does not know there are expertly made papier-mâché barns. Then we would not want to say that Henry knows it is a barn unless he has some evidence against it being a papier-mâché barn. But what if in the area where Henry is traveling, there are no papier-mâché red barns. Then if Henry perceives a red barn, he can then be said to know there is a red barn and therefore a barn.*

The apparent problem here is only a problem within a modal logic of knowledge. There are two ways of the agent "knowing" that a barn is red. One way is accidental. Henry may have a barn perception, and then believe that the object is a barn, but this is only accidentally true and therefore we don't want to say that Henry knows the object is a barn. If Henry perceives that the object is a red barn, he is then justified in knowing that the object is a red barn and can then infer correctly that the object is a barn. Modal logic does not distinguish between these two ways of knowing/believing.

Within justification logic (Artemov 2008; Artemov and Fitting 2012) there is no contradiction because the justifications differ. The modality of knowledge is an existential assertion that there is a justification of a proposition. In one case, there is a justification for the object being a barn via a barn perception and in the other case a justification for it being a barn via the red barn perception and propositional reasoning. Later in this paper, the example will be worked out in the situation calculus with justified knowledge.

Goldman (Goldman 1967) has argued that the tripartite analysis of knowledge needs to be augmented with the requirement that there is a causal chain from the truth of the proposition known to the knowledge of the proposition. Only in the case of knowledge via the red barn perception is this condition met. The justification of knowing that the object is a barn via the red barn perception can be seen as meeting this further condition, while the justification via the barn perception does not meet this condition (Artemov 2008; Artemov and Fitting 2012).

The author is not aware of any previous work on integrating the situation calculus with a notion of justified knowledge. There has been some work on integrating justifications into dynamic epistemic logic (Baltag, Renne, and Smets 2012; Renne 2008)

The Situation Calculus and the Frame Problem

The situation calculus (following the presentation in (Reiter 1991)) is a first-order language for representing dynamically changing worlds in which all of the changes are the result of named *actions* performed by some agent. Terms are used to represent states of the world, i.e., *situations*. If α is an action and s a situation, the result of performing α in s is represented by $\text{DO}(\alpha, s)$. The constant S_0 is used to denote the initial situation. Relations whose truth values vary from situation to situation, called *fluents*, are denoted by a predicate symbol taking a situation term as the last argument. For example, BROKEN (x, s) means that object x is broken in situation s. Functions whose denotations vary from situation to situation are called *functional fluents*. They are denoted by a function symbol with an extra argument taking a situation term, as in PHONE-NUMBER(BILL,s).

It is assumed that the axiomatizer has provided for each action $\alpha(\vec{x})$, an *action precondition axiom* of the form[1] given in (1), where $\pi_\alpha(\vec{x}, s)$ is the formula for $\alpha(\vec{x})$'s action preconditions.

Action Precondition Axiom

$$\text{POSS}(\alpha(\vec{x}), s) \equiv \pi_\alpha(\vec{x}, s) \tag{1}$$

An action precondition axiom for the action *drop* is given below.

$$\text{POSS}(\text{DROP}(x), s) \equiv \text{HOLDING}(x, s) \tag{2}$$

Furthermore, the axiomatizer has provided for each fluent F, two *general effect axioms* of the form given in 3 and 4.

General Positive Effect Axiom for Fluent F

$$\gamma^+_{\text{F}}(a, s) \;\rightarrow\; \text{F}(\text{DO}(a, s)) \tag{3}$$

General Negative Effect Axiom for Fluent F

$$\gamma^-_{\text{F}}(a, s) \;\rightarrow\; \neg\text{F}(\text{DO}(a, s)) \tag{4}$$

[1]By convention, variables are indicated by lower-case letters in italic font. When quantifiers are not indicated, the variables are implicitly universally quantified.

Here $\gamma^+_{\text{F}}(a, s)$ is a formula describing under what conditions doing the action a in situation s leads the fluent F to become true in the successor situation $\text{DO}(a, s)$ and similarly $\gamma^-_{\text{F}}(a, s)$ is a formula describing the conditions under which performing action a in situation s results in the fluent F becoming false in situation $\text{DO}(a, s)$.

For example, (5) is a positive effect axiom for the fluent BROKEN.

$$\begin{array}{c}[(a = \text{DROP}(y) \wedge \text{FRAGILE}(y)) \\ \vee \\ (\exists b\; a = \text{EXPLODE}(b) \wedge \text{NEXTO}(b, y, s))] \\ \rightarrow \text{BROKEN}(y, \text{DO}(a, s))\end{array} \tag{5}$$

Sentence 6 is a negative effect axiom for BROKEN.

$$a = \text{REPAIR}(y) \rightarrow \neg\text{BROKEN}(y, \text{DO}(a, s)) \tag{6}$$

It is also necessary to add *frame axioms* that specify when fluents remain unchanged. The frame problem arises because the number of these frame axioms in the general case is $2 \times \mathcal{A} \times \mathcal{F}$, where $\mathcal{A}$ is the number of actions and $\mathcal{F}$ is the number of fluents.

The approach to handling the frame problem (Reiter 1991; Pednault 1989; Schubert 1990) rests on a *completeness assumption*. This assumption is that axioms (3) and (4) characterize all the conditions under which action a can lead to a fluent F's becoming true (respectively, false) in the successor situation. Therefore, if action a is possible and F's truth value changes from *false* to *true* as a result of doing a, then $\gamma^+_{\text{F}}(a, s)$ must be *true* and similarly for a change from *true* to *false* ($\gamma^-_{\text{F}(a,s)}$ must be true). Additionally, *unique name axioms* are added for actions and situations.

Reiter(1991) shows how to derive a set of *successor state axioms* of the form given in 7 from the axioms (positive effect, negative effect and unique name) and the completeness assumption.

Successor State Axiom

$$\text{F}(\text{DO}(a, s)) \;\equiv\; \gamma^+_{\text{F}}(a, s) \;\vee\; (\text{F}(s) \;\wedge\; \neg\gamma^-_{\text{F}}(a, s)) \tag{7}$$

Similar successor state axioms may be written for functional fluents. A successor state axiom is needed for each fluent F, and an action precondition axiom is needed for each action a. The unique name axioms need not be explicitly represented as their effects can be compiled. Therefore only $\mathcal{F} + \mathcal{A}$ axioms are needed.

From (5) and (6), the following successor state axiom for BROKEN is obtained.

$$\begin{array}{c}\text{BROKEN}(y, \text{DO}(a, s)) \equiv \\ (a = \text{DROP}(y) \wedge \text{FRAGILE}(y)) \vee \\ (\exists b\; a = \text{EXPLODE}(b) \wedge\; \text{NEXTO}(b, y, s)) \vee \\ (\text{BROKEN}(y, s) \wedge a \neq \text{REPAIR}(y))\end{array} \tag{8}$$

Now note for example that if $\neg\text{BROKEN}(\text{OBJ}_1, \text{S}_0)$ holds, then it also follows (given the unique name axioms) that $\neg\text{BROKEN}(\text{OBJ}_1, \text{DO}(\text{DROP}(\text{OBJ}_2), \text{S}_0))$ holds as well.

Justification Logic

Justification logic adds to the machinery of propositional logic (or quantifier free first-order logic) justification terms that are built with justification variables $x, y, z, \ldots$ and justification constants $a, b, c, \ldots$ (using indices $i = 1, 2, 3, \ldots$ whenever needed) using the operations '$\cdot$' and '$+$.'

The logic of justifications includes (in addition to the classical propositional axioms and the rule of Modus Ponens), the following axioms

Application Axiom $s\!:\!(F \to G) \to (t\!:\!F \to [s \cdot t]\!:\!G)$,

Sum Axioms $s\!:\!F \to [s+t]\!:\!F$, $s\!:\!F \to [t+s]\!:\!F$

As needed, the following axioms are added.

Factivity $t\!:\!F \to F$

Positive Introspection $t\!:\!F \to !t\!:\!(t\!:\!F)$

Negative Introspection $\neg t\!:\!F \to ?t\!:\!(\neg t\!:\!F)$

Factivity is used in all logics of knowledge. The *Positive Introspection* operator '!' is a proof checker, that given t produces a justification $!t$ of $t : F$. The *negative introspection* operator '?' verifies that a justification assertion is false (Artemov 2008; Artemov and Fitting 2012; Rubtsova 2006).

The standard semantics for justification logics (Fitting 2005) are called *Fitting models* or *possible world justification models*. This is a combination of the usual Kripke/Hintikka possible world models with the necessary features to handle justifications (Mkrtychev 1997). A model for justification logic is a structure $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$. Here, $\langle \mathcal{G}, \mathcal{R} \rangle$ is a standard frame for modal logic with $\mathcal{G}$ being a set of possible worlds and $\mathcal{R}$ being a relation on the elements of $\mathcal{G}$. The element $\mathcal{V}$ is a mapping from ground propositions to $\mathcal{G}$ specifying which propositions are true in which worlds. In the work here, we assume that a particular element of $\mathcal{G}$ is specified as the actual world.

There is the evidence function $\mathcal{E}$ that maps justification terms and formulas to sets of worlds. The idea is that if a possible world $\Gamma \in \mathcal{E}(t, X)$ then t is relevant evidence for X at world Γ.

Given a Fitting model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$, the truth of a formula X at a possible world Γ, i.e., $\mathcal{M}, \Gamma \models X$ is given as follows:

1. $\mathcal{M}, \Gamma \models P$ iff $\Gamma \in V(P)$ for P a propositional letter;
2. It is not the case that $\mathcal{M}, \Gamma \models \perp$;
3. $\mathcal{M}, \Gamma \models X \to Y$ iff it is not the case that $\mathcal{M}, \Gamma \models X$ or $\mathcal{M}, \Gamma \models Y$;
4. $\mathcal{M}, \Gamma \models (t\!:\!X)$ iff $\Gamma \in \mathcal{E}(t, X)$ and for every $\Delta \in \mathcal{G}$, with $\Gamma \mathcal{R} \Delta$, $\mathcal{M}, \Delta \models X$.

The last condition is the crucial one. It requires that for something to be known, it both needs to be believed in the sense that it is true in every accessible world and that t is relevant evidence for x at that world. So, t:X holds iff X is believable and t is relevant evidence for X.

The following conditions need to be placed on the Evidence function:

- $\mathcal{E}(s, X \to Y) \cap \mathcal{E}(t, X) \subseteq \mathcal{E}(s \cdot t, Y)$
- $\mathcal{E}(s, X) \cup \mathcal{E}(t, X) \subseteq \mathcal{E}(s + t, X)$

These ensure that the application and sum axioms hold.

Additionally, the issue of a *constant specification* needs to be mentioned. All axioms of propositional logic that are used need to have justifications. Degrees of logical awareness can be distinguished through the constant specification. The constant specification (CS) is a set of justified formulas (axioms of propositional logic). A model $\mathcal{M}$ meets the constant specification CS as long as the following condition is met:

$$\text{if } c\!:\!X \in CS \text{ then } \mathcal{E}(c, X) = \mathcal{G}$$

This ensures that the axiom is justified in all possible worlds.

Within justification logic, the derivation of a justified formula such as $s : F$ is the derivation of F being known. The justifications distinguish different ways of knowing. Additionally, they represent how difficult it is to know something and therefore a mechanism for addressing the logical omniscience problem (Artemov and Kuznets 2006).

Representing Justified Knowledge in the Situation Calculus

The approach we take to formalizing knowledge[2] is to adapt the semantics of justification logic described in the previous section to the situation calculus. Following (Moore 1980; Scherl and Levesque 2003), we think of there being a binary accessibility relation over situations, where a situation s' is understood as being accessible from a situation s if as far as the agent knows in situation s, he might be in situation s'.

To treat knowledge as a fluent, we introduce a binary relation $\mathrm{K}(s', s)$, (representing $\mathcal{R}$) read as "s' is accessible from s" and treat it the same way we would any other fluent. In other words, from the point of view of the situation calculus, the last argument to K is the official situation argument (expressing what is known in situation s), and the first argument is just an auxiliary like the y in $\textsc{Broken}(y, s)$.[3]

A fluent is introduced to represent the function $\mathcal{E}$. This is the relation $\mathrm{E}(t, X, s)$, where t is an evidence term, X is a formula and s is a situation. There is no need to represent the evidence function as a function from justifications and formulas to a set of situations. Since each fluent already contains a situation argument, a relational fluent naturally represents the justifications for formulas at that situation.

We can now introduce the notation $\textbf{Knows}(t, \mathrm{P}, s)$ (t is justification for knowing P in situation s) as an abbreviation for a formula that uses K and E. For example:

$$\begin{aligned}\textbf{Knows}(t, \textsc{Broken}(y), s) &\stackrel{\text{def}}{=} \mathrm{E}(t, \textsc{Broken}(y), s) \wedge \\ &\forall s'\ \mathrm{K}(s', s) \to \textsc{Broken}(y, s').\end{aligned}$$

Note that this notation supplies the appropriate situation argument to the fluent on expansion.

Turning now to knowledge-producing actions imagine a $\textsc{sense}_\mathrm{P}$ action for a fluent P, such that after doing a $\textsc{sense}_\mathrm{P}$, the truth value of P is known. We introduce the

[2]The situation calculus is a first-order formalism. But the knowledge that we are modeling is a knowledge of propositional or quantifier-free first-order formulas.

[3]Note that using this convention means that the arguments to K are reversed from their normal modal logic use.

notation **Kwhether**(P, s) as an abbreviation for a formula indicating that the truth value of a fluent P is known.

$$\mathbf{Kwhether}(t, \mathrm{P}, s) \stackrel{\text{def}}{=} \mathbf{Knows}(t, \mathrm{P}, s) \vee \mathbf{Knows}(t, \neg \mathrm{P}, s),$$

It will follow from our specification in the next section that $\exists t\ \mathbf{Kwhether}(t, \mathrm{P}, \mathrm{DO}(\mathrm{SENSE}_{\mathbf{P}}, s))$ holds.

For clarity, a number of sorts[4] are gradually introduced. The sort SIT is used to distinguish between situations and other objects. It is assumed that we have axioms asserting that the initial situation S_0 is of type SIT and that everything of the form $\mathrm{DO}(a, s)$ is of type SIT. The letter s, possibly with subscripts, is used to as indication that the variable is of type SIT, without explicit use of the sort predicate.

Integrating Justified Knowledge and Action

The approach being developed here rests on the specification of a successor state axiom for the K relation. This successor state axiom will ensure that for all situations $\mathrm{DO}(a, s)$, the K relation will be completely determined by the K relation at s and the action a.

The successor state axiom for K will be developed in several steps through an illustration of possible models for an axiomatization. First, we illustrate the initial picture, without any actions. Then, we add a successor state axiom for K that works with ordinary non-knowledge-producing actions. Finally, we add knowledge-producing actions.

The Initial Picture: Without Actions

For illustration, consider a model for an axiomatization of the initial situation (without any actions) We can imagine that the term S_0 denotes the situation `S1` (an object in a model). Three situations (`S1`, `S2` and `S3`) are accessible via the K relation from `S1`. Proposition P is true in all of these situations[5], while proposition Q is true in `S1` and `S3`, but is false in `S2`. We also, have in `S1` that t_1 is evidence for P. Hence,$\mathrm{E}(t_1, \mathrm{P}, \mathrm{S}_0)$ holds. Therefore[6] the agent in `S1` knows P, but does not know Q. In other words, we have a model of **Knows**$(t_1, \mathrm{P}, \mathrm{S}_0)$ and $\forall t \neg$**Knows**$(t, \mathrm{Q}, \mathrm{S}_0)$.

Setting up the Initial Picture Restrictions need to be placed on the K relation so that it correctly models the accessibility relation of a particular justification logic. The problem is to do this in a way that does not interfere with the successor state axioms for K, which must completely specify the K relation for non-initial situations. The solution is to axiomatize the restrictions for the initial situation and then verify that the restrictions are then obeyed at all situations.

The sort INIT is used to restrict variables to range only over S_0 and those situations accessible from S_0. It is necessary to stipulate that:

$$\mathrm{INIT}(\mathrm{S}_0)$$
$$\forall s, s_1 \mathrm{INIT}(s_1) \rightarrow (\mathrm{K}(s, s_1) \rightarrow \mathrm{INIT}(s))$$
$$\forall s, s_1 \neg \mathrm{INIT}(s_1) \rightarrow (\mathrm{K}(s, s_1) \rightarrow \neg \mathrm{INIT}(s))$$
$$\mathrm{INIT}(s) \rightarrow \neg \exists s' (s = \mathrm{DO}(a, s'))$$

We want to require that the situation S_0 is a member of the sort INIT, everything K-accessible from an INIT situation is also INIT, and that everything K-accessible from a situation that is not INIT is also not INIT. Also it is necessary to require that none of the situations that result from the occurrence of an action are INIT. We also need to specify that everything of type INIT is also of type SIT.

Given the decision that we are to use a particular modal logic of knowledge, it is necessary to axiomatize the corresponding restrictions that need to be placed on the K relation. These are listed below and are merely first-order representations of the conditions on the accessibility relations for the standard modal logics of knowledge discussed in the literature (Hughes and Cresswell 1968; Kripke 1963; Chellas 1980; Bull and Segerberg 1984). All of these modal logics have corresponding justification logics (Artemov 2008; Artemov and Fitting 2012). The reflexive restriction is always added as we want a modal logic of knowledge. Some subset of the other restrictions are then added to semantically define a particular modal logic[7].

Reflexive $\forall s_1{:}\mathrm{INIT}\ \mathrm{K}(s_1, s_1)$

Euclidean $\forall s_1{:}\mathrm{INIT}, s_2{:}\mathrm{INIT}, s_3{:}\mathrm{INIT}$
$\mathrm{K}(s_2, s_1) \wedge \mathrm{K}(s_3, s_1) \rightarrow \mathrm{K}(s_3, s_2)$

Symmetric $\forall s_1{:}\mathrm{INIT}, s_2{:}\mathrm{INIT}\ \mathrm{K}(s_2, s_1) \rightarrow \mathrm{K}(s_1, s_2)$

Transitive $\forall s_1{:}\mathrm{INIT}, s_2{:}\mathrm{INIT}, s_3{:}\mathrm{INIT}$
$\mathrm{K}(s_2, s_1) \wedge \mathrm{K}(s_3, s_2) \rightarrow \mathrm{K}(s_3, s_1)$

For clarity a sort JUST is used to specify which objects are justifications. The letter *t*, possibly with subscripts, is used to indicate that the variable ranges over justifications, at times without explicit indication of the sort. It is also necessary to distinguish between those justifications that are handled by successor state axioms and those that are asserted to hold in every situation. The justifications handled by successor state axioms are of type BASIC. Both of these will be explained shortly. We also need a sort FORM to range over formulas of propositional logic. Variables X and Y are used to range over formulas without explicit use of the sort predicate.

We need the following to handle the application axiom

$$\begin{aligned} &\forall t{:}\mathrm{JUST}\, \forall a{:}\mathrm{JUST}\, \forall s_1{:}\mathrm{SIT}\ \mathrm{E}(a, X \rightarrow Y, s_1) \\ &\quad \wedge\ \mathrm{E}(t, X, s_1) \\ &\quad\quad \rightarrow \mathrm{E}(a \cdot t, Y, s_1) \end{aligned} \tag{9}$$

and the following for the sum axiom.

$$\begin{aligned} &\forall t{:}\mathrm{JUST}\ a{:}\mathrm{JUST}\ \forall s_1{:}\mathrm{SIT}\ \mathrm{E}(a, X, s_1) \vee \mathrm{E}(t, X, s_1) \\ &\quad \rightarrow \mathrm{E}(a + t, X, s_1) \end{aligned} \tag{10}$$

[4]Here sorts or types are simply one place predicates. But commonly $\forall s{:}\mathrm{SIT}\ \varphi$ is used an abbreviation for $\forall s\ \mathrm{SIT}(s) \rightarrow \varphi$

[5]For expository purposes we speak informally of a proposition being true in a situation rather than saying that the situation is in the relation denoted by the predicate symbol P.

[6]Note that the the justification is needed for the agent to know a proposition. In (Scherl and Levesque 2003), anything true in all accessible worlds is known.

[7]As in (Scherl and Levesque 2003) it can be shown that these properties persist through all successor situations.

Additionally, for every $t : X \in CS$, we need to have:

$$\forall s{:}\textsc{Sit}\ \mathrm{E}(t, X, s) \tag{11}$$

The axiomatization needs to specify that the justifications introduced as part of the constant specification are not BASIC as are those that are formed by the operators $+$ and $\cdot$ (even if the arguments are of type BASIC).

Adding Ordinary Actions

Now the language includes more terms describing situations. In addition to S_0, there is the DO function along with the presence of actions in the language. More situations are added to the model described earlier. The function denoted by DO maps the initial set of situations to these other situations. (These in turn are mapped to yet other situations, and so on). These situations intuitively represent the occurrence of actions. The situations `S1`, `S2`, and `S3` are mapped by DO and the action terms MOVE, PICKUP, or DROP to various other situations. The question is what is the K relation between these situations. Our axiomatization of the K relation places constraints on the K relation in the models. We first cover the simpler case of non-knowledge-producing actions and then discuss knowledge-producing actions.

For non-knowledge-producing actions (e.g. $\textsc{drop}(x)$), the specification is as follows:

$$\begin{array}{l} \mathrm{K}(s'', \textsc{do}(\textsc{drop}(x), s)) \equiv \\ \quad \exists s' \, (\textsc{Poss}(\textsc{drop}(x), s') \wedge \mathrm{K}(s', s) \wedge \\ \quad\quad s'' = \textsc{do}(\textsc{drop}(x), s')) \end{array} \tag{12}$$

The idea here is that as far as the agent at world s knows, he could be in any of the worlds s' such that $\mathrm{K}(s', s)$. At $\textsc{do}(\textsc{drop}(x), s)$ as far as the agent knows, he can be in any of the worlds $\textsc{do}(\textsc{drop}(x), s')$ for any s' such that both $\mathrm{K}(s', s)$ and $\textsc{Poss}(\textsc{drop}(x), s')$ hold. So the only change in knowledge (given only 12) that occurs in moving from s to $\textsc{do}(\textsc{drop}(x), s)$ is the knowledge that the action DROP has been performed.

To continue our example of the initial arrangement of situations and the fluents P and Q, we imagine that an action named DROP makes P false, but does not change the truth value of Q. We have the following successor state axioms:

$$\mathrm{P}(\textsc{do}(a, s)) \equiv a \neq \textsc{drop} \wedge \mathrm{P}(s) \tag{13}$$

$$\mathrm{Q}(\textsc{do}(a, s)) \equiv \mathrm{Q}(s) \tag{14}$$

We now have additional situations resulting from the DO function applied to DROP and the successor state axiom for K fully specifies the K relation between these situations. Here we see the situation `do(drop,S1)`, denoted by $\textsc{do}(\textsc{drop}, \mathrm{S}_0)$, which represents the result of performing a drop action in the situation denoted by S_0. Our axiomatization requires that this situation be K related only to the situations `do(drop,S1)`, `do(drop,S2)` and `do(drop,S3)`.

The DROP action does not affect the truth of Q, but makes P false. So, we see that proposition P is false in each of `do(drop,S1)`, `do(drop,S2)` and `do(drop,S3)`, while proposition Q is true in `do(drop,S1)` and `do(drop,S3)`, but is false in `do(drop,S2)`. Therefore in `do(drop,S1)` the fluent ¬P holds in all K accessible situations, but this is not the case for the fluent Q.

We need a successor state axiom[8] for E. Corresponding to each successor state axiom of the form given in (7), there must be

$$\begin{array}{l} \forall t{:}\textsc{basic}\ \mathrm{E}(t, X, \textsc{do}(a, s)) \equiv \\ (\mathrm{E}(t, X, s) \wedge X \neq P \wedge X \neq \neg P) \vee \\ \quad ((t = \textsc{mkjust}(\textsc{do}(a, s)) \wedge \\ \quad\quad ((\gamma^+_{\mathrm{F}}(a, s) \vee (\mathrm{F}(s) \wedge \neg\gamma^-_{\mathrm{F}}(a, s))) \wedge \\ \quad\quad\quad X = P) \\ \quad\quad\quad\quad \vee \\ \quad\quad ((\gamma^-_{\mathrm{F}}(a, s) \vee (\neg\mathrm{F}(s) \wedge \neg\gamma^+_{\mathrm{F}}(a, s))) \wedge \\ \quad\quad\quad (X = \neg P))) \end{array} \tag{15}$$

The axiomatization needs to specify that all justifications formed from MKJUST are of type BASIC.

To return to our running axiomatization, we have

$$\begin{array}{l} \forall t{:}\textsc{basic}\ \mathrm{E}(t, X, \textsc{do}(a, s)) \equiv \\ \quad (\mathrm{E}(t, X, s) \wedge X \neq P) \vee \\ \quad\quad (X = \neg P \wedge a = \textsc{drop} \wedge \\ \quad\quad\quad t = \textsc{mkjust}(\textsc{do}(\textsc{drop}(x), s))) \end{array} \tag{16}$$

The following two sentences hold in this model: **Knows**$(\textsc{mkjust}(\textsc{do}(\textsc{drop}(x), s)), \neg\mathrm{P}, \textsc{do}(\textsc{drop}, \mathrm{S}_0))$ and $\forall t$ ¬**Knows**$(t, \mathrm{Q}, \textsc{do}(\textsc{drop}, \mathrm{S}_0))$. The agent's knowledge of Q has remained the same, and the knowledge of P is a result of the knowledge of P in the previous situation along with the knowledge of the effect of the action DROP.

Adding Knowledge-Producing Actions

Now consider the simple case of a knowledge-producing action $\textsc{sense}_Q$ that determines whether or not the fluent Q is true (following Moore (1980; 1985)). There may also be ordinary actions, which are not knowledge-producing.

We imagine that the action has an associated sensing result function. This result is "YES" if "Q" is true and "NO" otherwise. The symbols are given in quotes to indicate that they are not fluents. We axiomatize the sensing result as follows:

$$\begin{array}{l} \textsc{sr}(\textsc{sense}_Q, s) = r \equiv (r = \text{“YES”} \wedge \mathrm{Q}(s)) \\ \quad \vee (r = \text{“NO”} \wedge \neg\mathrm{Q}(s)) \end{array} \tag{17}$$

The question that we need to consider is what situations are K accessible from $\textsc{do}(\textsc{sense}_Q, s_0)$.

$$\begin{array}{l} \mathrm{K}(s'', \textsc{do}(\textsc{sense}_{\mathrm{Q}}, s)) \equiv \\ \exists s' \, (\textsc{Poss}(\textsc{sense}_{\mathrm{Q}}, s') \wedge \mathrm{K}(s', s) \wedge \\ \quad s'' = \textsc{do}(\textsc{sense}_Q, s') \wedge \\ \quad\quad \textsc{sr}(\textsc{sense}_{\mathrm{Q}}, s) = \textsc{sr}(\textsc{sense}_Q, s')) \end{array} \tag{18}$$

Again, as far as the agent at world s knows, he could be in any of the worlds s' such that $\mathrm{K}(s', s)$ holds. At $\textsc{do}(\textsc{sense}_Q, s)$ as far as the agent knows, he can be in any of the worlds $\textsc{do}(\textsc{sense}_Q, s')$ such that $\mathrm{K}(s', s)$ and

[8]Here MKJUST is a gensym like function that creates a justification out of the action that has occurred. The intuition is that the occurrence of the action is the justification for the knowledge of the changes that are caused by the action.

$\text{POSS}(\text{SENSE}_Q, s')$ hold by (18), and also $Q(s) \equiv Q(s')$ by the combination of (17) and (18) holds. The idea here is that in moving from s to $\text{DO}(\text{SENSE}_Q, s)$, the agent not only knows that the action SENSE_Q has been performed (since every accessible situation results from the DO function and the SENSE_Q action), but also the truth value of the predicate Q. Observe that the successor state axiom for Q (sentence 14) guarantees that Q is true at $\text{DO}(\text{SENSE}_Q, s)$ if and only if Q is true at s, and similarly for s' and $\text{DO}(\text{SENSE}_Q, s')$. Therefore, Q has the same truth value in all worlds s'' such that $K(s'', \text{DO}(\text{SENSE}_Q, s))$, and so $\mathbf{Kwhether}(Q, \text{DO}(\text{SENSE}_Q, s))$ is true.

To return to our running example, which is the illustration of the result of a SENSE_Q action, note that the only situations accessible via the K relation from `do(sense,S1)` (denoted by $\text{DO}(\text{SENSE}_Q, s_0)$) are `do(sense, S1)` and `do(sense,S3)`. The situation `do(sense,S2)` is not K accessible. Therefore $\mathbf{Knows}(t, P, \text{DO}(\text{SENSE}_Q, S_0))$ is true as it was before the action was executed, but also now $\mathbf{Knows}(t', Q, \text{DO}(\text{SENSE}_Q, S_0))$ is true where t' is a new justification as introduced in the successor state axiom for E given below. The knowledge of the agent being modeled has increased.

In general, there may be many knowledge-producing actions, as well as many ordinary actions. To characterize all of these, we have a function SR (for sensing result), and for each action α, a sensing-result axiom of the form:

$$\text{SR}(\alpha(\vec{x}), s) = r \equiv \phi_\alpha(\vec{x}, r, s) \tag{19}$$

For ordinary actions, the result is always the same, with the specific result not being significant. For example, we could have:

$$\text{SR}(\text{PICKUP}(x), s) = r \equiv r = \text{“OK”} \tag{20}$$

The successor state axiom for K is as follows:

Successor State Axiom for K

$$\begin{aligned} &K(s'', \text{DO}(a, s)) \equiv \\ &\quad (\exists s'\, s'' = \text{DO}(a, s') \\ &\quad\quad \wedge K(s', s) \wedge \text{POSS}(a, s') \\ &\quad\quad\quad \wedge \text{SR}(a, s) = \text{SR}(a, s')) \end{aligned} \tag{21}$$

The relation K at a particular situation $\text{DO}(a, s)$ is completely determined by the relation at s and the action a.

We need a successor state axiom for E and the sensing action.

$$\begin{aligned} &\forall t{:}\text{BASIC}\ E(t, X, \text{DO}(\text{SENSE}_Q)) \equiv \\ &\quad (E(t, X, s) \wedge X \neq Q \wedge X \neq \neg Q) \vee \\ &\quad (((X = Q \wedge \text{SR}(\text{SENSE}_Q, s) = \text{“YES”}) \vee \\ &\quad (X = \neg Q \wedge \text{SR}(\text{SENSE}_Q, s) = \text{“YES”})) \\ &\quad \wedge t = \text{MKJUST}(\text{DO}(\text{DROP}(x), s))) \end{aligned} \tag{22}$$

For every sensing-result axiom of the form (19) we need an axiom of the form (22). The axiomatization also needs to specify that all justifications formed from MKJUST are of type BASIC.

Example

Consider the red barn example mentioned earlier[9]. We have two sensing actions; $\text{SENSE}_{B\wedge R}$ and SENSE_B. The first represents the action of sensing whether there is a red barn and the second is the sensing of whether there is a barn. Note that by the problem description only the first is a causal justification for knowledge. This is meta-information, not available to the agent.

[9] Here the example follows (Artemov 2008; Artemov and Fitting 2012).

The sensing result axioms are as follows:

$$\begin{aligned} &\text{SR}(\text{SENSE}_{B\wedge R}, s) = r \equiv \\ &\quad (r = \text{“YES”} \wedge (\text{RED}(s) \wedge \text{BARN}(s)) \\ &\quad \vee (r = \text{“NO”} \wedge \neg(\text{RED}(s) \wedge \text{BARN}(s)) \end{aligned} \tag{23}$$

$$\begin{aligned} &\text{SR}(\text{SENSE}_B, s) = r \equiv \\ &\quad (r = \text{“YES”} \wedge \text{BARN}(s)) \\ &\quad \vee (r = \text{“NO”} \wedge \neg\text{BARN}(s)) \end{aligned} \tag{24}$$

We assume that E has been axiomatized following the approach in the previous section. It is also necessary to add the following: $\text{BARN}(S_0)$ and $\text{RED}(S_0)$. Additionally, we need to add a propositional axiom $(B \wedge R) \rightarrow B$ to the constant specification. So, it is justified by justification A.

$$\forall s\, E(\text{A}, (B \wedge R) \rightarrow B, s) \tag{25}$$

The successor state axioms for BARN and RED need to be added, but they are simple since there are no actions that change these fluents. The successor state axioms for the sensing action are of the form given in the previous section.

Now the axiomatization entails

$$\begin{aligned} &\mathbf{Knows}(\text{MKJUST}(\text{DO}(\text{SENSE}_B, S_0)), \\ &\quad \text{BARN}, \text{DO}(\text{SENSE}_B, S_0)) \end{aligned} \tag{26}$$

and

$$\begin{aligned} &\mathbf{Knows}((\text{A} \cdot \text{MKJUST}(\text{DO}(\text{SENSE}_{B\wedge R}, S_0))), \\ &\quad \text{BARN}, \text{DO}(\text{SENSE}_{B\wedge R}, S_0)) \end{aligned} \tag{27}$$

By the meta-information given in the problem description only the second is true knowledge. The formalism allows the two justifications for the knowledge of barn to be distinguished, while the modal logic based approach of (Scherl and Levesque 2003) does not allow them to be distinguished.

Summary

This paper has presented preliminary results on integrating the justification logic model of knowledge into the situation calculus with knowledge and knowledge producing actions. The positive results of this work is that (as compared to the situation calculus with a modal view of knowledge) one is able to make a more fine-grained representation of the different ways an agent may have knowledge. Additionally, the agent is not logically omniscient.

Some of the properties (Scherl and Levesque 2003) for the situation calculus with knowledge carry over to the case of justified knowledge. Space does not permit a full exposition. But all of these properties show that actions only affect knowledge in the appropriate way. Note that the property (from (Scherl and Levesque 2003)) that agents know the consequences of acquired knowledge does not hold as knowledge of the consequences depends on having the justification that incorporates the reasoning involved.

References

Artemov, S. N., and Fitting, M. 2012. Justification logic. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Fall 2012 edition.

Artemov, S. N., and Kuznets, R. 2006. Logical omniscience via proof complexity. In Ésik, Z., ed., *Computer Science Logic, 20th International Workshop, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 25–29, 2006, Proceedings*, volume 4207 of *Lecture Notes in Computer Science*. Springer. 135–149.

Artemov, S. 2001. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic* 7(1):1–36.

Artemov, S. N. 2008. The logic of justification. *The Review of Symbolic Logic* 1(4):477–513.

Baltag, A.; Renne, B.; and Smets, S. 2012. The logic of justified belief change, soft evidence and defeasible knowledge. In Ong, L., and de Queiroz, R., eds., *Proceedings of the 19th Workshop on Logic, Language, Information, and Computation (WoLLIC 2012)*, volume 7456 of *Lecture Notes in Computer Science*, 168–190. Buenos Aires, Argentina: Springer-Verlag Berlin Heidelberg.

Bull, R., and Segerberg, K. 1984. Basic modal logic. In Gabbay, D., and Guenther, F., eds., *Handbook of Philosophical Logic, Vol. II*. Dordrecht: D. Reidel Publishing Company. chapter 1, 1–88.

Chellas, B. F. 1980. *Modal Logic: An Introduction*. Cambridge: Cambridge University Press.

Dretske, F. 2005. Is knowledge closed under known entailment? the case against closure. In *Contemporary Debates in Epistemology*. Blackwell. 13–26.

Fitting, M. 2005. The logic of proofs, semantically. *Annals of Pure and Applied Logic* 132(1):1–25.

Giacomo, G. D.; Lespérance, Y.; and Levesque, H. J. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence* 109–169.

Goldman, A. I. 1967. A causal theory of knowing. *Journal of Philosophy* 64(12):357–372.

Hendricks, V. 2007. *Mainstream and Formal Epistemology*. New York, NY: Cambridge University Press.

Hughes, G., and Cresswell, M. 1968. *An Introduction to Modal Logic*. London: Methuen and Co.

Ichikawa, J. J., and Steup, M. 2014. The analysis of knowledge. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition.

Kripke, S. 1963. Semantical considerations on modal logic. *Acta Philosophica Fennica* 16:83–94.

Kripke, S. 2011. *Philosophical Troubles: Collected Papers, Volume 1*. New York, NY: Oxford University Press.

Lehrer, K. 1990. *Theory of Knowledge*. London, UK: Routledge.

Levesque, H., and Lakemeyer, G. 2007. Cognitive robotics. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*. Elsevier. to appear.

Levesque, H.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*.

Luper, S. 2012. The epistemic closure principle. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Fall 2012 edition.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*. Edinburgh, UK: Edinburgh University Press. 463–502.

McCarthy, J. 1968. Programs with common sense. In Minsky, M., ed., *Semantic Information Processing*. The MIT Press. chapter 7, 403–418.

Mkrtychev, A. 1997. Models for the logic of proofs. In Adian, S., and Nerode, A., eds., *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*. Springer. 266–275.

Moore, R. C. 1980. Reasoning about knowledge and action. *Technical Note 191, SRI International*.

Moore, R. 1985. A formal theory of knowledge and action. In Hobbs, J., and Moore, R., eds., *Formal Theories of the Commonsense World*. Norwood, NJ: Ablex. 319–358.

Nozick, R. 1981. *Philosophical Explanations*. Cambridge, Massachusetts: The Belknap Press of Harvard University Press.

Pednault, E. 1989. ADL: exploring the middle ground between STRIPS and the sit uation calculus. In Brachman, R.; Levesque, H.; and Reiter, R., eds., *Proceedings of the First International Conference on Princi ples of Knowledge Representation and Reasoning*. San Mateo, California: Morgan Kaufmann Publishers, Inc. 324–332.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. San Diego, CA: Academic Press. 359–380.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. Cambridge, Massachusetts: The MIT Press.

Renne, B. 2008. *Dynamic Epistemic Logic with Justification*. Ph.D. Dissertation, City University of New York.

Rubtsova, N. M. 2006. On realization of S5-modality by evidence terms. *Journal of Logic and Computation* 16(5):671–684.

Scherl, R., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(1–2):1–39.

Schubert, L. 1990. Monotonic solution of the frame problem in the situation calcul us: an efficient method for worlds with fully specified actions. In Kyberg, H. E.; Loui, R.; and Carlson, G., eds., *Knowledge Representation and Defeasible Reasoning*. Boston, Mass.: Kluwer Academic Press. 23–67.