Situations, si! Situation terms, no!

Gerhard Lakemeyer
Dept. of Computer Science
RWTH Aachen
52056 Aachen
Germany
gerhard@cs.rwth-aachen.de

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

Abstract

The situation calculus, as proposed by McCarthy and Hayes, and developed over the last decade by Reiter and co-workers, is reconsidered. A new logical variant is proposed that captures much of the expressive power of the original, but where certain technical results are much more easily proven. This is illustrated using two existing non-trivial results: the regression theorem and the determinacy of knowledge theorem of Reiter. We also obtain a regression theorem for knowledge, and show how to reduce reasoning about knowledge and action to non-epistemic non-dynamic reasoning about the initial situation.

Introduction

The situation calculus, as proposed by McCarthy and Hayes (McCarthy 1963; McCarthy and Hayes 1969) is a dialect of first-order logic for representing and reasoning about the preconditions and effects of actions. A second-order refinement of the language, developed by Reiter and his colleagues (Reiter 2001a), forms the theoretical and implementation foundation for *Golog* (Levesque et al. 1997), a language for the high-level control of robots and other agents (see, for example, (Burgard et al. 2000; McIlraith and Son 2002)). Over the past decade, a number of extensions have been proposed to deal with issues such as time, natural actions, knowledge of agents, numerical uncertainty, or utilities (see (Reiter 2001a) and the references therein).

As a formalism, the situation calculus is based on *axioms*. In Reiter's formulation, which is also our starting point, these take the form of so-called *basic action theories*. These consist of a number of foundational axioms, which define the space of situations, unique-name axioms for actions, axioms describing action preconditions and effects, and axioms about the initial situation.

What makes basic action theories particularly useful is the formulation of action effects in terms of *successor state axioms*, which not only provide a simple solution to the frame problem (Reiter 1991) but also allow the use of regression-based reasoning, which has been used in planning (Finzi, Pirri, and Reiter 2000) and forms the core of every Golog interpreter, for example. Derivations using regression are simple, clear, and computationally feasible.

Since the situation calculus is defined axiomatically, no special semantics is needed. Tarskian models suffice, provided they satisfy the foundational axioms. When the focus is on logical entailments, which is the case in the execution of Golog programs, for example, this approach seems perfectly adequate.

However, when we wish to consider theoretical questions about basic action theories that are not direct entailment questions, problems arise. For example, suppose we are doing an analysis of our system, and want to know, if whenever *Theory1* entails *Formula1*, is it also true that *Theory2* entails *Formula2*? Here we can run into serious complications in an axiomatic setting unless there are ways to take a derivation of *Formula1* from *Theory1* and convert it into a derivation of *Formula2* from *Theory2*. (Similar issues arise with consistency questions.)

For instance, consider the epistemic extension of the situation calculus, as introduced by Moore and later extended by Scherl and Levesque (Moore 1985; Scherl and Levesque 2003). If $\mathit{Know}(A)$ entails $(\mathit{Know}(B) \vee \mathit{Know}(C))$ in this theory, is it also true that $\mathit{Know}(A)$ entails $\mathit{Know}(B)$ or $\mathit{Know}(A)$ entails $\mathit{Know}(C)$? For restricted A, B, C, the answer is *yes*, but the proof requires a multi-page argument using considerable proof-theoretic machinery such as Craig's Interpolation Lemma (Reiter 2001b).

One might wonder whether a semantic proof using Tarski structures would be any easier. The answer, in short, is *no*. The problem is that different structures can have different domains and effort is required to standardize the domains, identify the situations, and amalgamate multiple structures into a single structure that satisfies the foundational axioms. While certainly possible, the argument is again long and complicated.

In contrast, in the logic of only-knowing $\mathcal{OL}$ (Levesque and Lakemeyer 2001), the semantic proof of the above determinacy of knowledge theorem is simple, clear and direct. One reason for this is the use of a semantic formulation involving possible worlds for knowledge (Hintikka 1962; Fagin et al. 1995). Typical of these formalisms, situations and possible worlds are not reified in the language itself. Beyond this, however, a major factor in the simplicity of proofs in $\mathcal{OL}$ is the use of *standard names*, which allows a substitutional interpretation of the first-order quantifiers. While there have been philosophical arguments against substitutional quantification (Kripke 1976), our experience has been that its technical simplicity has been of tremendous help in tackling issues such as quantifying-in (Kaplan 1971), which are rarely addressed in other formalisms.

Since $\mathcal{OL}$ only deals with static knowledge bases, an amalgamation of $\mathcal{OL}$ and the situation calculus was previously proposed (Lakemeyer and Levesque 1998). However, this formalization kept situations reified, did not allow substitutional quantification, and the definition of knowledge required second-order logic, all of which again complicated the proofs considerably, even semantic ones.

In this paper, we propose a rather different amalgamation of $\mathcal{OL}$ and the situation calculus called $\mathcal{ES}$. The idea is to keep the simplicity of $\mathcal{OL}$, and while dropping some of the expressiveness of the ordinary situation calculus, retain its main benefits, like successor state axioms to solve the frame problem and regression-based reasoning. In particular, we will use a possible-world semantics where situations are part of the semantics but do not appear as terms in the language. In order to represent what is true in a situation after a number of actions have occurred, we use special modal operators. For example, we will have formulas like those of traditional dynamic logic (Pratt 1976; Harel 1984), such as

$$[\mathit{pickup}(\mathit{obj5})]\,[\mathit{drop}(\mathit{obj5})]\,\mathit{Broken}(\mathit{obj5})$$

to say that *obj5* is broken after doing the two actions. In contrast to other modal approaches to reasoning about action such as (Castilho, Gasquet, and Herzig 1999; Herzig et al. 2000; Demolombe 2003), we also allow formulas of the form $\forall a.([a]\mathit{Broken}(\mathit{obj5}) \equiv \phi)$, where modalities contain (action) variables. This feature will be key in reconstructing Reiter's basic action theories in our language. Moreover, unlike standard modal logics (including dynamic logics), we will be able to use a substitutional interpretation for first-order quantifiers.[1] This is perhaps the main reason why we cannot afford situation terms as part of our language. The epistemic situation calculus requires us to consider an uncountable number of initial situations (see (Levesque, Pirri, and Reiter 1998) for a second-order foundational axiom that makes this explicit). In a language with only countably many situation terms, this would preclude a substitutional interpretation of quantifiers.

Yielding much simpler proofs (like the determinacy of knowledge and the correctness of regression) is not the only benefit of this approach. As we will see, the use of regression allows us to reduce reasoning about basic action theories (possibly involving knowledge) to reasoning about the initial situation (possibly with knowledge), as in the original situation calculus. What is new here, however, is that we can then leverage the representation theorem of $\mathcal{OL}$ (Levesque and Lakemeyer 2001) and show that certain forms of reasoning about knowledge and action reduce overall to strictly first-order reasoning about the initial situation, without action and without knowledge.

[1]For simplicity, we will only consider the first-order version of our proposal here. The second-order version, which is necessary to formalize Golog, will appear in a companion paper (Lakemeyer and Levesque 200x).

The rest of the paper is organized as follows. In the next section we introduce the syntax and semantics of $\mathcal{ES}$, followed by a discussion of basic action theories and regression for the non-epistemic case. In the following section, we consider properties of knowledge, extend regression and show the connection to the representation theorem of $\mathcal{OL}$. We end the paper with a discussion of related work and concluding remarks.

The Logic $\mathcal{ES}$

The Language

The language consists of formulas over symbols from the following vocabulary:

- variables $V = \{x_1, x_2, x_3, \ldots, y_1, \ldots, z_1, \ldots, a_1, \ldots\}$;
- fluent predicates of arity k: $F^k = \{f_1^k, f_2^k, \ldots\}$; for example, *Broken*; we assume this list includes the distinguished predicates *Poss* and *SF* (for sensing);
- rigid functions of arity k: $G^k = \{g_1^k, g_2^k, \ldots\}$; for example, *obj5*, *pickup*; note that G^0 is a set of non-fluent constants (or standard names);
- connectives and other symbols: $=$, $\wedge$, $\neg$, $\forall$, *Know*, *OKnow*, $\Box$, round and square parentheses, period, comma.

For simplicity, we do not include rigid (non-fluent) predicates or fluent (non-rigid) functions. The *terms* of the language are the least set of expressions such that

1. Every first-order variable is a term;
2. If $t_1, \ldots, t_k$ are terms, then so is $g^k(t_1, \ldots, t_k)$.

We let R denote the set of all rigid terms (here, all ground terms). For simplicity, instead of having variables of the *action* sort distinct from those of the *object* sort as in the situation calculus, we lump both of these together and allow ourselves to use any term as an action or as an object.[2] Finally, the *well-formed formulas* of the language form the least set such that

1. If $t_1, \ldots, t_k$ are terms, then $f^k(t_1, \ldots, t_k)$ is an (atomic) formula;
2. If t_1 and t_2 are terms, then $(t_1 = t_2)$ is a formula;
3. If t is a term and α is a formula, then $[t]\alpha$ is a formula;
4. If α and β are formulas, then so are $(\alpha \wedge \beta)$, $\neg\alpha$, $\forall x.\alpha$, $\Box\alpha$, *Know*(α), *OKnow*(α).

We read $[t]\alpha$ as "α holds after action t", $\Box\alpha$ as "α holds after any sequence of actions," *Know*(α) as "α is known", and *OKnow*(α) as "α is all that is known." As usual, we treat $\exists x.\alpha$, $\exists x\phi.\alpha$, $(\alpha \vee \beta)$, $(\alpha \supset \beta)$, and $(\alpha \equiv \beta)$ as abbreviations. We call a formula without free variables a *sentence*. We sometimes use a finite set of sentences Σ as part of a formula, where it should be understood conjunctively.

In the following, we will sometimes refer to special sorts of formulas and use the following terminology:

- a formula with no $\Box$ operators is called *bounded;*

[2]Equivalently, the version in this paper can be thought of as having action terms but no object terms.

- a formula with no $\Box$ or $[t]$ operators is called *static;*
- a formula with no *Know* or *OKnow* operators is called *objective;*
- a formula with no fluent, $\Box$, or $[t]$ operators outside the scope of a *Know* or *OKnow* is called *subjective;*
- a formula with no *Know*, *OKnow*, $\Box$, $[t]$, *Poss*, or *SF* is called a *fluent* formula.

The semantics

Intuitively, a world w will determine which fluents are true, but not just initially, also after any sequence of actions. We let P denote the set of all pairs $\sigma{:}\rho$ where $\sigma \in R^*$ is considered a sequence of actions, and $\rho = f^k(r_1, \ldots, r_k)$ is a ground fluent atom. In general, formulas are interpreted relative to a model $M = \langle e, w \rangle$ where $e \subseteq W$ and $w \in W$, and where $W = [P \rightarrow \{0, 1\}]$. The e determines all the agent knows initially, and is referred to as an *epistemic state.*

We interpret first-order variables substitutionally over the rigid terms R, that is, we treat R as being isomorphic to a fixed universe of discourse. This is similar to $\mathcal{OL}$, where we used standard names as the domain. We also define $w' \simeq_\sigma w$ (read: w' and w agree on the sensing for σ) inductively by the following:

1. when $\sigma = \langle\,\rangle$, $w' \simeq_\sigma w$, for every w' and w;
2. $w' \simeq_{\sigma \cdot r} w$ iff $w' \simeq_\sigma w$ and $w'[\sigma{:}\mathit{SF}(r)] = w[\sigma{:}\mathit{SF}(r)]$.

Here is the complete semantic definition of $\mathcal{ES}$: Given a model $M = \langle e, w \rangle$ and sequence of actions σ, let

1. $e, w, \sigma \models f(r_1, \ldots, r_k)$ iff $w[\sigma{:}f(r_1, \ldots, r_k)] = 1$;
2. $e, w, \sigma \models (r_1 = r_2)$ iff r_1 and r_2 are identical;
3. $e, w, \sigma \models (\alpha \wedge \beta)$ iff $e, w, \sigma \models \alpha$ and $e, w, \sigma \models \beta$;
4. $e, w, \sigma \models \neg\alpha$ iff $e, w, \sigma \not\models \alpha$;
5. $e, w, \sigma \models \forall x.\, \alpha$ iff $e, w, \sigma \models \alpha^x_r$, for every $r \in R$;
6. $e, w, \sigma \models [r]\alpha$ iff $e, w, \sigma \cdot r \models \alpha$;
7. $e, w, \sigma \models \Box\alpha$ iff $e, w, \sigma \cdot \sigma' \models \alpha$, for every $\sigma' \in R^*$;
8. $e, w, \sigma \models \mathit{Know}(\alpha)$ iff for all $w' \simeq_\sigma w$, if $w' \in e$ then $e, w', \sigma \models \alpha$;
9. $e, w, \sigma \models \mathit{OKnow}(\alpha)$ iff for all $w' \simeq_\sigma w$, $w' \in e$ iff $e, w', \sigma \models \alpha$;

When α is a sentence, we sometimes write $e, w \models \alpha$ instead of $e, w, \langle\,\rangle \models \alpha$. In addition, when α is *objective*, we write $w \models \alpha$ and when α is *subjective*, we write $e \models \alpha$. When Σ is a set of sentences and α is a sentence, we write $\Sigma \models \alpha$ (read: Σ logically entails α) to mean that for every e and w, if $e, w \models \alpha'$ for every $\alpha' \in \Sigma$, then $e, w \models \alpha$. Finally, we write $\models \alpha$ (read: α is valid) to mean $\{\} \models \alpha$.

Basic Action Theories and Regression

Let us now consider the equivalent of basic action theories of the situation calculus. Since in our logic there is no explicit notion of situations and the uniqueness of names is built into the semantics, our basic action theories do not require foundational axioms. For now we only consider the objective (non-epistemic) case.

Given a set of fluent predicates $\mathcal{F}$, a set $\Sigma \subseteq \mathcal{ES}$ of sentences is called a *basic action theory* over $\mathcal{F}$ iff $\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}}$ where Σ mentions only fluents in $\mathcal{F}$ and

1. Σ_0 is any set of fluent sentences;
2. Σ_{pre} is a singleton sentence of the form $\Box \mathit{Poss}(a) \equiv \pi$, where π is a fluent formula;[3]
3. Σ_{post} is a set of sentences of the form $\Box[a]f(\vec{x}) \equiv \gamma_f$, one for each fluent $f \in \mathcal{F}$, and where γ_f is a fluent formula.[4]

The idea here is that Σ_0 expresses what is true initially (in the initial situation), Σ_{pre} is one large precondition axiom, and Σ_{post} is a set of successor state axioms, one per fluent, which incorporate the solution to the frame problem proposed by Reiter (Reiter 1991).

Here is an example basic action theory from the blocks world. There are three fluents, *Fragile*(x) (object x is fragile), *Holding*(x) (object x is being held by some unnamed robot), and *Broken*(x) (object x is broken), and three actions, *drop*(x), *pickup*(x), *repair*(x). The initial theory Σ_0 consists of the following two sentences:

$$\neg \mathit{Broken}(\mathit{obj5}),$$
$$\forall z \neg \mathit{Holding}(z).$$

This says that initially *obj5* is not broken and the robot is not holding anything. The precondition axiom Σ_{pre} is the following:

$$\begin{aligned}
&\Box \mathit{Poss}(a) \equiv \\
&\quad \exists x. a = \mathit{pickup}(x) \wedge \forall z. \neg \mathit{Holding}(z)) \vee \\
&\quad \exists x. a = \mathit{drop}(x) \wedge \mathit{Holding}(x) \vee \\
&\quad \exists x. a = \mathit{repair}(x) \wedge \mathit{Holding}(x) \wedge \mathit{Broken}(x).
\end{aligned}$$

This says that a pickup action is possible if the robot is not holding anything, that a drop action is possible if the object in question is held, and that a repair action is possible if the object is both held and broken. Note the use of $\Box$ here, which plays the role of the universally quantified situation variable in the situation calculus, ensuring that these preconditions hold after any sequence of actions. The set of successor state axioms Σ_{post} in the example has the following three elements:

$$\begin{aligned}
&\Box[a]\mathit{Holding}(x) \equiv \\
&\quad a = \mathit{pickup}(x) \vee \mathit{Holding}(x) \wedge a \neq \mathit{drop}(x), \\
&\Box[a]\mathit{Fragile}(x) \equiv \mathit{Fragile}(x), \\
&\Box[a]\mathit{Broken}(x) \equiv \\
&\quad a = \mathit{drop}(x) \wedge \mathit{Fragile}(x) \vee \\
&\quad \mathit{Broken}(x) \wedge a \neq \mathit{repair}(x).
\end{aligned}$$

This tells us precisely under what conditions each of the three fluents is affected by doing an action: an object is held iff it was just picked up or was already held and not dropped; the fragility of an object is unaffected by any action; an object is broken iff it was just dropped and fragile, or it was already broken and not just repaired. Note that the solution

[3] We follow the usual situation calculus convention that free variables are universally quantified from the outside. We also assume that $\Box$ has lower syntactic precedence than the logical connectives, so that $\Box \mathit{Poss}(a) \equiv \pi$ stands for $\forall a. \Box(\mathit{Poss}(a) \equiv \pi)$.

[4] The $[t]$ construct has higher precedence than the logical connectives. So $\Box[a]f(\vec{x}) \equiv \gamma_f$ abbreviates $\forall a. \Box([a]f(\vec{x}) \,.\equiv\, \gamma_f)$.

to the frame problem depends on the universally quantified variable a, but the unique-name aspect is built into the semantics of the language.

For a given basic action theory Σ, a fundamental reasoning task is *projection*, that is, determining what holds after a number of actions have occurred. For example, is *obj5* broken after first picking it up and then dropping it? Formally, this corresponds to determining if

$$\Sigma \models [\mathit{pickup}(\mathit{obj5})][\mathit{drop}(\mathit{obj5})]\mathit{Broken}(\mathit{obj5}).$$

It is not hard to see that this conclusion does not follow from the action theory above (since it is left open whether or not *obj5* is fragile). In general, the projection task involves determining if

$$\Sigma \models [r_1]\ldots[r_k]\alpha,$$

where Σ is a basic action theory, the r_i are ground terms (representing actions), and α is an arbitrary sentence. Reiter showed how successor state axioms allow the use of *regression* to solve this reasoning task for certain α (which he called the *regressable formulas*). The idea is to successively replace fluents in α by the right-hand side of their successor state axioms until the resulting sentence contains no more actions, at which point one need only check whether that sentence follows from the sentences in the initial theory.

We remark that, although the projection problem is defined for linear sequences of actions, a solution such as regression also allows us to reason about conditional plans. For example, verifying whether a conditional plan is guaranteed to satisfy a goal α amounts to determining whether α holds at the end of every branch of the conditional.[5]

We now show how to do regression in $\mathcal{ES}$ given an basic action theory Σ. In our account, any bounded, objective sentence α is considered regressable, and we define $\mathcal{R}[\alpha]$, *the regression of* α *wrt* Σ, to be the fluent formula $\mathcal{R}[\langle\rangle, \alpha]$, where for any sequence of terms σ (not necessarily ground), $\mathcal{R}[\sigma, \alpha]$ is defined inductively on α by:

1. $\mathcal{R}[\sigma, (t_1 = t_2)] = (t_1 = t_2)$;
2. $\mathcal{R}[\sigma, \forall x\alpha] = \forall x\mathcal{R}[\sigma, \alpha]$;
3. $\mathcal{R}[\sigma, (\alpha \wedge \beta)] = (\mathcal{R}[\sigma, \alpha] \wedge \mathcal{R}[\sigma, \beta])$;
4. $\mathcal{R}[\sigma, \neg\alpha] = \neg\mathcal{R}[\sigma, \alpha]$;
5. $\mathcal{R}[\sigma, [t]\alpha] = \mathcal{R}[\sigma \cdot t, \alpha]$;
6. $\mathcal{R}[\sigma, \mathit{Poss}(t)] = \mathcal{R}[\sigma, \pi^a_t]$;
7. $\mathcal{R}[\sigma, f(t_1, \ldots, t_k)]$ is defined inductively on σ by:
   (a) $\mathcal{R}[\langle\rangle, f(t_1, \ldots, t_k)] = f(t_1, \ldots, t_k))$;
   (b) $\mathcal{R}[\sigma \cdot t, f(t_1, \ldots, t_k)] = \mathcal{R}[\sigma, (\gamma_f)^{a\,x_1}_{t\,t_1} \cdots ^{x_k}_{t_k}]$.

Note that this definition uses the right-hand sides of both the precondition and successor state axioms from Σ.

Using the semantics of $\mathcal{ES}$, we will now reprove Reiter's Regression Theorem, and show that it is possible to reduce reasoning with formulas that contain $[t]$ operators to reasoning with fluent formulas in the initial state.

We begin by defining for any world w and basic action theory Σ another world w_Σ which is like w except that it is defined to satisfy the Σ_{pre} and Σ_{post} sentences of Σ.

[5] In the case of Golog, regression is applied even to plans with loops and nondeterministic action choices.

Definition 1 *Let w be a world and Σ a basic action theory with fluent predicates $\mathcal{F}$. Then w_Σ is a world satisfying the following conditions:*

1. *for $f \notin \mathcal{F}$, $w_\Sigma[\sigma{:}f(r_1, \ldots, r_k)] = w[\sigma{:}f(r_1, \ldots, r_k)]$;*
2. *for $f \in \mathcal{F}$, $w_\Sigma[\sigma{:}f(r_1, \ldots, r_k)]$ is defined inductively:*
   (a) $w_\Sigma[\langle\rangle{:}f(r_1, \ldots, r_k)] = w[\langle\rangle{:}f(r_1, \ldots, r_k)]$;
   (b) $w_\Sigma[\sigma \cdot r{:}f(r_1, \ldots, r_k)] = 1$ iff $w_\Sigma, \sigma \models (\gamma_f)^{a\,x_1}_{r\,r_1} \cdots ^{x_k}_{r_k}$.
3. *$w_\Sigma[\sigma{:}\mathit{Poss}(r)] = 1$ iff $w_\Sigma, \sigma \models \pi^a_r$.*

Note that this again uses the π and γ_f formulas from Σ. Then we get the following simple lemmas:

Lemma 1 *For any w, w_Σ exists and is uniquely defined.*

Proof: w_Σ clearly exists. The uniqueness follows from the fact that π is a fluent formula and that for all $f \in \mathcal{F}$, once the initial values of f are fixed, then the values after any number of actions are uniquely determined by Σ_{post}. ∎

Lemma 2 *If $w \models \Sigma_0$ then $w_\Sigma \models \Sigma$.*

Proof: Directly from the definition of w_Σ, we have that $w_\Sigma \models \forall a\Box\mathit{Poss}(a) \equiv \pi$ and $w_\Sigma \models \forall a\forall\vec{x}\Box[a]f(\vec{x}) \equiv \gamma_f$. ∎

Lemma 3 *If $w \models \Sigma$ then $w = w_\Sigma$.*

Proof: If $w \models \Box\mathit{Poss}(a) \equiv \pi$ and $w \models \Box[a]f(\vec{x}) \equiv \gamma_f$, then w satisfies the definition of w_Σ. ∎

Lemma 4 *Let α be any bounded, objective sentence. Then $w \models \mathcal{R}[\sigma, \alpha]$ iff $w_\Sigma, \sigma \models \alpha$.*

Proof: The proof is by induction on the length of α (treating the length of $\mathit{Poss}(t)$ as the length of π^a_t plus 1). The only tricky case is for $\mathit{Poss}(r)$ and for fluent atoms. We have that $w_\Sigma, \sigma \models \mathit{Poss}(r)$ iff (by definition of w_Σ) $w_\Sigma, \sigma \models \pi^a_r$ iff (by induction) $w \models \mathcal{R}[\sigma, \pi^a_r]$ iff (by definition of $\mathcal{R}$) $w \models \mathcal{R}[\sigma, \mathit{Poss}(r)]$. Finally, we consider fluent atoms, and prove the lemma by a sub-induction on σ:

1. $w_\Sigma, \langle\rangle \models f(r_1, \ldots, r_k)$ iff (by definition of w_Σ),
   $w, \langle\rangle \models f(r_1, \ldots, r_k)$ iff (by definition of $\mathcal{R}$),
   $w \models \mathcal{R}[\langle\rangle, f(r_1, \ldots, r_k)]$;
2. $w_\Sigma, \sigma \cdot r \models f(r_1, \ldots, r_k)$ iff (by definition of w_Σ),
   $w_\Sigma, \sigma \models (\gamma_f)^{a\,x_1}_{r\,r_1} \cdots ^{x_k}_{r_k}$ iff (by the sub-induction),
   $w \models \mathcal{R}[\sigma, (\gamma_f)^{a\,x_1}_{r\,r_1} \cdots ^{x_k}_{r_k}]$ iff (by definition of $\mathcal{R}$),
   $w \models \mathcal{R}[\sigma \cdot r, f(r_1, \ldots, r_k)]$,

which completes the proof. ∎

Theorem 1 *Let $\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}}$ be a basic action theory and let α be an objective, bounded sentence. Then $\mathcal{R}[\alpha]$ is a fluent sentence and satisfies*

$$\Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \models \alpha \quad \textit{iff} \quad \Sigma_0 \models \mathcal{R}[\alpha].$$

Proof: Suppose $\Sigma_0 \models \mathcal{R}[\alpha]$. We prove that $\Sigma \models \alpha$. Let w be any world such that $w \models \Sigma$. Then, $w \models \Sigma_0$, and so $w \models \mathcal{R}[\alpha]$. By Lemma 4, $w_\Sigma \models \alpha$. By Lemma 3, $w_\Sigma = w$, and so $w \models \alpha$.

Conversely, suppose $\Sigma \models \alpha$. We prove that $\Sigma_0 \models \mathcal{R}[\alpha]$. Let w be any world such that $w \models \Sigma_0$. From Lemma 2, $w_\Sigma \models \Sigma$, and so $w_\Sigma \models \alpha$. By Lemma 4, $w \models \mathcal{R}[\alpha]$. ∎

Note that the conciseness of this proof depends crucially on the fact that Lemma 4 is proven by induction over *sentences*, which is possible only because quantification is interpreted substitutionally.

Knowledge

The interpretation of knowledge in $\mathcal{ES}$ is just a special case of possible-world semantics (Kripke 1963; Hintikka 1962). In particular, as we model knowledge as a set of "worlds", it is not surprising that we obtain the usual properties of *weak S5* (Fagin et al. 1995). Since we assume a fixed universe of discourse, the Barcan formula for knowledge (Property 4 of the following theorem) and its existential version (Property 5) hold as well. Moreover, these properties hold after any number of actions have been performed.

Theorem 2

1. $\models \Box(\mathit{Know}(\alpha) \wedge \mathit{Know}(\alpha \supset \beta) \supset \mathit{Know}(\beta))$;
2. $\models \Box(\mathit{Know}(\alpha) \supset \mathit{Know}(\mathit{Know}(\alpha)))$;
3. $\models \Box(\neg\mathit{Know}(\alpha) \supset \mathit{Know}(\neg\mathit{Know}(\alpha)))$;
4. $\models \Box(\forall x.\mathit{Know}(\alpha) \supset \mathit{Know}(\forall x.\alpha))$;
5. $\models \Box(\exists x.\mathit{Know}(\alpha) \supset \mathit{Know}(\exists x.\alpha))$.

Proof:

1. Let $e, w, \sigma \models \mathit{Know}(\alpha) \wedge \mathit{Know}(\alpha \supset \beta)$. Then for all $w' \simeq_\sigma w$, if $w' \in e$ then $e, w', \sigma \models \alpha$ and $e, w', \sigma \models (\alpha \supset \beta)$. Hence, $e, w', \sigma \models \beta$ and, therefore, we have that $e, w, \sigma \models \mathit{Know}(\beta)$.
2. Let $e, w, \sigma \models \mathit{Know}(\alpha)$. Let w' and w'' be worlds in e such that $w' \simeq_\sigma w$ and $w'' \simeq_\sigma w'$. Since $\simeq_\sigma$ is an equivalence relation, we have $w'' \simeq_\sigma w$ and, therefore, $e, w'', \sigma \models \alpha$ by assumption. As this is true for all $w'' \in e$ with $w'' \simeq_\sigma w'$, we have $e, w', \sigma \models \mathit{Know}(\alpha)$ and, hence, $e, w, \sigma \models \mathit{Know}(\mathit{Know}(\alpha))$.
3. Let $e, w, \sigma \models \neg\mathit{Know}(\alpha)$. Thus for some w', $w' \simeq_\sigma w$, $w' \in e$ and $e, w', \sigma \not\models \alpha$. Let w'' be any world such that $w'' \simeq_\sigma w'$ and $w'' \in e$. Clearly, $e, w'', \sigma \models \neg\mathit{Know}(\alpha)$. Since $w'' \simeq_\sigma w$, $e, w, \sigma \models \mathit{Know}(\neg\mathit{Know}(\alpha))$ follows.
4. Let $e, w, \sigma \models \forall x.\mathit{Know}(\alpha)$. Hence for all $r \in R$, $e, w, \sigma \models \mathit{Know}(\alpha^x_r)$ and thus for all $w' \simeq_\sigma w$, if $w' \in e$ then for all $r \in R$, $e, w, \sigma \models \alpha^x_r$, from which $e, w, \sigma \models \mathit{Know}(\forall x.\alpha)$ follows.
5. Let $e, w, \sigma \models \exists x.\mathit{Know}(\alpha)$. Then $e, w, \sigma \models \mathit{Know}(\alpha^x_r)$ for some $r \in R$. By the definition of *Know*, it follows that $e, w, \sigma \models \mathit{Know}(\exists x.\alpha)$. ∎

We remark that the converse of the Barcan formula (Property 4) holds as well. However, note that this is not the case for Property 5: $\Box(\mathit{Know}(\exists x.\alpha) \supset \exists x.\mathit{Know}(\alpha))$ is not valid in general. Despite the fact that quantification is understood substitutionally, knowing that someone satisfies α does not entail knowing who that individual is, just as it should be.

Perhaps more interestingly, we can show a generalized version of the determinacy of knowledge:

Theorem 3 *Suppose α is an objective sentence and β is an objective formula with one free variable x, such that $\models \mathit{Know}(\alpha) \supset \exists x.\mathit{Know}(\beta)$. Then for some rigid term r, $\models \mathit{Know}(\alpha) \supset \mathit{Know}(\beta^x_r)$.*

Proof: Suppose not. Then for every r, $\mathit{Know}(\alpha)$ does not entail $\mathit{Know}(\beta^x_r)$, and so, by the Lemma below, α does not entail β^x_r. So for every r, there is a world w_r such that $w_r \models (\alpha \wedge \neg\beta^x_r)$. Let $e = \{w_r \mid r \in R\}$. Then we have that $e \models \mathit{Know}(\alpha)$ and for every $r \in R$, $e \models \neg\mathit{Know}(\beta^x_r)$, and so $e \models \forall x.\neg\mathit{Know}(\beta^x_r)$. This contradicts the fact that $\mathit{Know}(\alpha)$ entails $\exists x.\mathit{Know}(\beta)$. ∎

Lemma 5 *If α and β are objective, and $\models (\alpha \supset \beta)$, then $\models (\mathit{Know}(\alpha) \supset \mathit{Know}(\beta))$.*

Proof: Suppose that some $e \models \mathit{Know}(\alpha)$. Then for every $w \in e$, $w \models \alpha$. Then for every $w \in e$, $w \models \beta$. Thus $e \models \mathit{Know}(\beta)$. ∎

This proof is exactly as it would be in $\mathcal{OL}$. Again it is worth noting that the proof of this theorem in the ordinary situation calculus (for the simpler case involving disjunction rather than existential quantification) is a multi-page argument involving Craig's Interpolation Lemma.

Regressing Knowledge

In the previous section we introduced basic action theories as representations of dynamic domains. With knowledge, we need to distinguish between what is true in the world and what the agent knows or believes about the world. Perhaps the simplest way to model this is to have two basic action theories Σ and Σ', where Σ is our account of how the world is and will change as the result of actions, and Σ' is the agent's version of the same. The corresponding epistemic state is then simply $\{w \mid w \models \Sigma'\}$, which we also denote as $\Re[\![\Sigma']\!]$. It is easy to see that

Lemma 6 $\Re[\![\Sigma]\!], w \models \mathit{OKnow}(\Sigma)$.

Proof: Let w' be any world. Then $w' \simeq_{\langle\rangle} w$ by the definition of $\simeq_\sigma$. By the definition of $\Re[\![\Sigma]\!]$ we have that $w' \in \Re[\![\Sigma]\!]$ iff $w' \models \alpha$. Hence $\Re[\![\Sigma]\!] \models \mathit{OKnow}(\Sigma)$. ∎

As discussed in (Scherl and Levesque 2003), actions can be divided into ordinary actions which change the world like $pickup(\mathit{obj5})$ and knowledge-producing or sensing actions such as sensing the color of a litmus paper to test the acidity of a solution. To model the outcome of these sensing actions, we extend our notion of a basic action theory to be

$$\Sigma = \Sigma_0 \cup \Sigma_{\mathrm{pre}} \cup \Sigma_{\mathrm{post}} \cup \Sigma_{\mathrm{sense}},$$

where Σ_{sense} is a singleton sentence exactly parallel to the one for *Poss* of the form

$$\Box \mathit{SF}(a) \equiv \varphi$$

where φ is a fluent formula. For example, assume we have a sensing action *seeRed* which tells the agent whether or not the *Red* fluent is true (that is, some nearby litmus paper is red), and that no other action returns any useful sensing result. In that case, Σ_{sense} would be the following:

$$\Box \mathit{SF}(a) \;\equiv\; [a = \mathit{seeRed} \wedge \mathit{Red} \vee a \neq \mathit{seeRed}].$$

For ease of formalization, we assume that *SF* is characterized for all actions including ordinary non-sensing ones, for which we assume that *SF* is vacuously true.[6]

The following theorem can be thought of as a successor-state axiom for knowledge, which will allow us to extend regression to formulas containing *Know*. Note that, in contrast to the successor state axioms for fluents, this is a *theorem* of the logic not a stipulation as part of a basic action theory:

Theorem 4 $\models \Box[a]\mathit{Know}(\alpha) \equiv$
$$\mathit{SF}(a) \wedge \mathit{Know}(\mathit{SF}(a) \supset [a]\alpha) \quad \vee$$
$$\neg\mathit{SF}(a) \wedge \mathit{Know}(\neg\mathit{SF}(a) \supset [a]\alpha).$$

Proof: Let $e, w, \sigma \models [r]\mathit{Know}(\alpha^a_r)$ for $r \in R$. We write α' for α^a_r. Suppose $e, w, \sigma \models \mathit{SF}(r)$. (The case where $e, w, \sigma \models \neg\mathit{SF}(r)$ is analogous.) It suffices to show that $e, w, \sigma \models \mathit{Know}(\mathit{SF}(r) \supset [r]\alpha')$. So suppose $w' \simeq_\sigma w$ and $w' \in e$. Thus $w'[\sigma : \mathit{SF}(r)] = w[\sigma : \mathit{SF}(r)] = 1$ by assumption, that is, $w' \simeq_{\sigma\cdot r} w$. Since $e, w, \sigma \models [r]\mathit{Know}(\alpha')$ by assumption, $e, w', \sigma \cdot r \models \alpha'$, from which $e, w', \sigma \models [r]\alpha'$ follows.

Conversely, let $e, w, \sigma \models \mathit{SF}(r) \wedge [r]\mathit{Know}(\mathit{SF}(r) \supset \alpha')$. (The other case is similar.) We need to show that $e, w, \sigma \models [r]\mathit{Know}(\alpha')$, that is, $e, w, \sigma \cdot r \models \mathit{Know}(\alpha')$. Let $w' \simeq_{\sigma\cdot r} w$ and $w' \in e$. Then $w'[\sigma : \mathit{SF}(r)] = w[\sigma : \mathit{SF}(r)] = 1$ by assumption. Hence $e, w', \sigma \models \mathit{SF}(r)$. Therefore, by assumption, $e, w', \sigma \cdot r \models \alpha'$, from which $e, w, \sigma \models [r]\mathit{Know}(\alpha')$ follows. ∎

We consider this a successor state axiom for knowledge in the sense that it tells us for any action a what will be known after doing a in terms of what was true before. In this case, knowledge after a depends on what was known before doing a about what the future would be like after doing a, contingent on the sensing information provided by a. Unlike (Scherl and Levesque 2003), this is formalized without a fluent for the knowledge accessibility relation, which would have required situation terms in the language.

We are now ready to extend regression to deal with knowledge. Instead of being defined relative to a basic action theory Σ, the regression operator $\mathcal{R}$ will be defined relative to a *pair* of basic action theories $\langle\Sigma', \Sigma\rangle$ where, as above, Σ' represents the beliefs of the agent. We allow Σ and Σ' to differ arbitrarily and indeed to contradict each other, so that agents may have false beliefs about what the world is like, including its dynamics.[7] The idea is to regress *wrt.* Σ outside of *Know* operators and *wrt.* Σ' inside. To be able to distinguish between these cases, $\mathcal{R}$ now carries the two basic action theories with it as extra arguments.

Rule 1–7 of the new regression operator $\mathcal{R}$ are exactly as before except for the extra arguments Σ' and Σ. Then we add the following:

8. $\mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{SF}(t)] = \mathcal{R}[\Sigma', \Sigma, \sigma, \varphi^a_t]$;
9. $\mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{Know}(\alpha)]$ is defined inductively on σ by:
   (a) $\mathcal{R}[\Sigma', \Sigma, \langle\rangle, \mathit{Know}(\alpha)] = \mathit{Know}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$;
   (b) $\mathcal{R}[\Sigma', \Sigma, \sigma \cdot t, \mathit{Know}(\alpha)] = \mathcal{R}[\Sigma', \Sigma, \sigma, \beta^a_t]$, where β is the right-hand side of the equivalence in Theorem 4.

For simplicity, we write $\mathcal{R}[\alpha]$ instead of $\mathcal{R}[\Sigma', \Sigma, \langle\rangle, \alpha]$.

To prove the regression theorem for formulas involving *Know*, we first need to extend the definition of w_Σ of the previous section to account for *SF*-atoms. For any world w let w_Σ be as in Definition 1 with the additional constraint:

4. $w_\Sigma[\sigma{:}\mathit{SF}(r)] = 1$ iff $w_\Sigma, \sigma \models \varphi^a_r$.

As before, since φ is a fluent formula, w_Σ is uniquely defined for any w. It is easy to see that Lemma 2 and 3 carry over to basic action theories extended by a Σ_{sense} formula. In the following we simply refer to the original lemmas with the understanding that they apply to the extended basic action theories as well. We also assume for Σ (resp. Σ'), a basic action theory, that Σ_0 (resp. Σ'_0) is the sub-theory about the initial state of the world.

Here is the extension of Lemma 2 to knowledge:

Lemma 7 *If* $e \models \mathit{OKnow}(\Sigma_0)$ *then* $e_\Sigma \models \mathit{OKnow}(\Sigma)$.

Proof: Let $e \models \mathit{OKnow}(\Sigma_0)$, that is, for all w, $w \in e$ iff $w \models \Sigma_0$. We need to show that for all w, $w \in e_\Sigma$ iff $w \models \Sigma$.

Suppose $w \models \Sigma$. Then $w \models \Sigma_0$ and hence $w \in e$ and, by definition, $w_\Sigma \in e_\Sigma$. By Lemma 3, $w_\Sigma = w$ and, therefore, $w \in e_\Sigma$.

Conversely, let $w \in e_\Sigma$. By definition, there is a $w' \in e$ such that $w = w'_\Sigma$. Since $w' \models \Sigma_0$, by Lemma 2, $w'_\Sigma \models \Sigma$, that is, $w \models \Sigma$. ∎

We now turn to the generalization of Lemma 4 for knowledge. Given any epistemic state e and any basic action theory Σ, we first define $e_\Sigma = \{w_\Sigma \mid w \in e\}$.

Lemma 8 $e, w \models \mathcal{R}[\Sigma', \Sigma, \sigma, \alpha]$ *iff* $e_{\Sigma'}, w_\Sigma, \sigma \models \alpha$.

Proof: The proof is by induction on σ with a sub-induction on α.

Let $\sigma = \langle\rangle$. As with the case of *Poss* in Lemma 4, we take the length of $\mathit{SF}(r)$ to be the length of φ^a_r plus 1. The proof for *Poss*, fluent atoms, and the connectives $\neg$, $\wedge$, and $\forall$ is exactly analogous to Lemma 4.

For *SF*, we have the following:

$e_{\Sigma'}, w_\Sigma, \langle\rangle \models \mathit{SF}(r)$ iff (by the definition of w_Σ),
$e_{\Sigma'}, w_\Sigma, \langle\rangle \models \varphi^a_r$ iff (by induction),
$e, w \models \mathcal{R}[\Sigma', \Sigma, \langle\rangle, \varphi^a_r]$ iff (by the definition of $\mathcal{R}$),
$e, w \models \mathcal{R}[\Sigma', \Sigma, \langle\rangle, \mathit{SF}(r)]$.

For formulas $\mathit{Know}(\alpha)$ we have:

$e_{\Sigma'} \models \mathit{Know}(\alpha)$ iff
for all $w \in e_{\Sigma'}$, $e_{\Sigma'}, w \models \alpha$ iff (by definition of $e_{\Sigma'}$),
for all $w \in e$, $e_{\Sigma'}, w_{\Sigma'} \models \alpha$ iff (by induction),
for all $w \in e$, $e, w, \models \mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha]$ iff
$e \models \mathit{Know}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$ iff (by definition of $\mathcal{R}$),
$e \models \mathcal{R}[\Sigma', \Sigma, \langle\rangle, \mathit{Know}(\alpha)]$.

This concludes the base case $\sigma = \langle\rangle$.

Now consider the case of $\sigma \cdot r$, which again is proved by a sub-induction on α. The proof is exactly like the sub-induction for the base case except for *Know*, for which we have the following:

[6] Here we restrict ourselves to sensing truth values. See (Scherl and Levesque 2003) for how to handle arbitrary values.

[7] This is like (Lakemeyer and Levesque 1998) but in contrast to Scherl and Levesque (Scherl and Levesque 2003), who can only handle true belief. While we allow for false beliefs, we continue to use the terms knowledge and belief interchangeably.

$e_{\Sigma'}, w_\Sigma, \sigma \cdot r \models Know(\alpha)$ iff (by Theorem 4),
$e_{\Sigma'}, w_\Sigma, \sigma \models \beta^a_r$ (where the β is from Theorem 4)
iff (by the main induction),
$e, w \models \mathcal{R}[\Sigma', \Sigma, \sigma, \beta^a_r]$ iff (by definition of $\mathcal{R}$),
$e, w \models \mathcal{R}[\Sigma', \Sigma, \sigma \cdot r, Know(\alpha)]$,

which completes the proof. ∎

Finally, here is the general regression theorem:

Theorem 5 *Let α be a bounded sentence with no OKnow operators. Then $\mathcal{R}[\alpha]$ is a static sentence and satisfies*

$$\Sigma \wedge OKnow(\Sigma') \models \alpha \quad \textit{iff} \quad \Sigma_0 \wedge OKnow(\Sigma'_0) \models \mathcal{R}[\alpha].$$

Proof: To prove the only-if direction, let us suppose that $\Sigma \wedge OKnow(\Sigma') \models \alpha$ and that $e, w \models \Sigma_0 \wedge OKnow(\Sigma'_0)$. Thus $w \models \Sigma_0$ and, by Lemma 2, $w_\Sigma \models \Sigma$. Also, $e \models OKnow(\Sigma'_0)$ and thus, by Lemma 7, $e_{\Sigma'} \models OKnow(\Sigma')$. Therefore, $e_{\Sigma'}, w_\Sigma \models \Sigma \wedge OKnow(\Sigma')$. By assumption, $e_{\Sigma'}, w_\Sigma \models \alpha$ and, by Lemma 8, $e, w \models \mathcal{R}[\alpha]$.

Conversely, suppose $\Sigma_0 \wedge OKnow(\Sigma'_0) \models \mathcal{R}[\alpha]$ and let $e, w \models \Sigma \wedge OKnow(\Sigma')$. (Note that e is unique as $e = \Re[\![\Sigma']\!]$ by Lemma 6.) Then $w \models \Sigma_0$. Now suppose $e' \models OKnow(\Sigma'_0)$. Then, by assumption, $e', w \models \mathcal{R}[\alpha]$. Then $e'_{\Sigma'}, w_\Sigma \models \alpha$. By Lemma 7, $e'_{\Sigma'} \models OKnow(\Sigma')$. By Lemma 3, $w_\Sigma = w$ and, by the uniqueness of e, $e'_{\Sigma'} = e$. Therefore, $e, w \models \alpha$. ∎

The reader will have noticed that we left out *OKnow* from our definition of regression. While it seems perfectly reasonable to ask what one only knows after doing an action a, it is problematic to deal with for at least two reasons. For one, the current language does not seem expressive enough to represent what is only-known in non-initial situations. Roughly, this is because after having done a one knows that one has just done a and that certain things held before the action. However, the language does not allow us to refer to previous situations. But even if were able to extend the language to deal with this issue, another problem is that whenever $[a]OKnow(\alpha)$ holds, the α in question would in general not be regressable. To see why, recall that regressable formulas are restricted to be *bounded*, that is, they do not mention $\Box$. In our discussion above we make the reasonable assumption that, initially, the agent only-knows a basic action theory Σ', which contains sentences like successor state axioms, which are not bounded. Then, whatever is only-known after doing a must somehow still refer to these axioms, but they are not regressable by our definition. For these reason, we have nothing to say about only-knowing in non-initial situations.

As a consolation, being able to reason about what is known in future states, as opposed to only-known, seems to be sufficient for most practical purposes.

An Example

To illustrate how regression works in practice, let us consider the litmus-test example adapted from (Scherl and Levesque 2003). In addition to the action *seeRed*, which we described above as sensing whether or not the fluent *Red* is true (the litmus paper is red), there is a second action *dipLitmus* (dipping the litmus paper into the solution), which makes *Red* true just in case the solution is acidic, represented by the fluent *Acid*. For this example, we will use the Σ_{sense} from above, and $\Sigma_{\text{pre}} = \{\Box Poss(a) \equiv true\}$, which states that all actions are always possible, for simplicity. We let Σ_{post}, the successor state axioms, be the following:

$$\begin{aligned} &\Box[a]Acid \;\equiv\; Acid, \\ &\Box[a]Red \;\equiv \\ &\quad a = dipLitmus \wedge Acid \;\vee \\ &\quad Red \wedge a \neq dipLitmus, \end{aligned}$$

that is, the acidity of the solution is unaffected by any action, and the litmus paper is red iff the last action was to dip it into an acidic solution, or it was already red and was not dipped. Finally, we let Σ_0, the initial theory, be the following:

$$Acid, \neg Red.$$

Now let us consider two basic action theories:

$$\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}} \quad \text{and}$$

$$\Sigma' = \{\} \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}.$$

The two are identical except that *Acid* is true and *Red* false initially in Σ. This amounts to saying that in reality the solution is acidic and the litmus paper is initially not red (in Σ), but that the agent has no knowledge about the initial state of the two fluents (in Σ'). Then we get the following:

1. $\Sigma \wedge OKnow(\Sigma') \models \neg Know(Acid)$;
2. $\Sigma \wedge OKnow(\Sigma') \models [dipLitmus]\neg Know(Acid)$;
3. $\Sigma \wedge OKnow(\Sigma') \models [dipLitmus][seeRed]Know(Acid)$.

In other words, after first dipping the litmus and then sensing the result, the agent comes to know not only that the litmus paper is red but that the solution is acidic. Informally, what happens is this: because *Acid* is true in reality, the *dipLitmus* action makes *Red* true; the agent knows that neither *Red* nor *Acid* are affected by *seeRed*, and so knows that if *Red* was made true by *dipLitmus* (because *Acid* was true), then both will be true after *seeRed*; after doing the *seeRed*, the agent learns that *Red* was indeed true, and so *Acid* was as well.

Observe that the agent only comes to these beliefs after doing both actions. (1.) and (2.) show the usefulness of only-knowing. In particular, $\neg Know(Acid)$ would *not* be entailed if we replaced *OKnow* by *Know* in the antecedent.

To see why (1.) holds, notice that $\mathcal{R}[\neg Know(Acid)] = \neg Know(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, Acid]) = \neg Know(Acid)$. Therefore, by Theorem 5, we get that (1.) reduces to

$$\Sigma_0 \wedge OKnow(true) \models \neg Know(Acid).$$

The entailment clearly holds because the set of all worlds e_0 is the unique epistemic state satisfying $OKnow(true)$, and e_0 contains worlds where *Acid* is false.

To see why (2.) holds, first note that

$$\mathcal{R}[[dipLitmus]\neg Know(Acid)] = \neg\mathcal{R}[\Sigma', \Sigma, d, Know(Acid)],$$

where we abbreviate *dipLitmus* as d. Then, using Rule (9b),

$$\begin{aligned} &\neg\mathcal{R}[\Sigma', \Sigma, d, Know(Acid)] = \\ &\quad \neg\mathcal{R}[\Sigma', \Sigma, \langle\rangle, SF(d) \wedge Know(SF(d) \supset [d]Acid) \vee \\ &\qquad \neg SF(d) \wedge Know(\neg SF(d) \supset [d]Acid)]. \end{aligned}$$

The right-hand side of the equality reduces to $\neg$*Know*(*Acid*) because both $\mathcal{R}[\Sigma', \Sigma, \langle\,\rangle, SF(d)]$ and $\mathcal{R}[\Sigma', \Sigma', \langle\,\rangle, SF(d)]$ reduce to *true* and $\mathcal{R}[\Sigma', \Sigma', d, Acid] = Acid$. Hence (2.) also reduces to

$$\Sigma_0 \wedge OKnow(true) \models \neg Know(Acid),$$

which was shown to hold above.

Finally, (3.) holds because

$$\mathcal{R}[\,[dipLitmus][seeRed]Know(Acid)\,]$$

reduces to $Acid \wedge Know(Acid \supset Acid) \vee \neg Acid \wedge Know(Acid)$, which again follows from $\Sigma_0 \wedge OKnow(true)$.

While regression allows us to reduce questions about knowledge and action to questions about knowledge alone, in the next section we go even further and replace reasoning about knowledge by classical first-order reasoning.

$\mathcal{OL}$ is part of $\mathcal{ES}$

If we restrict ourselves to static formulas without occurrences of *Poss* or *SF*, and where the only rigid terms are standard names (rigid terms from G^0), we obtain precisely the language $\mathcal{OL}$ of (Levesque 1990; Levesque and Lakemeyer 2001).[8] We call such formulas and sentences $\mathcal{OL}$-formulas and $\mathcal{OL}$-sentences, respectively.

For example, if n is a standard name and f_1 and f_2 are fluent predicates,

$\forall x. f_1(x) \supset Know(f_1(x))$ and
$OKnow(f_1(n)) \supset Know(\neg Know(f_2(n)))$

are $\mathcal{OL}$-sentences, but

$\forall x. Poss(x) \supset Know(Poss(x))$,
$OKnow(f_1(g(n))) \supset Know(\neg Know(f_2(g(n))))$, and
$OKnow(f_1(n)) \supset [t]Know(\neg Know(f_2(n)))$

are not. Note, in particular, that any fluent formula is also an objective $\mathcal{OL}$-formula. It turns out that the two logics are indeed one and the same when restricted to $\mathcal{OL}$-sentences.

Theorem 6 *For every $\mathcal{OL}$-sentence α, α is valid in $\mathcal{OL}$ iff α is valid in $\mathcal{ES}$.*

The proof is not difficult but tedious. Here we only go over the main ideas. A world in $\mathcal{OL}$ is simply a mapping from ground atoms with only standard names as arguments into $\{0, 1\}$. Similar to $\mathcal{ES}$, a model in $\mathcal{OL}$ consists of a pair $\langle e, w\rangle$, where w is an $\mathcal{OL}$-world and e a set of $\mathcal{OL}$-worlds. The theorem can be proved by showing that, for any $\mathcal{OL}$-model $\langle e, w\rangle$ there is an $\mathcal{ES}$-model $\langle e', w'\rangle$ so that both agree on the truth value of α, and vice versa. There are two complications that need to be addressed. One is that the domain of discourse of $\mathcal{OL}$ ranges over the standard names G^0, a proper subset of the domain of discourse R of $\mathcal{ES}$. This can be handled by using an appropriate bijection from G^0 into R when mapping models of one kind into the other. The other complication arises when mapping an $\mathcal{ES}$-model $\langle e, w\rangle$ into an appropriate $\mathcal{OL}$-model. For that we need the property that for all $\mathcal{ES}$-worlds w and w', if w and w' agree initially, that is, $w[\langle\,\rangle\!:\!f(\vec{t})] = w'[\langle\,\rangle\!:\!f(\vec{t})]$ for all ground atoms $f(\vec{t})$, then either both are in e or both are not in e. It can be shown that, with respect to $\mathcal{OL}$-sentences, we can restrict ourselves to $\mathcal{ES}$-models with this property without loss of generality.

[8] Actually, in (Levesque and Lakemeyer 2001) non-rigid function symbols are also considered, an issue we ignore here for simplicity.

We remark that while a previous embedding of $\mathcal{OL}$ into $\mathcal{AOL}$ (Lakemeyer and Levesque 1998) required an actual translation of formulas, none of that is needed here. Having $\mathcal{OL}$ fully embedded in $\mathcal{ES}$ has the advantage that existing results for $\mathcal{OL}$-formulas immediately carry over to the static part of $\mathcal{ES}$.

To see where this pays off, consider the right-hand side of the regression theorem for knowledge (Theorem 5). It is not hard to see that, provided that the arguments of fluents in Σ, Σ', and α are standard names, the right-hand side of Theorem 5 is an $\mathcal{OL}$-formula. It turns out that we can then leverage results from $\mathcal{OL}$ and show that in order to determine whether such implications hold, no modal reasoning at all is necessary!

The idea is perhaps best explained by an example. Suppose $OKnow(\phi)$ is true, where ϕ is $(P(a) \vee P(b)) \wedge P(c)$, where a, b, and c are standard names. In order to determine whether $\exists x P(x) \wedge \neg Know(P(x))$ is also known, it suffices to first determine the known instances of P. For our given ϕ, the only known instance of P is c, which we can express as $x = c$. (Note, in particular, that neither a nor b is known to satisfy P.) Then we replace $Know(P(x))$ by $x = c$ and check whether the resulting objective sentence $\exists x P(x) \wedge \neg(x = c)$ is entailed by ϕ, which it is. In general, determining the known instances of a formula with respect to ϕ always reduces to solving a series of first-order entailment questions. Hence no modal reasoning is necessary. This is the essence of the *Representation Theorem* of (Levesque and Lakemeyer 2001).

To make this precise, we will now define $\|\alpha\|_\phi$, which is the objective formula resulting from replacing in α all occurrences of subformulas $Know(\psi)$ by equality expressions as above, given that ϕ is all that is this known.

Formally, we first define $\mathrm{RES}[\psi, \phi]$, which is an equality expression representing the known instances of ψ with respect to ϕ. Here both ϕ and ψ are objective. $\|\cdot\|_\phi$ then applies RES to all occurrences of *Know* within a formula using a recursive descent.

Definition 2 *Let ϕ be an objective $\mathcal{OL}$-sentence and ψ an objective $\mathcal{OL}$-formula. Let $n_1, \ldots, n_k$ be all the standard names occurring in ϕ and ψ and let n' be a name not occurring in ϕ or ψ. Then $\mathrm{RES}[\psi, \phi]$ is defined as:*

1. *If ψ has no free variables, then $\mathrm{RES}[\psi, \phi]$ is TRUE, if $\phi \models \psi$, and FALSE, otherwise.*
2. *If x is a free variable in ψ, then $\mathrm{RES}[\psi, \phi]$ is*
$$((x = n_1) \wedge \mathrm{RES}[\psi^x_{n_1}, \phi]) \ \vee \ldots$$
$$((x = n_k) \wedge \mathrm{RES}[\psi^x_{n_k}, \phi]) \vee$$
$$((x \neq n_1) \wedge \ldots \wedge (x \neq n_k) \wedge \mathrm{RES}[\psi^x_{n'}, \phi]^{n'}_x).$$

Following (Levesque and Lakemeyer 2001), let us define a formula to be *basic* if it does not mention *OKnow*.[9]

[9] This should not be confused with *basic* action theories.

Definition 3 *Given an objective $\mathcal{OL}$-sentence ϕ and a basic $\mathcal{OL}$-formula α, $\|\alpha\|_\phi$ is the objective formula defined by*

$\|\alpha\|_\phi = \alpha$, *when α is objective;*
$\|\neg\alpha\|_\phi = \neg\|\alpha\|_\phi$;
$\|(\alpha \wedge \beta)\|_\phi = (\|\alpha\|_\phi \wedge \|\beta\|_\phi)$;
$\|\forall x\alpha\|_\phi = \forall x\|\alpha\|_\phi$;
$\|\mathit{Know}(\alpha)\|_\phi = \mathrm{RES}[\|\alpha\|_\phi, \phi]$.

Theorem 7 *Let ϕ and ψ be objective $\mathcal{OL}$-sentences, and let α be a basic $\mathcal{OL}$-sentence. Then*

$$\models \psi \wedge \mathit{OKnow}(\phi) \supset \alpha \quad \mathit{iff} \quad \models \psi \supset \|\alpha\|_\phi.$$

Proof: The statement holds in $\mathcal{OL}$ and the proof is a slight variant of the proof of Theorem 7.4.1 (the Representation Theorem) together with Theorem 8.4.1 of (Levesque and Lakemeyer 2001). By Theorem 6, the statement then holds in $\mathcal{ES}$ as well. ∎

Note that no modal reasoning is required to figure out $\|\alpha\|_\phi$. So standard theorem-proving techniques can be employed. There is a price to pay, however: in contrast to classical theorem proving, RES is not recursively enumerable since it appeals to provability, when returning TRUE, and *non-provability,* when returning FALSE.

We can now combine the previous theorem with the regression theorem for knowledge (Theorem 5) to reduce reasoning about bounded formulas to reasoning about static formulas that are now also objective. Formally, we have the following:

Theorem 8 *Given a pair of basic action theories Σ and Σ', and a bounded, basic sentence α,*

$$\Sigma \wedge \mathit{OKnow}(\Sigma') \models \alpha \quad \mathit{iff} \quad \models \Sigma_0 \supset \|\mathcal{R}[\alpha]\|_{\Sigma'_0}.$$

Proof: By Theorem 5, we have that $\Sigma \wedge \mathit{OKnow}(\Sigma') \models \alpha$ iff $\Sigma_0 \wedge \mathit{OKnow}(\Sigma'_0) \models \mathcal{R}[\alpha]$, which can be rewritten as $\models \Sigma_0 \wedge \mathit{OKnow}(\Sigma'_0) \supset \mathcal{R}[\alpha]$. By definition, Σ_0 and Σ'_0 are both fluent sentences and hence objective $\mathcal{OL}$-sentences. Since $\mathcal{R}[\alpha]$ is a basic $\mathcal{OL}$-sentence by Lemma 10 below, the result follows by Theorem 7. ∎

To show that $\mathcal{R}[\alpha]$ is a basic $\mathcal{OL}$-sentence, we proceed in two steps.

Lemma 9 *If α is a fluent sentence, then $\mathcal{R}[\Sigma', \Sigma, \sigma, \alpha]$ is an objective $\mathcal{OL}$-sentence.*

Proof: Since α is a fluent sentence, only Rules 1–4 and 7 of the definition of $\mathcal{R}$ apply. To simplify notation we write $\mathcal{R}[\sigma, \alpha]$ instead of $\mathcal{R}[\Sigma', \Sigma, \sigma, \alpha]$ with the understanding that regression is with respect to Σ. The proof is by induction on σ. Let $\sigma = \langle\rangle$. We proceed by a sub-induction on α. $\mathcal{R}[\langle\rangle, f(t_1, \ldots, t_k)] = f(t_1, \ldots, t_k)$, which is obviously an objective $\mathcal{OL}$-sentence, and the same for $\mathcal{R}[\langle\rangle, t_1 = t_2]$. The cases for $\neg$, $\wedge$ and $\forall$ follow easily by induction.

Suppose the lemma holds for σ of length n. Again, we proceed by sub-induction on α. $\mathcal{R}[\sigma \cdot t, f(t_1, \ldots, t_k)] = \mathcal{R}[\sigma, (\gamma_f)^{a\,x_1}_{t\,t_1} \cdots ^{x_k}_{t_k}]$, where $\Box[x]f(\vec{y}) \equiv \gamma_f$ is in Σ_{post}. Since γ_f is a fluent formula, $\mathcal{R}[\sigma, (\gamma_f)^{a\,x_1}_{t\,t_1} \cdots ^{x_k}_{t_k}]$ is an objective $\mathcal{OL}$-formula by the outer induction hypothesis. The case for $=$ is clear, and the cases for $\neg$, $\wedge$ and $\forall$ again follow easily by induction. ∎

Lemma 10 *Let α be a bounded, basic sentence. Then $\mathcal{R}[\Sigma', \Sigma, \sigma, \alpha]$ is a basic $\mathcal{OL}$-sentence.*

Proof: The proof is by induction on α. If α is a fluent sentence, then the lemma follows immediately from Lemma 9. $\mathcal{R}[\Sigma', \Sigma, \sigma, Poss(t)] = \mathcal{R}[\Sigma', \Sigma, \sigma, \pi^a_t]$. Since π is a fluent formula, the lemma again follows by Lemma 9. The same holds for $\mathit{SF}(t)$. $\mathcal{R}[\Sigma', \Sigma, \sigma, [t]\alpha] = \mathcal{R}[\Sigma', \Sigma, \sigma \cdot t, \alpha]$, which is a basic $\mathcal{OL}$-sentence by induction. The case for *Know* is proved by a sub-induction on σ. $\mathcal{R}[\Sigma', \Sigma, \langle\rangle, \mathit{Know}(\alpha)] = \mathit{Know}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$. By the outer induction hypothesis, $\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha]$ is a basic $\mathcal{OL}$-sentence and, hence, $\mathit{Know}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$ is one as well. $\mathcal{R}[\Sigma', \Sigma, \sigma \cdot t, \mathit{Know}(\alpha)] = \mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{SF}(t) \wedge \mathit{Know}(\mathit{SF}(t) \supset [t]\alpha) \vee \neg\mathit{SF}(t) \wedge \mathit{Know}(\neg\mathit{SF}(t) \supset [t]\alpha)] = \mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{SF}(t)] \wedge \mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{Know}(\mathit{SF}(t) \supset [t]\alpha)] \vee \mathcal{R}[\Sigma', \Sigma, \sigma, \neg\mathit{SF}(t)] \wedge \mathcal{R}[\Sigma', \Sigma, \sigma, \mathit{Know}(\neg\mathit{SF}(t) \supset [t]\alpha)]$. In each case, either Lemma 9 applies or the induction hypothesis for σ, and we are done. ∎

In (Lakemeyer and Levesque 1999), the Representation Theorem was applied to answering queries in $\mathcal{AOL}$. However, there reasoning was restricted to the initial situation or one that was progressed (Lin and Reiter 1997) after a number of actions had occurred. In other words, the representation theorem was applicable only after actions had actually been performed. Here, in contrast, we are able to answer questions about possible future states of the world, which is essential for planning and, for that matter, Golog.

Related Work

The closest approaches to ours are perhaps those combining dynamic logic with epistemic logic such as (Herzig et al. 2000) and (Demolombe 2003). In the language of (Herzig et al. 2000), it is possible to express things like $[\mathit{dipLitmus}][\mathit{seeRed}]\mathit{Know}(\mathit{Acid})$ using an almost identical syntax and where *Know* also has a possible-world semantics. Despite such similarities, there are significant differences, however. In particular, their language is propositional. Consequently, there is no quantification over actions, which is an essential feature of our form of regression (as well as Reiter's).

Demolombe (2003) proposes a translation of parts of the epistemic situation calculus into modal logic. He even considers a form of only-knowing. While his modal language is first-order, he considers neither a substitutional interpretation of quantifiers nor quantified modalities, which we find essential to capture successor state axioms. Likewise, there is no notion of regression.

Although they do not consider epistemic notions, the work by (Blackburn et al. 2001) is relevant as it reconstructs a version of the situation calculus in Hybrid Logic (Blackburn et al. 2001), a variant of modal logic which was inspired by the work on tense logic by Prior (Prior 1967). In a sense, though, this work goes only part of the way as an explicit reference to situations within the logic is retained. To us this presents a disadvantage when moving to an epistemic extension. As we said in the beginning, the problem is that the epistemic situation calculus requires us to consider uncountably many situations, which precludes a substitutional interpretation of quantification.

Conclusions

In this paper we proposed a language for reasoning about knowledge and action that has many of the desirable features of the situation calculus as presented in (Reiter 2001a) and of $\mathcal{OL}$ as presented in (Levesque and Lakemeyer 2001). From the situation calculus, we obtain a simple solution to the frame problem, and a regression property, which forms the basis for *Golog* (Levesque et al. 1997), among other things. From $\mathcal{OL}$, we obtain a simple quantified epistemic framework that allows for very concise semantic proofs, including proofs of the determinacy of knowledge and other properties of the situation calculus. To obtain these advantages, it was necessary to consider a language with a substitutional interpretation of the first-order quantifiers, and therefore (in an epistemic setting), a language that did not have situations as terms. Despite not having these terms, and therefore not having an accessibility relation K over situations as a fluent, we were able to formulate a successor state axiom for knowledge, and show a regression property for knowledge similar to that of (Scherl and Levesque 2003). This allowed reasoning about knowledge and action to reduce to reasoning about knowledge in the initial state. Going further, we were then able to use results from $\mathcal{OL}$ to reduce all reasoning about knowledge and action to ordinary first-order non-modal reasoning.

While $\mathcal{ES}$ seems sufficient to capture the basic action theories of the situation calculus, not all of the situation calculus is representable. For example, consider the sentence $\exists s \forall s'. s' \sqsubseteq s \supset P(s')$, which can be read as "there is a situation such that every situation preceding it satisfies P. There does not seem to be any way to say this in $\mathcal{ES}$. In a companion paper (Lakemeyer and Levesque 200x), we show that, by considering a second-order version of $\mathcal{ES}$, we regain the missing expressiveness under some reasonable assumptions. In addition, we show how to reconstruct all of Golog in the extended logic and, moreover, under the same assumptions, that the non-epistemic situation calculus and non-epistemic second-order $\mathcal{ES}$ are of equal expressiveness.

Acknowledgments

We thank the anonymous reviewers for their helpful comments.

References

P. Blackburn, J. Kamps, and M. Marx, Situation calculus as hybrid logic: First steps. In P. Brazdil and A. Jorge (Eds.) *Progress in Artificial Intelligence*, Lecture Notes in Artificial Intelligence 2258, Springer Verlag, 253-260, 2001.

W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.

M. A. Castilho, O. Gasquet, and A. Herzig, Formalizing Action and Change in Modal Logic I: the frame problem. *Journal of Logic and Computation*, 9(5), 701–735, 1999.

R. Demolombe, Belief change: from Situation Calculus to Modal Logic. *IJCAI Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'03),* Acapulco, Mexico, 2003.

R. Fagin, J. Halpern, Y. Moses and M. Vardi, *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.

A. Finzi, F. Pirri, and R. Reiter, Open world planning in the situation calculus. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00),* AAAI Press, 754–760, 2000.

D. Harel, Dynamic Logic, in D. Gabbay and F. Guenther (Eds.), *Handbook of Philosophical Logic*, Vol. 2, D. Reidel Publishing Company, 497–604, 1984.

A. Herzig, J. Lang, D. Longin, and T. Polacsek, A logic for planning under partial observability. In *Proc. AAAI-2000*, AAAI Press.

J. Hintikka,, *Knowledge and Belief*. Cornell University Press, Ithaca, 1962.

D. Kaplan, Quantifying-in. In L. Linsky (ed.), *Reference and Modality*. Oxford University Press, Oxford, 112–144, 1971.

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

S. A. Kripke, Is there a problem with substitutional quantification? G. Evans and J. McDowell (eds.), *Truth and Meaning*, Clarendon Press, Oxford, 325–419, 1976.

G. Lakemeyer and H. J. Levesque, $\mathcal{AOL}$: a logic of acting, sensing, knowing, and only knowing. *Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning,* Morgan Kaufmann, San Francisco, 316–327, 1998.

G. Lakemeyer and H. J. Levesque, Query evaluation and progression in $\mathcal{AOL}$ knowledge bases. *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99),* Morgan Kaufmann, San Francisco, 124–131, 1999.

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

H. J. Levesque and G. Lakemeyer, *The Logic of Knowledge Bases*, MIT Press, 2001.

G. Lakemeyer and H. J. Levesque, Reconstructing the situation calculus and Golog in a language without situations as objects, in preparation.

H. J. Levesque, F. Pirri, and R. Reiter, Foundations for the Situation Calculus. Linköping Electronic Articles in Computer and Information Science, 3(18), 1998.

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.

F. Lin and R. Reiter, How to Progress a Database. *Artificial Intelligence*, **92**, 1997, pp.131-167.

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

J. McCarthy and P. J. Hayes, Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer, D. Mitchie and M. Swann (Eds.) *Machine Intelligence 4,* Edinburgh University Press, 463–502, 1969.

S. McIlraith and T. C. Son, Adapting Golog for Composition of Semantic Web Services, *Proc. of the Eighth International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Francisco, 482–493, 2002.

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

V. R. Pratt, Semantical considerations on Floyd-Hoare logic. In *Proc. 17th Ann. IEEE Symp. on Foundations of Computer Science,* IEEE Computer Society Press, 109–121, 1976.

A. Prior, *Past, Present and Future.* Oxford University Press, 1967.

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

R. Reiter, *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press, 2001.

R. Reiter, On knowledge-based programming with sensing in the situation calculus. *ACM Transactions on Computational Logic,* 433-457, 2001.

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