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 Know(A) entails (Know(B) ∨ Know(C)) in this theory, is it also true that Know(A) entails Know(B) or Know(A) entails 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 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 OL is the use of standard names, which allows a substitutional interpretation of the first-order quantifiers. While there have been philosophical arguments against substitu-
tional 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 OC only deals with static knowledge bases, an amalgamation of OC 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 OC and the situation calculus called ES. The idea is to keep the simplicity of OC, 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

\[ \text{[pickup(obj5)] [drop(obj5)] \text{Broken}(obj5)} \]

to say that \text{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; Demolombe 2003), we also allow formulas of the form \( \forall a.(\lnot \text{Broken}(obj5) \equiv \phi) \), 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.\footnote{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).} 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 OC (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.

The rest of the paper is organized as follows. In the next section we introduce the syntax and semantics of 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 OC. We end the paper with a discussion of related work and concluding remarks.

The Language

Formulas over symbols from the following vocabulary:

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

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.\footnote{Equivalently, the version in this paper can be thought of as having action terms but no object terms.}

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 \( \alpha \) is a formula, then \( [t] \alpha \) is a formula;
4. If \( \alpha \) and \( \beta \) are formulas, then so are \( (\alpha \land \beta) \), \( \lnot \alpha \), \( \forall x. \alpha \), \( \Box \alpha \), \( \text{Know}(\alpha) \), \( \text{OKnow}(\alpha) \).

We read \( [t] \alpha \) as “\( \alpha \) holds after action \( t \)”, \( \Box \alpha \) as “\( \alpha \) holds after any sequence of actions,” \( \text{Know}(\alpha) \) as “\( \alpha \) is known”, and \( \text{OKnow}(\alpha) \) as “\( \alpha \) is all that is known.” As usual, we treat \( \exists x. \alpha \), \( \exists x. \alpha \), \( (\alpha \lor \beta) \), \( (\alpha \supset \beta) \), and \( (\alpha \equiv \beta) \) as abbreviations. We call a formula without free variables a sentence. We sometimes use a finite set of sentences \( \Sigma \) 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:

\begin{itemize}
  \item a formula with no \( \Box \) operators is called bounded;
\end{itemize}
• a formula with no $\square$ 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 $\sigma \cdot p$ where $\sigma \in R^*$ is considered a sequence of actions, and $p = f^l(r_1, \ldots, r_k)$ is a ground fluent atom. In general, formulas are interpreted relative to a model $M = (e, w)$ 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 $\sigma$) 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 $w'[\sigma; SF(r)] = w[\sigma; SF(r)]$.

Here is the complete semantic definition of $\mathcal{ES}$: Given a model $M = (e, w)$ and sequence of actions $\sigma$, 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. $e, w, \sigma \models (\alpha \land \beta)$ iff $e, w, \sigma \models \alpha$ and $e, w, \sigma \models \beta$;
4. $e, w, \sigma \models \neg \alpha$ iff $e, w, \sigma \models \alpha$;
5. $e, w, \sigma \models \forall x. \alpha$ iff $e, w, \sigma \models \alpha^x$, 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 $e, w, \sigma \cdot \gamma' \models \alpha$, for every $\gamma' \in R^*$;
8. $e, w, \sigma \models \text{Know}(\alpha)$ iff for all $w' \simeq_\sigma w$,

   if $w' \models e$ then $e, w', \sigma \models \alpha$;
9. $e, w, \sigma \models \text{OKnow}(\alpha)$ iff for all $w' \simeq_\sigma w$,

   $w' \models e$ iff $e, w', \sigma \models \alpha$.

When $\alpha$ is a sentence, we sometimes write $e, w \models \alpha$ instead of $e, w, \langle \rangle \models \alpha$. In addition, when $\alpha$ is objective, we write $w \models \alpha$ and when $\alpha$ is subjective, we write $e \models \alpha$. When $\Sigma$ is a set of sentences and $\alpha$ is a sentence, we write $\Sigma \models \alpha$ (read: $\Sigma$ logically entails $\alpha$) to mean that for every $e$ and $w$, if $e, w \models \alpha$ for every $\gamma' \in \Sigma$, then $e, w \models \alpha$. Finally, we write $\models \alpha$ (read: $\alpha$ 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 $F$, a set $\Sigma \subseteq \mathcal{ES}$ of sentences is called a basic action theory over $F$ iff $\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}}$ where $\Sigma$ mentions only fluents in $F$ and

1. $\Sigma_0$ is any set of fluent sentences;
2. $\Sigma_{\text{pre}}$ is a singleton sentence of the form $\Box \text{poss}(a) \equiv \pi$, where $\pi$ is a fluent formula;
3. $\Sigma_{\text{post}}$ is a set of sentences of the form $\Box [\alpha] f(\vec{x}) \equiv \gamma_f$, one for each fluent $f \in F$, and where $\gamma_f$ is a fluent formula.

The idea here is that $\Sigma_0$ expresses what is true initially (in the initial situation), $\Sigma_{\text{pre}}$ is one large precondition axiom, and $\Sigma_{\text{post}}$ 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, $\text{fragile}(x)$ (object $x$ is fragile), $\text{Holding}(x)$ (object $x$ is being held by some unnamed robot), and $\text{Broken}(x)$ (object $x$ is broken), and three actions, $\text{drop}(x)$, $\text{pickup}(x)$, $\text{repair}(x)$. The initial theory $\Sigma_0$ consists of the following two sentences:

$\neg \text{Broken}(\text{obj5})$,
$\forall x. \neg \text{Holding}(x)$.

This says that initially obj5 is not broken and the robot is not holding anything. The precondition axiom $\Sigma_{\text{pre}}$ is the following:

$\Box \text{poss}(a) \equiv$
$\exists x. a = \text{pickup}(x) \land \forall z. \neg \text{Holding}(z) \lor$
$\exists x. a = \text{drop}(x) \land \text{Holding}(x) \lor$
$\exists x. a = \text{repair}(x) \land \text{Holding}(x) \land \text{Broken}(x)$.

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 $\Sigma_{\text{post}}$ in the example has the following three elements:

$\Box [\alpha] \text{Holding}(x) \equiv$
$\forall a. a = \text{pickup}(x) \lor \text{Holding}(x) \land a \neq \text{drop}(x)$,
$\Box [\alpha] \text{fragile}(x) \equiv \text{fragile}(x)$,
$\Box [\alpha] \text{Broken}(x) \equiv$
$\forall a. a = \text{drop}(x) \land \text{fragile}(x) \lor$
$\text{Broken}(x) \land a \neq \text{repair}(x)$.

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 $\Box \text{poss}(a) \equiv \pi$ stands for $\forall a. \Box \text{poss}(a) \equiv \pi$.

$^4$The $[t]$ construct has higher precedence than the logical connectives. So $\Box [\alpha] f(\vec{x}) \equiv \gamma_f$ abbreviates $\forall a. \Box \text{(}[\alpha] (\vec{x}) \equiv \gamma_f)$.
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 \( \Sigma \), a fundamental reasoning task is projection, that is, determining what holds after a number of actions have occurred. For example, is \texttt{obj5} broken after first picking it up and then dropping it? Formally, this corresponds to determining if

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

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 \texttt{obj5} is fragile). In general, the projection task involves determining if

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

where \( \Sigma \) is a basic action theory, the \( r_i \) are ground terms (representing actions), and \( \alpha \) is an arbitrary sentence. Reiter showed how successor state axioms allow the use of regression to solve this reasoning task for certain \( \alpha \) (which he called the regreessable formulas). The idea is to successively replace fluents in \( \alpha \) 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 \( \alpha \) amounts to determining whether \( \alpha \) holds at the end of every branch of the conditional.

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

1. \( R[\sigma, (t_1 = t_2)] = (t_1 = t_2) \);
2. \( R[\sigma, \forall x \alpha] = \forall x R[\sigma, \alpha] \);
3. \( R[\sigma, (\alpha \land \beta)] = (R[\sigma, \alpha] \land R[\sigma, \beta]) \);
4. \( R[\sigma, \neg \alpha] = \neg R[\sigma, \alpha] \);
5. \( R[\sigma, \pi[t]] = R[\sigma \cdot t, \alpha] \);
6. \( R[\sigma, \text{post}(t)] = R[\sigma, \pi[t]] \);
7. \( R[\sigma, f(t_1, \ldots, t_k)] \) is defined inductively on \( \sigma \) by:
   (a) \( R[\{\}, f(t_1, \ldots, t_k)] = f(t_1, \ldots, t_k) \);
   (b) \( R[\sigma \cdot f(t_1, \ldots, t_k)] = R[\sigma, (f(\gamma_1 t_1, \ldots, \gamma_k t_k))]. \)

Note that this definition uses the right-hand sides of both the precondition and successor state axioms from \( \Sigma \).

Using the semantics of \( ES \), 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 \( \Sigma \) another world \( w_{\Sigma} \) which is like \( w \) except that it is defined to satisfy the \( \Sigma_{\text{pre}} \) and \( \Sigma_{\text{post}} \) sentences of \( \Sigma \).

**Definition 1** Let \( w \) be a world and \( \Sigma \) a basic action theory with fluent predicates \( F \). Then \( w_{\Sigma} \) is a world satisfying the following conditions:

1. for \( f \notin F \), \( w_{\Sigma}[\sigma : f(r_1, \ldots, r_k)] = w[\sigma : f(r_1, \ldots, r_k)] \);
2. for \( f \in F \), \( w_{\Sigma}[\sigma : f(r_1, \ldots, r_k)] \) is defined inductively:
   (a) \( w_{\Sigma}[\{\} : f(r_1, \ldots, r_k)] = w[\{\} : f(r_1, \ldots, r_k)] \);
   (b) \( w_{\Sigma}[\sigma \cdot r : f(r_1, \ldots, r_k)] = 1 \) if \( w[\sigma, \sigma = (f(\gamma_1 t_1, \ldots, \gamma_k t_k))]. \)
3. \( w[\sigma : \text{Poss}(r)] = 1 \) if \( w[\sigma, \sigma = \pi[t]. \)

Note that this again uses the \( \pi \) and \( \gamma_f \) formulas from \( \Sigma \). Then we get the following simple lemmas:

**Lemma 1** For any \( w \), \( w_{\Sigma} \) exists and is uniquely defined.

**Proof:** \( w_{\Sigma} \) clearly exists. The uniqueness follows from the fact that \( \pi \) is a fluent formula and that for all \( f \in F \), once the initial values of \( \alpha \) are fixed, then the values after any number of actions are uniquely determined by \( \Sigma_{\text{post}} \).

**Lemma 2** If \( w \models \Sigma_0 \) then \( w_{\Sigma} \models \Sigma_0 \).

**Proof:** Directly from the definition of \( w_{\Sigma} \), we have that \( w_{\Sigma} \models \forall a \text{Poss}(a) \equiv \pi \) and \( w_{\Sigma} \models \forall a \forall x \exists a \exists f(x, \gamma_f) \equiv \gamma_f \).

**Lemma 3** If \( w \models \Sigma \) then \( w = w_{\Sigma} \).

**Proof:** If \( w \models \text{Poss}(a) \equiv \pi \) and \( w \models \square[a] f(x) \equiv \gamma_f \), then \( w \) satisfies the definition of \( w_{\Sigma} \).

**Lemma 4** Let \( \alpha \) be any bounded, objective sentence. Then \( w \models R[\sigma, \alpha] \) if \( w_{\Sigma} \models \sigma = \alpha \).

**Proof:** The proof is by induction on the length of \( \alpha \) (treating the length of \( \text{Poss}(t) \) as the length of \( \pi[t] \) plus 1). The only tricky case is for \( \text{Poss}(r) \) and for fluent atoms. We have that \( w_{\Sigma} \models \sigma = \text{Poss}(r) \) iff (by definition of \( w_{\Sigma} \)) \( w_{\Sigma} \models \sigma = \pi[t] \) iff (by induction) \( w \models R[\sigma, \pi[t] \).

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

**Proof:** Suppose \( \Sigma_0 \models R[\alpha] \). We prove that \( \Sigma \models \alpha \). Let \( w \) be any world such that \( w \models \Sigma \). Then, \( w \models \Sigma_0 \), and so \( w \models R[\alpha] \). 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 \( \Sigma_0 \models R[\alpha] \). 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, \( w \models R[\alpha] \).
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 $\mathcal{DS}$ (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. $\models \Box (\text{Know}(\alpha) \land \text{Know}(\alpha \supset \beta) \supset \text{Know}(\beta))$;
2. $\models \Box (\text{Know}(\alpha))$;
3. $\models \Box (\neg \text{Know}(\alpha) \supset \text{Know}(\neg \text{Know}(\alpha)))$;
4. $\models \Box (\forall x. \text{Know}(\alpha) \supset \text{Know}(\forall x. \alpha))$;
5. $\models \Box (\exists x. \text{Know}(\alpha) \supset \text{Know}(\exists x. \alpha))$.

**Proof:**

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

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: $\Box (\text{Know}(\exists x. \alpha) \supset \text{Know}(\exists x. \text{Know}(\beta)))$ is not valid in general. Despite the fact that quantification is understood substitutionally, knowing that someone satisfies $\alpha$ 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 $\alpha$ is an objective sentence and $\beta$ is an objective formula with one free variable $x$, such that $\models \text{Know}(\alpha) \supset \exists x. \text{Know}(\beta)$. Then for some rigid term $r$, $\models \text{Know}(\alpha) \supset \text{Know}(\beta^x_r)$.

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

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

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

This proof is exactly as it would be in $\mathcal{OC}$. Again it is worth noting that the proof of this theorem in the ordinary situation (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 $\Sigma$ and $\Sigma'$, where $\Sigma$ is our account of how the world is and will change as the result of actions, and $\Sigma'$ 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 $\mathcal{R}[\Sigma']$. It is easy to see that

**Lemma 6** $\mathcal{R}[\Sigma], w \models \text{OKnow}(\Sigma)$.

**Proof:** Let $w'$ be any world. Then $w' \equiv_{\sigma} w$ by the definition of $\equiv_{\sigma}$. By the definition of $\mathcal{R}[\Sigma]$ we have that $w' \in \mathcal{R}[\Sigma]$ iff $w' \models \alpha$. Hence $\mathcal{R}[\Sigma] \models \text{OKnow}(\Sigma)$.

As discussed in (Scherl and Levesque 2003), actions can be divided into ordinary actions which change the world like pickup(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

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

where $\Sigma_{\text{sense}}$ is a singleton sentence exactly parallel to the one for Poss of the form

$$\Box SF(a) \equiv \varphi$$

where $\varphi$ 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, $\Sigma_{\text{sense}}$ would be the following:

$$\Box SF(a) \equiv [a = \text{seeRed} \land \text{Red} \lor a \not= \text{seeRed}].$$
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.\footnote{Here we restrict ourselves to sensing truth values. See (Scherl and Levesque 2003) for how to handle arbitrary values.}

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

**Theorem 4** \[ \square [a] \text{Know}(\alpha) \equiv SF(a) \land \text{Know}(SF(a) \supset [a] \alpha) \lor \neg SF(a) \land \text{Know}(\neg SF(a) \supset [a] \alpha). \]

*Proof:* Let $e, w, \sigma \models [r] \text{Know}(\alpha^r)$ for $r \in R$. We write $\alpha'$ for $\alpha^r$. Suppose $e, w, \sigma \models SF(r)$. (The case where $e, w, \sigma \not\models SF(r)$ is analogous.) It suffices to show that $e, w, \sigma \models \text{Know}(SF(r) \supset [r] \alpha')$. So suppose $w' \simeq_r w$ and $w' \in e$. Thus $w'[\sigma : SF(r)] = w[\sigma : SF(r)] = 1$ by assumption, that is, $w' \simeq_r w$. Since $e, w, \sigma \models [r] \text{Know}(\alpha')$ by assumption, $e, w', \sigma' \models [r] \alpha'$ follows, from which $e, w', \sigma \models [r] \alpha'$ follows.

Conversely, let $e, w, \sigma \models SF(r) \land [r] \text{Know}(SF(r) \supset [r] \alpha')$. (The other case is similar.) We need to show that $e, w, \sigma \models \text{Know}(\alpha')$, that is, $e, w, \sigma \models \text{Know}(\alpha')$. Let $w' \simeq_{r-r} w$ and $w' \in e$. Then $w'[\sigma : SF(r)] = w[\sigma : SF(r)] = 1$ by assumption. Hence $e, w', \sigma' \models [r] \alpha'$, from which $e, w, \sigma \models [r] \text{Know}(\alpha')$ 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 $\Sigma$, the regression operator $R$ will be defined relative to a pair of basic action theories $(\Sigma', \Sigma)$ where, as above, $\Sigma'$ represents the beliefs of the agent. We allow $\Sigma$ and $\Sigma'$ 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.\footnote{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.}

The proof for $\text{Know}(\alpha)$ is by induction on $\sigma$ with a sub-induction on $\alpha$.

Let $\sigma = \langle \rangle$. As with the case of $\text{Poss}$ in Lemma 4, we take the length of $SF(r)$ to be the length of $\varphi^p_\sigma$ plus 1. The proof for $\text{Poss}$, fluent atoms, and the connectives $\land$, $\lor$ and $\forall$ is exactly analogous to Lemma 4.

**Lemma 7** \[ e, w \models \text{OKnow}(\Sigma_0) \Rightarrow e_\xi \models \text{OKnow}(\Sigma). \]

*Proof:* Let $e \models \text{OKnow}(\Sigma_0)$, that is, for all $w, w' \in e$ if $w' \models \Sigma_0$. We need to show that for all $w, w', w' \in e_\xi$ iff $w' \models \Sigma_0$.

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

Conversely, let $w \models \Sigma_0$. By definition, there is a $w' \in e_\xi$ such that $w = w'$. Since $w' \models \Sigma_0$, by Lemma 2, $w_\xi \models \Sigma$, 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 $\Sigma$, we first define $e_\xi = \{ w_\xi : w \in e \}$.

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

*Proof:* The proof is by induction on $\sigma$ with a sub-induction on $\alpha$.

Let $\sigma = \langle \rangle$. As with the case of $\text{Poss}$ in Lemma 4, we take the length of $\text{SF}(r)$ to be the length of $\varphi^p_\sigma$ plus 1. The proof for $\text{Poss}$, fluent atoms, and the connectives $\land$, $\lor$ and $\forall$ is exactly analogous to Lemma 4.

For $\text{SF}$, we have the following:

1. $e_{\Sigma'}, w_\xi, \langle \rangle \models \text{SF}(r)$ iff (by the definition of $w_\xi$), $e_{\Sigma'}, w_\xi, \alpha \models \varphi^p_\sigma$ (by induction),
2. $e, w \models R[\Sigma', \Sigma, \langle \rangle, \varphi^p_\sigma]$ (by the definition of $R$), $e, w \models R[\Sigma', \Sigma, \langle \rangle, \text{SF}(r)]$.

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

1. $e_{\Sigma'} \models \text{Know}(\alpha)$ iff for all $w \in e_{\Sigma'}$, $e_{\Sigma'}, w \models \alpha$ (by definition of $e_{\Sigma'}$),
2. for all $w \in e, e_{\Sigma'}, w_\xi \models \alpha$ (by induction),
3. for all $w \in e, e, w \models R[\Sigma', \Sigma, \langle \rangle, \alpha]$ (by definition of $R$),
4. $e \models \text{Know}(R[\Sigma', \Sigma, \langle \rangle, \alpha])$ (by definition of $R$), $e \models R[\Sigma', \Sigma, \langle \rangle, \text{Know}(\alpha)]$.

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 $\alpha$. The proof is exactly like the sub-induction for the base case except for $\text{Know}$, for which we have the following:
\[d\text{ipLitmus}\text{ is true (the litmus paper is red), there is a second action }\]
\[\text{Red}\text{ seeRed}\text{ sider the litmus-test example adapted from (Scherl and}
\text{To illustrate how regression works in practice, let us con-
\text{known in future states, as opposed to only-known, seems
\text{situation.

\[\text{seeRed}\text{ will be true after }\]
\[\text{dipLitmus}\text{ was indeed true, and so }\]
\[\text{Red}\text{ is the unique epistemic state satisfying
\text{makes Red true just in case the solution is acidic, represented
\text{from above, and }\Sigma_{\text{pre}} = \{\Box\text{Poss}(a) \equiv \text{true}\}, \text{which states
\text{actions are always possible, for simplicity. We let }\Sigma_{\text{post}}\text{-the successor state axioms, be the following:

\[\Box[a]\text{Acid} \equiv \text{Acid},\]
\[\Box[a]\text{Red} \equiv \]
\[a = \text{dipLitmus} \land \text{Acid} \lor \text{Red} \land a \neq \text{dipLitmus},\]

that is, the acidity of the solution is unaffected by any action,
\text{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.}
\text{Finally, we let }\Sigma_0,\text{ the initial theory, be the following:

\text{Acid, }\neg\text{Red}.
\text{Now let us consider two basic action theories:
\[\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}\]
\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 }\Sigma.\text{ This amounts to saying that in reality the solution
is acidic and the litmus paper is initially not red (in }\Sigma),\text{ but
that the agent has no knowledge about the initial state of the two
fluents (in }\Sigma').\text{ Then we get the following:

1. }\Sigma \land \text{OKnow}(\Sigma') \models \neg\text{Know(Acid)};
2. }\Sigma \land \text{OKnow}(\Sigma') \models [\text{dipLitmus}]\neg\text{Know(Acid)};
3. }\Sigma \land \text{OKnow}(\Sigma') \models [\text{dipLitmus}][\text{seeRed}]\text{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\text{Know(Acid)}\text{ would not be en-
tailed if we replaced }\text{OKnow}\text{ by }\text{Know}\text{ in the antecedent.

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

\[\Sigma_0 \land \text{OKnow(true)} \models \neg\text{Know(Acid)}\text{.}\]

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

To see why (2.) holds, first note that
\[\mathcal{R}[\text{dipLitmus}]\neg\text{Know(Acid)}] = \neg\mathcal{R}[\Sigma', \Sigma, \text{d}, \text{Know(Acid)}],\]
\text{where we abbreviate dipLitmus as }d\text{. Then, using Rule (9b),
\[\neg\mathcal{R}[\Sigma', \Sigma, \text{d}, \text{Know(Acid)}] = \]
\[\neg\mathcal{R}[\Sigma', \Sigma, \text{d}, \text{SF}(d) \land \text{Know(SF}(d) \supset [d]\text{Acid}) \lor \]
\[\neg\text{SF}(d) \land \text{Know}(\neg\text{SF}(d) \supset [d]\text{Acid})].\]
The right-hand side of the equality reduces to \(\neg\text{Know}(\text{Acid})\) because both \(R[\Sigma', \Sigma, \{d\}, \text{SF}(d)]\) and \(R[\Sigma', \Sigma', \{d\}, \text{SF}(d)]\) reduce to true and \(R[\Sigma, \Sigma', d, \text{Acid}] = \text{Acid}\). Hence (2.) also reduces to
\[\Sigma_0 \land \text{OKnow}(\text{true}) \models \neg\text{Know}(\text{Acid}),\]
which was shown to hold above.

Finally, (3.) holds because
\[R[\text{dipLitmus}] [\text{seeRed}, \text{Know}(\text{Acid})] \]
reduces to \(\text{Acid} \land \text{Know}(\text{Acid} \supset \text{Acid}) \lor \neg\text{Acid} \land \text{Know}(\text{Acid})\), which again follows from \(\Sigma_0 \land \text{OKnow}(\text{true})\).

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.

**OL is part of 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 OL of (Levesque 1990; Levesque and Lakemeyer 2001). We call such formulas and sentences OL-formulas and OL-sentences, respectively.

For example, if \(n\) is a standard name and \(f_1\) and \(f_2\) are fluent predicates,
\[\forall x. f_1(x) \supset \text{Know}(f_1(x)) \quad \text{and} \quad \text{OKnow}(f_1(n)) \supset \text{Know}(\neg\text{Know}(f_2(n)))\]
are OL-sentences, but
\[\forall x. \text{Poss}(x) \supset \text{Know}(\text{Poss}(x)), \quad \text{OKnow}(f_1(g(n))) \supset \text{Know}(\neg\text{Know}(f_2(g(n)))), \quad \text{and} \quad \text{OKnow}(f_1(n)) \supset [\text{Know}(\neg\text{Know}(f_2(n)))\]
are not. Note, in particular, that any fluent formula is also an objective OL-formula. It turns out that the two logics are indeed one and the same when restricted to OL-sentences.

**Theorem 6** For every OL-sentence \(\alpha\), \(\alpha\) is valid in OL iff \(\alpha\) is valid in ES.

The proof is not difficult but tedious. Here we only go over the main ideas. A world in OL is simply a mapping from ground atoms with only standard names as arguments into \(\{0, 1\}\). Similar to ES, a model in OL consists of a pair \((e, w)\), where \(w\) is an OL-world and \(e\) a set of OL-worlds. The theorem can be proved by showing that, for any OL-model \((e, w)\), there is an ES-model \((e', w')\) so that both agree on the truth value of \(\alpha\), and vice versa. There are two complications that need to be addressed. One is that the domain of discourse of OL ranges over the standard names \(G^0\), a proper subset of the domain of discourse \(R\) of 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 ES-model \((e, w)\) into an appropriate OL-model. For that we need the property that

---

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

9 This should not be confused with **basic** action theories.
Definition 3: Given an objective OL-sentence \( \phi \) and a basic OL-formula \( \alpha \), \( ||\alpha||_\phi \) is the objective formula defined by

\[
\begin{align*}
||\alpha||_\phi &= \alpha, \quad \text{when } \alpha \text{ is objective;} \\
\neg||\alpha||_\phi &= \neg\alpha; \\
(\alpha \land \beta)||_\phi &= ((\alpha)||_\phi \land (\beta)||_\phi); \\
\forall x\alpha||_\phi &= \forall x(\alpha)||_\phi; \\
\text{Know}(\alpha)||_\phi &= \text{RES}(\alpha)||_\phi.
\end{align*}
\]

Theorem 7: Let \( \phi \) and \( \psi \) be objective OL-sentences, and let \( \alpha \) be a basic OL-sentence. Then

\[\models \psi \land \text{Know}(\phi) \supset \alpha \iff \models \psi \supset ||\alpha||_\phi.\]

Proof: The statement holds in 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 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 \( \Sigma \) and \( \Sigma' \), and a bounded, basic sentence \( \alpha \),

\[
\Sigma \land \text{Know}(\Sigma') \models \alpha \iff \Sigma_0 \models \text{RES}[[\Sigma'], \alpha].
\]

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

To show that \( \text{RES}[[\Sigma'], \alpha] \) is a basic OL-sentence, we proceed in two steps.

Lemma 9: If \( \alpha \) is a fluent sentence, then \( \text{RES}[[\Sigma'], \Sigma, \sigma, \alpha] \) is an objective OL-sentence.

Proof: Since \( \alpha \) is a fluent sentence, only Rules 1–4 and 7 of the definition of \( \text{RES} \). To simplify notation we write \( \text{RES}[[\sigma, \alpha]] \) instead of \( \text{RES}[[\Sigma'], \Sigma, \sigma, \alpha] \) with the understanding that this regression is with respect to \( \Sigma \). The proof is by induction on \( \sigma \). Let \( \sigma = () \). We proceed by a sub-induction on \( \alpha \). \( \text{RES}[[(), f(t_1, \ldots, t_k)] = \text{RES}[[\sigma, t, f(t_1, \ldots, t_k)] = \text{RES}[[\sigma, (\gamma_f)] t_1, \ldots, t_k], \) which is obviously an objective OL-sentence, and the same for \( \text{RES}[[(), t_1 = t_2] \). The cases for \( \gamma, \land \) and \( \lor \) follow easily by induction.

Suppose the lemma holds for \( \sigma \) of length \( n \). Again, we proceed by sub-induction on \( \alpha \). \( \text{RES}[[\sigma, t, f(t_1, \ldots, t_k)] = \text{RES}[[\sigma, (\gamma_f)] t_1, \ldots, t_k], \) where \( \Box[x]f(y) \equiv \gamma_f \) is in \( \text{SF}_{\text{out}} \). Since \( \gamma_f \) is a fluent formula, \( \text{RES}[[\sigma, (\gamma_f)] t_1, \ldots, t_k] \) is an objective OL-formula by the outer induction hypothesis. The case for \( = \) is clear, and the cases for \( \gamma, \land \) and \( \lor \) again follow easily by induction.
countably 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 OCL 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 OCL, 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 OCL to reduce all reasoning about knowledge and action to ordinary first-order non-modal reasoning.

While $ES$ 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 $\exists s \forall s'. s' \subseteq s \sqcap P(s')$, 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 $ES$. In a companion paper (Lakemeyer and Levesque 200x), we show that, by considering a second-order version of $ES$, 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 $ES$ are of equal expressiveness.

Acknowledgments

We thank the anonymous reviewers for their helpful comments.

References


