Artificial Intelligence 175 (2011) 142–164

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

A semantic characterization of a useful fragment of the situation calculus with knowledge

Gerhard Lakemeyer [a,*], Hector J. Levesque [b]

[a] Dept. of Computer Science, RWTH Aachen, 52056 Aachen, Germany
[b] Dept. of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3A6

ARTICLE INFO

Article history:
Available online 3 April 2010

Keywords:
Knowledge representation
Reasoning about action

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 called $\mathcal{ES}$ is proposed that captures much of the expressive power of the original, but where certain technical results are much more easily proved. This is illustrated using two existing non-trivial results: the determinacy of knowledge theorem of Reiter and the regression theorem, which reduces reasoning about the future to reasoning about the initial situation. Furthermore, we show the correctness of our approach by embedding $\mathcal{ES}$ in Reiter's situation calculus.

1. Introduction

Among the many contributions of John McCarthy, the formalism of the *situation calculus* has proved to be an extremely useful tool for reasoning precisely about action and change. It was originally proposed in [29,30] as a dialect of first-order logic. A second-order refinement of the language, developed by Reiter and his colleagues [36], forms the theoretical and implementation foundation for *Golog* [27], a language for the high-level control of robots and other agents (see, for example, [2,31]). 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 [36] 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 [35] but also allow the use of regression-based reasoning, which has been used in planning [8] 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

* Corresponding author.
E-mail addresses: gerhard@cs.rwth-aachen.de (G. Lakemeyer), hector@cs.toronto.edu (H.J. Levesque).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2010.04.005

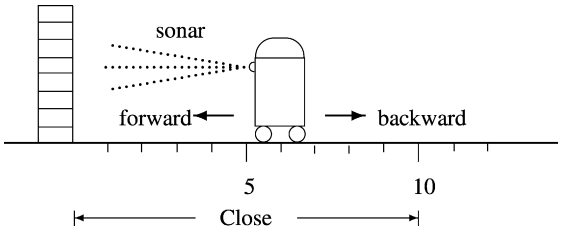

Fig. 1. A simple robot.

axiomatic setting unless there are ways to take derivations of *Formula1* from *Theory1* and convert them into derivations 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 [32,39]. If $Know(A)$ entails $(Know(B) \vee Know(C))$ 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 by Reiter requires a multi-page argument using considerable proof–theoretic machinery, including Craig's Interpolation Lemma [37].

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 Tarski structures can have different domains and considerable 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 epistemic logic $\mathcal{KL}$ [22], 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 [12,7]. 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 $\mathcal{KL}$ (and its extension, $\mathcal{OL}$) is the use of *standard names*, which allows a substitutional interpretation of the first-order quantifiers.[1] While there have been philosophical arguments against substitutional quantification [18], our experience has been that its technical simplicity has been of tremendous help in tackling issues such as quantifying-in [15], which are rarely addressed in other formalisms.

Since $\mathcal{KL}$ only deals with static knowledge bases, an amalgamation of $\mathcal{KL}$ and the situation calculus was previously proposed [19]. 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 $\mathcal{KL}$ and the situation calculus called $\mathcal{ES}$. The idea is to keep the simplicity of $\mathcal{KL}$, 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 [33,10], such as

$$[forward]\,[forward]\,distance = 4$$

to say that a robot is four units away from a wall after moving forward twice (see Fig. 1 for an illustration). In contrast to other modal approaches such as [3,11,5] but similar to [6], we also allow formulas of the form $\forall a, x.([a](distance = x) \equiv \phi)$, 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. 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 [26] 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) still leaves open the question of the overall correctness of the approach. In other words, is $\mathcal{ES}$ really a faithful reconstruction of the situation calculus? We will prove that it is by providing an embedding of $\mathcal{ES}$ in Reiter's version of the situation calculus, showing that the valid sentences of $\mathcal{ES}$ can be cast as entailments in Reiter's original version (modulo some modest assumptions). This shows that $\mathcal{ES}$ is a notational variant for a fragment of the situation calculus that can be given a clean and workable semantics. In addition, this result allows us to automatically transfer results obtained for $\mathcal{ES}$ to that fragment of the situation calculus, which is expressive enough to formulate basic action theories, and more.

The rest of the paper is organized as follows. In the next section we introduce the syntax and semantics of $\mathcal{ES}$ and discuss some of the properties of knowledge. In Section 3, we introduce the $\mathcal{ES}$-version of Reiter's basic action theories,

[1] Roughly speaking, this amounts to assuming at the outset that the domain of quantification is countably infinite and that there is a set of special constants called standard names uniquely denoting each element of the domain. A first-order universal sentence then ends up being true iff every instance of the sentence, where a standard name substitutes for the variable, is true.

followed by regression theorems for the non-epistemic and epistemic case. Section 5 provides the results on embedding $\mathcal{ES}$ in the situation calculus. We end the paper with a discussion of related work, only-knowing, and concluding remarks.

2. The logic $\mathcal{ES}$

The language is a second-order modal dialect with equality and sorts of type object and action. Before presenting the formal details, here are the main features:

- *Standard names*: Unlike other languages (but similar to $\mathcal{KL}$), the language includes (countably many) standard names for both objects and actions. These can be thought of as special extra constants that satisfy the unique name assumption and an infinitary version of domain closure. This allows first-order quantification to be understood substitutionally. Equality can also be given a simpler treatment: every ground term will have a coreferring standard name, and two terms are considered equal if their coreferring standard names are identical.
- *Fluent and rigid functions and predicates*: The language also contains both fluent and rigid predicate and function symbols. Fluents vary as the result of actions and have values that may be unknown, but rigids do not. These are present in the original situation calculus, of course. For example, we might have a sentence like this in the situation calculus:

  $$\mathit{Fragile}(c) \wedge \neg \mathit{Broken}(c, S_0).$$

  Here we can see that the first predicate is rigid and the second one is fluent just by seeing if the last argument is a situation. In our case, we do not have situation terms, and so we will need to distinguish the two sorts of predicates syntactically and semantically. Furthermore, for second-order quantification, we will need to distinguish rigid and fluent predicate variables as well.[2]
- *Knowledge and truth*: The language includes a modal operator *Know* for knowledge. This allows us to distinguish between sentences that are true and sentences that are known (by some implicit agent). For example, we can model situations where a robot is close to a wall but does not yet know it. We can also model situations where a robot has false beliefs about its world or how its world changes. The connection between knowledge and truth is made with *sensing*. Every action is assumed to have a binary sensing result and after performing the action, the agent learns that the action was possible (as indicated by the *Poss* predicate) and whether the sensing result for the action was 1 or 0 (as indicated by the *SF* predicate).

2.1. The language

Definition 1. The symbols of $\mathcal{ES}$ are taken from the following vocabulary:

- first-order variables: $x_1, x_2, \ldots, y_1, y_2, \ldots, a_1, a_2, \ldots$;
- fluent second-order variables of arity k: $P_1^k, P_2^k, \ldots$;
- rigid second-order variables of arity k: $Q_1^k, Q_2^k, \ldots$;
- standard names: $n_1, n_2, \ldots$ for objects and actions;
- fluent function symbols of arity k: $f_1^k, f_2^k, \ldots$; for example, *distance*;
- rigid function symbols of arity k: $g_1^k, g_2^k, \ldots$; for example, *forward*;
- fluent predicate symbols of arity k: $F_1^k, F_2^k, \ldots$; for example, *Broken*;
- rigid predicate symbols of arity k: $G_1^k, G_2^k, \ldots$; for example, *BrotherOf*;
- connectives and other symbols: $=$, $\wedge$, $\neg$, $\forall$, *Know*, $\Box$, round and square parentheses, period, comma.

We assume that all action function symbols are rigid and that the fluent predicates include the special predicates *Poss* and *SF*.

Definition 2. The *terms* of the language are of sort *action* or *object*, and form the least set of expressions such that

1. Every standard name and first-order variable is a term of the corresponding sort;
2. If $t_1, \ldots, t_k$ are terms and h is a k-ary function symbol then $h(t_1, \ldots, t_k)$ is a term of the same sort as h.

By a *primitive term* we mean one of the form $h(n_1, \ldots, n_k)$ where h is a (fluent or rigid) function symbol and all of the n_i are standard names.

[2] We follow Reiter in including both types of symbols. It is possible to live with just fluents, however, and treat rigids as fluents that happen not to change.

Definition 3. The *well-formed formulas* of the language form the least set such that

1. If $t_1, \ldots, t_k$ are terms, and H is a k-ary predicate symbol then $H(t_1, \ldots, t_k)$ is an (atomic) formula;
2. If $t_1, \ldots, t_k$ are terms, and V is a k-ary second-order variable, then $V(t_1, \ldots, t_k)$ is an (atomic) formula;
3. If t_1 and t_2 are terms, then $(t_1 = t_2)$ is a formula;
4. If t is an action term and α is a formula, then $[t]\alpha$ is a formula;
5. If α and β are formulas, v is a first-order variable, and V is a second-order variable, then the following are also formulas: $(\alpha \wedge \beta)$, $\neg\alpha$, $\forall v.\alpha$, $\forall V.\alpha$, $\Box\alpha$, $\mathit{Know}(\alpha)$.

We read $[t]\alpha$ as "α holds after action t", and $\Box\alpha$ as "α holds after any sequence of actions." So, for example, here is a successor state axiom in this language (we follow the usual convention of having free variables universally quantified from the outside):

$$\Box\big([a]\,\mathit{Broken}(x) \equiv \big(a = \mathit{drop}(x) \wedge \mathit{Fragile}(x)\big) \vee \big(\mathit{Broken}(x) \wedge a \neq \mathit{repair}(x)\big)\big)$$

In English: after any sequence of actions, an object x will be broken after doing action a iff a is the dropping of x when x is fragile or x was already broken and a is not the action of repairing it.

As usual, we treat $(\alpha \vee \beta)$, $(\alpha \supset \beta)$, $(\alpha \equiv \beta)$, $\exists v.\alpha$, and $\exists V.\alpha$ as abbreviations. To ease notation, we leave the type of variables implicit. We reserve the symbol a to denote a variable of type action.

We use α^x_n to mean formula α with all free occurrences of variable x replaced by name n. We call a formula without free variables a *sentence*. By a *primitive sentence* we mean a formula of the form $H(n_1, \ldots, n_k)$ where H is a (fluent or rigid) predicate symbol and all of the n_i are standard names.

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

- a formula with no $\Box$ operators is called *bounded*;
- a formula with no $\Box$ or $[t]$ operators is called *static*;
- a formula with no *Know* operators is called *objective*;
- a formula with no fluent, $\Box$, or $[t]$ operators outside the scope of a *Know* is called *subjective*;
- a formula with no *Know*, $\Box$, $[t]$, *Poss*, or *SF* is called a *fluent* formula.[3]

2.2. The semantics

The main purpose of the semantics we are about to present is to be precise about how we handle fluents, which may vary as the result of actions and whose values may be unknown. Intuitively, to determine whether or not a sentence α is true after a sequence of actions z has been performed, we need to specify two things: a world w and an epistemic state e. We write $e, w, z \models \alpha$. A world determines truth values for the primitive sentences and coreferring standard names for the primitive terms after any sequence of actions. An epistemic state is defined by a set of worlds, as in possible-world semantics.

More precisely, let $\mathcal{N}$ denote the set of all standard names and $\mathcal{Z}$ the set of all finite sequences of standard action names, including $\langle\rangle$, the empty sequence. Then

- a world $w \in W$ is any function from the primitive sentences and $\mathcal{Z}$ to $\{0, 1\}$, and from the primitive terms and $\mathcal{Z}$ to $\mathcal{N}$ (preserving sorts), and satisfying the rigidity constraint: if g is a rigid function or predicate symbol, then for all z and z' in $\mathcal{Z}$, $w[g(n_1, \ldots, n_k), z] = w[g(n_1, \ldots, n_k), z']$;
- an epistemic state $e \subseteq W$ is any set of worlds.

We extend the idea of coreferring standard names to arbitrary ground terms as follows. Given a term t without variables, a world w, and an action sequence z, we define $|t|^z_w$ (read: the coreferring standard name for t given w and z) by:

1. If $t \in \mathcal{N}$, then $|t|^z_w = t$;
2. $|h(t_1, \ldots, t_k)|^z_w = w[h(n_1, \ldots, n_k), z]$, where $n_i = |t_i|^z_w$.

So to find a coreferring standard name for $h(t_1, \ldots, t_k)$, we find coreferring names for the t_i recursively, and then use the function w on the resulting primitive term.

[3] In the situation calculus, these correspond to formulas that are uniform in some situation term.

To interpret formulas with free variables, we proceed as follows. First-order variables are handled substitutionally using the standard names. To handle the quantification over second-order variables, we use second-order *variable maps* defined as follows:

The *second-order primitives* are formulas of the form $V(n_1, \ldots, n_k)$ where V is a (fluent or rigid) second-order variable and all of the n_i are standard names. A *variable map* u is a function from worlds, second-order primitives, and $\mathcal{Z}$ to $\{0, 1\}$, satisfying the rigidity constraint: if Q is a rigid second-order variable, then for all w and w' in W, and all z and z' in $\mathcal{Z}$, $u[w, Q(n_1, \ldots, n_k), z] = u[w', Q(n_1, \ldots, n_k), z']$.

Let u and u' be variable maps, and let V be a (fluent or rigid) second-order variable; we write $u' \sim_V u$ to mean that u and u' agree except perhaps on the second-order primitives involving V.

Finally, to interpret what is known after a sequence of actions has taken place, we define $w' \simeq_z w$ (read: w' agrees with w agree on the sensing throughout action sequence z) inductively by the following:

1. $w' \simeq_{\langle\rangle} w$ iff w' and w agree on the value of every primitive rigid term and sentence;
2. $w' \simeq_{z \cdot n} w$ iff $w' \simeq_z w$, $w'[Poss(n), z] = 1$, and

$$w'\big[SF(n), z\big] = w\big[SF(n), z\big].$$

Note that $\simeq_z$ is not quite an equivalence relation because of the use of *Poss* here. As will become clearer, this is because we are insisting that the agent comes to believe that *Poss* was true after performing an action, even in those "non-legal" situations where the action was not possible in reality.[4]

Putting all these together, here is the semantic definition of truth. Given a sentence α of $\mathcal{ES}$, an epistemic state $e \subseteq W$ and a world $w \in W$, we define $e, w \models \alpha$ (read: α is true at e and w) as $e, w, \langle\rangle, u \models \alpha$ for any second-order variable map u, where for any $z \in \mathcal{Z}$ we have:

1. $e, w, z, u \models H(t_1, \ldots, t_k)$ iff $w[H(n_1, \ldots, n_k), z] = 1$, where $n_i = |t_i|^z_w$;
2. $e, w, z, u \models V(t_1, \ldots, t_k)$ iff $u[w, V(n_1, \ldots, n_k), z] = 1$, where $n_i = |t_i|^z_w$;
3. $e, w, z, u \models (t_1 = t_2)$ iff n_1 and n_2 are identical, where $n_i = |t_i|^z_w$;
4. $e, w, z, u \models [t]\alpha$ iff $e, w, z \cdot n, u \models \alpha$, where $n = |t|^z_w$;
5. $e, w, z, u \models (\alpha \wedge \beta)$ iff $e, w, z, u \models \alpha$ and $e, w, z, u \models \beta$;
6. $e, w, z, u \models \neg\alpha$ iff $e, w, z, u \not\models \alpha$;
7. $e, w, z, u \models \forall x.\alpha$ iff $e, w, z, u \models \alpha^x_n$, for every standard name n of the right sort;
8. $e, w, z, u \models \forall V.\alpha$ iff $e, w, z, u' \models \alpha$, for every $u' \sim_V u$;
9. $e, w, z, u \models \Box\alpha$ iff $e, w, z \cdot z', u \models \alpha$, for every $z' \in \mathcal{Z}$;
10. $e, w, z, u \models \mathit{Know}(\alpha)$ iff $e, w', z, u \models \alpha$, for every $w' \in e$ such that $w' \simeq_z w$.

When α is *objective* (has no *Know* operators), we can leave out the e and write $w \models \alpha$. Similarly, when α is *subjective*, we can leave out the w and 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 $\{\} \models \alpha$.

2.3. Knowledge

At this point we will not go into a detailed discussion of the properties of $\mathcal{ES}$. Instead we will focus on knowledge as a first example of how the semantics of $\mathcal{ES}$ allows us to prove properties with relative ease. A more complete picture of $\mathcal{ES}$ will emerge later when we establish a formal connection with Reiter's situation calculus.

The interpretation of knowledge in $\mathcal{ES}$ is just a special case of possible-world semantics [17,12]. One minor feature worth noting is that we do not simply require truth in all elements of e, the given set of "possible worlds," as in $\mathcal{KL}$. In fact, e represents the *initial* state of knowledge, and as knowledge is acquired though action, some of those initial worlds will no longer be considered possible. This is reflected in the $\simeq_z$ relation. In a nutshell, we look for truth in all elements of e that agree with the real world w in terms of sensing. It will then follow that after doing a sequence of actions, the agent will know the correct values of the sensing results in the real world (and everything it can conclude from that).

Regarding the more traditional logical properties of knowledge, it is not surprising that we obtain the usual properties of *weak S5* or *K45* [7]. 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.

[4] An alternate account that would state that the agent learns the true value of *Poss* (analogous to *SF*) is a bit more cumbersome, but would allow $\simeq_z$ to be a full equivalence relation.

Theorem 1.

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

Proof.

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

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: $\Box(Know(\exists x.\alpha) \supset \exists x.\, Know(\alpha))$ 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 easily prove a generalized version of the determinacy of knowledge:

Theorem 2. *Suppose α is an objective sentence and β is an objective formula with one free variable x, such that $\models Know(\alpha) \supset \exists x.\, Know(\beta)$. Then for some standard name n, $\models Know(\alpha) \supset Know(\beta^x_n)$.*

Proof. Suppose not. Then for every n (of the right sort), $Know(\alpha)$ does not entail $Know(\beta^x_n)$, and so, by the lemma below, α does not entail β^x_n. So for every n, there is a world w_n such that $w_n \models (\alpha \wedge \neg\beta^x_n)$. Now let $e = \{w_n \mid n \text{ a standard name}\}$. Then we have that $e \models Know(\alpha)$ and for every standard name n, $e \models \neg Know(\beta^x_n)$, and so $e \models \forall x.\neg Know(\beta)$. This contradicts the fact that $Know(\alpha)$ entails $\exists x.\, Know(\beta)$. □

Lemma 1. *If α and β are objective, and $\models (\alpha \supset \beta)$, then $\models (Know(\alpha) \supset Know(\beta))$.*

Proof. Suppose that some $e \models Know(\alpha)$. Then for every $w \in e$, $w \models \alpha$. Then for every $w \in e$, $w \models \beta$. Thus $e \models Know(\beta)$. □

This proof is exactly as it would be in $\mathcal{KL}$. 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.

3. Basic action theories

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 our basic action theories do not require foundational axioms like Reiter's second-order induction axiom for situations [36]. In fact, for this and the next section we will have no use for second-order logic at all and only consider the first-order fragment of $\mathcal{ES}$.

Definition 4. Given a set of fluents $\mathcal{F}$, a set $\Sigma \subseteq \mathcal{ES}$ of sentences is called a *basic action theory* over $\mathcal{F}$ iff $\Sigma = \Sigma_{\text{una}} \cup \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$ where Σ mentions only fluents in $\mathcal{F}$ and

1. Σ_{una} is a set of unique names axioms for action functions;
2. Σ_0 is any set of fluent sentences;
3. Σ_{pre} is a singleton sentence of the form $\Box\, Poss(a) \equiv \pi$, where π is a fluent formula[5];

[5] We assume that $\Box$ has lower syntactic precedence than the logical connectives, so that $\Box\, Poss(a) \equiv \pi$ stands for $\forall a.\Box(Poss(a) \equiv \pi)$.

4. Σ_{post} is a set of sentences of the form $\Box[a]F(\vec{v}) \equiv \gamma_F$ or $\Box[a]f(\vec{v}) = y \equiv \gamma_f$, one for each relational fluent F and functional fluent f, respectively, and where γ_F and γ_f are fluent formulas[6];
5. Σ_{sense} is a sentence exactly parallel to the one for *Poss* of the form $\Box SF(a) \equiv \varphi$, where φ is a fluent formula.

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 [35]. Here we follow the convention of [39] that every action returns a sensing result. Σ_{sense} then captures the outcome of sensing actions. For actions like *forward*, which do not return any useful sensing information, *SF* can be defined to be vacuously true (see below for an example).[7]

Since an agent's beliefs may differ from what is true, we will, in general, need two basic action theories: Σ for what is true in the world, including its dynamics, and Σ' for what the agent believes to be true. Except for the unique names assumption for actions, the two are allowed to differ arbitrarily and even contradict each other to allow for false beliefs. A state of affairs can then be characterized by sentences of the form Σ to denote what is actually true, $\mathit{Know}(\Sigma')$ to denote what the agent believes to be true, and perhaps a set of sentences $\neg \mathit{Know}(\phi)$ to capture some of the agent's ignorance.[8] We will be interested in the what is entailed by such theories.

As an example, imagine a robot that lives in a 1-dimensional world, and that can move towards or away from a fixed wall. The robot also has a sonar sensor that tells it when it gets close to the wall, say, less than 10 units away. See Fig. 1. So we might imagine three actions, *forward* and *backward* which move the robot one unit towards and away from the wall, and a *sonar* sensing action which tells the robot if it is close to the wall but has no effect on the world. For simplicity, we will simply assume that these three are standard names, that is, we do not need to stipulate unique names axioms for these. We have a single fluent, *distance*, which gives the actual distance from the robot to the wall.[9]

Let us consider informally how sensing relates knowledge to truth here. We start in some initial epistemic state e and world w. Initially, before any actions have taken place, the action sequence z is $\langle\rangle$. We might suppose that $w[\mathit{distance}, \langle\rangle] = 6$ as in the diagram. If the robot does not know where it is, there may be a $w^* \in e$ where $w^*[\mathit{distance}, \langle\rangle] = 13$. Now suppose the robot performs a *sonar* action. In this case, we would expect that $w[SF(\mathit{sonar}), \langle\rangle] = 1$, but $w^*[SF(\mathit{sonar}), \langle\rangle] = 0$. In other words, if the sonar is doing its job, in w it would tell us that the robot is close to the wall and in w^* it would tell us that the robot is far from the wall. So if we now let $z = \langle \mathit{sonar} \rangle$, we see that $w^* \not\simeq_z w$, since they disagree on the *SF* value. In fact, for every w' such that $w' \simeq_z w$, we will have that $w'[SF(\mathit{sonar}), \langle\rangle] = 1$. Since the definition of *Know* uses $\simeq$, when we consider what is known after doing the *sonar* action, the robot will believe (correctly) that it is close to the wall: $e, w, \langle \mathit{sonar} \rangle \models \mathit{Know}(\mathit{distance} < 10)$.

Let us now make all this precise. We begin our formalization by writing preconditions for the three actions:

$$\begin{aligned}
&\Box\, \mathit{Poss}(a) \equiv \\
&\quad a = \mathit{forward} \wedge \mathit{distance} > 0 \quad \vee \\
&\quad a = \mathit{backward} \wedge \text{TRUE} \quad \vee \\
&\quad a = \mathit{sonar} \wedge \text{TRUE}.
\end{aligned}$$

In other words, while *backward* and *sonar* are always possible, *forward* is executable only when the robot is not already at the wall. Next, we define the sensing results for the actions:

$$\begin{aligned}
&\Box SF(a) \equiv \\
&\quad a = \mathit{forward} \wedge \text{TRUE} \quad \vee \\
&\quad a = \mathit{backward} \wedge \text{TRUE} \quad \vee \\
&\quad a = \mathit{sonar} \wedge \mathit{distance} < 10.
\end{aligned}$$

Since *backward* and *forward* are not expected to return any useful sensing information, *SF* is vacuously true for them, while $SF(\mathit{sonar})$ says that the sonar returns 1 precisely when the distance to the wall is less than 10. Finally, we write a successor state axiom for our only fluent:

$$\begin{aligned}
&\Box[a](\mathit{distance} = x) \equiv \\
&\quad a = \mathit{forward} \wedge \mathit{distance} = x + 1 \quad \vee
\end{aligned}$$

[6] The $[t]$ construct has higher precedence than the logical connectives. So $\Box[a]F(\vec{x}) \equiv \gamma_F$ abbreviates $\forall a.\Box([a]F(\vec{x}). \equiv \gamma_F)$.

[7] In this paper we restrict ourselves to sensing truth values. See [39] for how to handle arbitrary values.

[8] When we use Σ as part of a sentence we mean the conjunction of all the finitely many sentences contained in Σ.

[9] Here and below, we use simple arithmetic involving $<$, $+$, and $-$, which can easily be defined in second-order terms with the standard names acting as natural numbers. We omit the details.

$$a = \mathit{backward} \wedge \mathit{distance} = x - 1 \quad \vee$$

$$a \neq \mathit{forward} \wedge a \neq \mathit{backward} \wedge \mathit{distance} = x.$$

In other words, the distance to the wall increases or decreases by 1 depending on whether *backward* or *forward* is executed, or it remains as before for all other actions.

Now we are ready to consider some specifics having to do with what is true initially by defining an action theory. Let *Close* stand for the formula "*distance* < 10." Let ϕ denote the conjunction of the sentences above. We assume that ϕ is true and the robot knows it. We also assume the robot is located initially 6 units away from the wall, but that the robot has no idea where it is. So, we let $\Sigma = \{\phi\} \cup \{\mathit{distance} = 6\}$ and $\Sigma' = \{\phi\}$. Then we get this:

Example 1. The following are logical entailments of

$$\Sigma \wedge \mathit{Know}(\Sigma') \wedge \forall x.\neg \mathit{Know}(\mathit{distance} \neq x):$$

1. $\mathit{Close} \wedge \neg \mathit{Know}(\mathit{Close}) \wedge [\mathit{forward}]\neg \mathit{Know}(\mathit{Close})$
   the robot is close to the wall, but does not know it, and continues not to know it after moving forward;
2. $[\mathit{sonar}](\mathit{Know}(\mathit{Close}) \wedge [\mathit{forward}]\, \mathit{Know}(\mathit{Close}))$
   after reading the sonar, the robot knows it is close, and continues to know it after moving forward;
3. $[\mathit{sonar}][\mathit{backward}]\neg \mathit{Know}(\mathit{Close})$
   after reading the sonar and then moving backward, the robot no longer knows that it is close to the wall;
4. $[\mathit{backward}][\mathit{sonar}]\, \mathit{Know}(\mathit{Close})$
   after moving backward and then reading the sonar, the robot knows that it is close to the wall;
5. $[\mathit{sonar}][\mathit{forward}][\mathit{backward}]\, \mathit{Know}(\mathit{Close})$
   after reading the sonar, moving forward, and then backward, the robot knows that it is still close to the wall;
6. $[\mathit{sonar}]\, \mathit{Know}([\mathit{forward}]\mathit{Close})$
   after reading the sonar, the robot knows that it will remain close after moving forward;
7. $\neg \mathit{Know}([\mathit{sonar}]\, \mathit{Know}(\mathit{Close}))$
   the robot does not know initially that it will know that it is close after reading the sonar;
8. $\mathit{Know}([\mathit{sonar}](\mathit{Know}(\mathit{Close}) \vee \mathit{Know}(\neg \mathit{Close})))$
   the robot does know initially that after reading the sonar, it will then know whether or not it is close to the wall;
9. $\mathit{Know}([\mathit{sonar}][\mathit{backward}]\neg \mathit{Know}(\mathit{Close}))$
   the robot knows initially that it will not know that it is close after reading the sonar and moving backwards.

Proof. The proofs of these are similar. Here we will only do item 3. Let $z = \langle \mathit{sonar} \cdot \mathit{backward} \rangle$, and suppose that $e, w \models \Sigma \wedge \mathit{Know}(\Sigma') \wedge \forall x.\neg \mathit{Know}(\mathit{distance} \neq x)$; we must show that $e, w, z \models \neg \mathit{Know}(\mathit{Close})$. Because $e \models \forall x.\neg \mathit{Know}(\mathit{distance} \neq x)$, there exists $w' \in e$ such that $w' \simeq_{\langle\rangle} w$ and $w'[\mathit{distance}, \langle\rangle] = 9$. Since $9 < 10$, we also have that $w' \simeq_z w$. However, $w'[\mathit{distance}, z] = 10$. So there exists $w' \in e$ such that $w' \simeq_z w$ and $w', z \models \neg \mathit{Close}$. Therefore, $e, w, z \models \neg \mathit{Know}(\mathit{Close})$. □

4. Projection by regression

The examples of the previous section all involve *projection* as a fundamental reasoning task, that is, determining what holds after a number of actions have occurred, as in

$$\Sigma \wedge \mathit{Know}(\Sigma') \wedge \forall x.\neg \mathit{Know}(\mathit{distance} \neq x) \models [\mathit{sonar}][\mathit{backward}]\neg \mathit{Know}(\mathit{Close}).$$

When we are not concerned with knowledge, things are somewhat simpler as we only need a single basic action theory as in

$$\Sigma \models [\mathit{forward}][\mathit{backward}]\mathit{Close}.$$

For this simpler case, Reiter [36] showed how successor state axioms allow the use of *regression* to solve this reasoning task for certain α (which he called the *regressable formulas*) and which, roughly, correspond to bounded objective formulas in $\mathcal{ES}$. 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. Later Scherl and Levesque [39] extended these results to handle knowledge in the situation calculus. In this section we will show how these ideas carry over to $\mathcal{ES}$, beginning with the non-epistemic fragment of the language.

4.1. Regressing objective formulas

Here we consider regression to determine entailments of the form $\Sigma \models \alpha$, where Σ is a basic action theory and α is any bounded objective sentence. To start with we assume, from now on, that all basic action theories and queries are *rectified*, that is, that each quantifier has a distinct variable. This is needed for regression to work properly and later for the

translation of $\mathcal{ES}$ to the situation calculus.[10] To simplify the formal details, we will define regression only for formulas in the following normal form *NF*.

Definition 5. A sentence α is in *NF* if it is rectified and every function symbol f in α occurs only in equality expressions of the form $(f(n_1,\ldots,n_k)=n)$, where the n_i and n here are either variables or standard names.

It is easy to show that every sentence can be transformed into an equivalent one in *NF* and the transformation is linear in the size of the original sentence. For example, if b is a rigid constant and f a functional fluent, then the normal form of $F(f(b))$ is $\exists x,y.(b=x)\wedge(f(x)=y)\wedge F(y)$. Note that, for any formula in *NF*, if a term t appears in $[t]$ or as an argument to a function or predicate, then t is either a variable or a standard name. In the following we will make use of sequences which consist of action variables or action standard names. We will reserve the symbol r to denote such sequences. (We continue to use z to denote the special case where all elements of the sequence are standard names.)

In our account, any bounded, objective sentence α in *NF* is considered regressable. By the transformation above any bounded, objective sentence becomes regressable by first converting it into *NF* and then applying regression to the result.

Definition 6. We define $\mathcal{R}[\alpha]$, *the regression of* α *wrt* Σ, to be $\mathcal{R}[\langle\rangle,\alpha]$, where for any sequence r consisting of action variables or standard names, $\mathcal{R}[r,\alpha]$ is defined inductively on α by:

1. $\mathcal{R}[r,\forall x\alpha]=\forall x\mathcal{R}[r,\alpha]$;
2. $\mathcal{R}[r,(\alpha\wedge\beta)]=(\mathcal{R}[r,\alpha]\wedge\mathcal{R}[r,\beta])$;
3. $\mathcal{R}[r,\neg\alpha]=\neg\mathcal{R}[r,\alpha]$;
4. $\mathcal{R}[r,[t]\alpha]=\mathcal{R}[r\cdot t,\alpha]$;
5. $\mathcal{R}[r,Poss(t)]=\mathcal{R}[r,\pi^a_t]$;
6. $\mathcal{R}[r,SF(t)]=\mathcal{R}[r,\varphi^a_t]$;
7. $\mathcal{R}[r,G(t_1,\ldots,t_k)]=G(t_1,\ldots,t_k)$ for rigid predicate G;
8. $\mathcal{R}[r,F(t_1,\ldots,t_k)]$ for fluent predicate F is defined inductively on r by:
   (a) $\mathcal{R}[\langle\rangle,F(t_1,\ldots,t_k)]=F(t_1,\ldots,t_k)$;
   (b) $\mathcal{R}[r\cdot t,F(t_1,\ldots,t_k)]=\mathcal{R}[r,(\gamma_F)^{a\,v_1\ \cdots\ v_k}_{t\,t_1\ \cdots\ t_k}]$;
9. $\mathcal{R}[r,(t_1=t_2)]=(t_1=t_2)$ if t_1 and t_2 do not mention functional fluents;
10. $\mathcal{R}[r,(f(n_1,\ldots,n_k)=n)]$ for functional fluent f is defined inductively by:
    (a) $\mathcal{R}[\langle\rangle,(f(n_1,\ldots,n_k)=n)]=(f(n_1,\ldots,n_k)=n)$;
    (b) $\mathcal{R}[r\cdot t,(f(n_1,\ldots,n_k)=n)]=\exists y.(\gamma_f)^{a\,v_1\ \cdots\ v_k}_{t\,n_1\ \cdots\ n_k}\wedge(y=n)$.

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

It is not hard to show that $\mathcal{R}$ always transforms a bounded objective formula into a fluent formula.

Lemma 2. *Let α be a bounded objective formula and r a sequence of action variables or standard names. Then there is a unique fluent formula ϕ such that $\mathcal{R}[r,\alpha]=\phi$.*

Proof. The proof is simple but tedious and we will skip the details here. Perhaps the only interesting aspect is the structure of the proof itself, which is also used in other proofs of properties of regression below. First, the lemma is proved for static formulas only. This is achieved by an induction on the length of r and a sub-induction on the length of α, counting the number of logical operators and where occurrences of $Poss(t)$ and $SF(t)$ are counted as the length of π^a_t+1 and φ^a_t+1, respectively. Note, in particular, that the induction is well-behaved because the formulas π, φ, γ_F, and γ_f are themselves fluent formulas, that is, they are static and mention neither *Poss* nor *SF*.

Having proved the lemma for static α, the case of bounded formulas is established by another simple induction on the number of $[t]$-operators in α. □

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 satisfies the Σ_{pre}, Σ_{sense}, and Σ_{post} sentences of Σ.

Definition 7. Let w be a world, $z\in\mathcal{Z}$, and Σ a basic action theory with fluents $\mathcal{F}$. Then w_Σ is a world satisfying the following conditions:

[10] See also the proof of Lemma 6 below, where this is needed to establish the induction for $\forall$.

1. for $h \notin \mathcal{F}$ (predicate or function), $w_\Sigma[h(n_1,\ldots,n_k),z] = w[h(n_1,\ldots,n_k),z]$;
2. for predicate $F \in \mathcal{F}$, $w_\Sigma[F(n_1,\ldots,n_k),z]$ is defined inductively:
   (a) $w_\Sigma[F(n_1,\ldots,n_k),\langle\rangle] = w[F(n_1,\ldots,n_k),\langle\rangle]$;
   (b) $w_\Sigma[F(n_1,\ldots,n_k),z\cdot m] = 1$ iff $w_\Sigma, z \models (\gamma_F)^{a\ v_1}_{mn_1}\cdots{}^{v_k}_{n_k}$;
3. for function $f \in \mathcal{F}$, $w_\Sigma[f(n_1,\ldots,n_k),z]$ is defined inductively:
   (a) $w_\Sigma[f(n_1,\ldots,n_k),\langle\rangle] = w[f(n_1,\ldots,n_k),\langle\rangle]$;
   (b) $w_\Sigma[f(n_1,\ldots,n_k),z\cdot m] = n$ iff $w_\Sigma, z \models (\gamma_f)^{a\ y\ v_1}_{mn\ n_1}\cdots{}^{v_k}_{n_k}$;
4. $w_\Sigma[Poss(n),z] = 1$ iff $w_\Sigma, z \models \pi^a_n$;
5. $w_\Sigma[SF(n),z] = 1$ iff $w_\Sigma, z \models \varphi^a_n$.

Note that this again uses the π, γ, and φ formulas from Σ. Then we get the following simple lemmas[11]:

Lemma 3. *For any w, w_Σ exists and is uniquely defined.*

Proof. w_Σ clearly exists. The uniqueness follows from the fact that π and φ are fluent formulas and that for all fluents in $\mathcal{F}$, once their initial values are fixed, then the values after any number of actions are uniquely determined by Σ_{post}. □

Lemma 4. *If $w \models \Sigma_{\text{una}} \cup \Sigma_0$ then $w_\Sigma \models \Sigma$.*

Proof. Directly from the definition of w_Σ, we have that $w_\Sigma \models \forall a \Box Poss(a) \equiv \pi$, $w_\Sigma \models \forall a \Box SF(a) \equiv \varphi$, $w_\Sigma \models \forall a \forall \vec{x} \Box[a]F(\vec{x}) \equiv \gamma_F$, and $w_\Sigma \models \forall a \forall \vec{x} \forall y \Box[a]f(\vec{x}) = y \equiv \gamma_f$. □

Lemma 5. *If $w \models \Sigma$ then $w = w_\Sigma$.*

Proof. If $w \models \forall a.\Box Poss(a) \equiv \pi$, $w \models \forall a \Box SF(a) \equiv \varphi$, $w \models \forall a \forall \vec{x} \Box[a]F(\vec{x}) \equiv \gamma_f$, and $w \models \forall a \forall \vec{x} \forall y \Box[a]f(\vec{x}) = y \equiv \gamma_f$, then w satisfies the definition of w_Σ. □

The following property of regression is used to prove the main lemma needed for the Regression Theorem. Given a sequence of action variables or standard names r, let r^x_n denote r with all occurrences of variable x replaced by standard name n.

Lemma 6. *For any bounded objective formula α and sequence of action variables or standard names r, $\mathcal{R}[r,\alpha]^x_n = \mathcal{R}[r^x_n, \alpha^x_n]$.*

Proof. The proof is long but simple and follows the structure of the proof of Lemma 2. Here we only consider static α and three cases: fluent predicates, assuming that the lemma holds for $|r| = k-1$, *Poss* and $\forall$, assuming in the sub-induction that the lemma holds for formulas of length $m-1$.

1. Let $r = r'\cdot t$. Then $\mathcal{R}[r, F(\vec{t})]^x_n = \mathcal{R}[r', \gamma_F{}^{a\,\vec{u}}_{t\,\vec{t}}]^x_n$ (def. of $\mathcal{R}$) $= \mathcal{R}[r'^x_n, (\gamma_F{}^{a\,\vec{u}}_{t\,\vec{t}})^x_n]$ (by induction) $= \mathcal{R}[r'^x_n, (\gamma_F{}^{a\ \vec{u}}_{t^x_n\ \vec{t}^x_n})]$ (since x not in γ_F) $= \mathcal{R}[(r'\cdot t)^x_n, F(\vec{t})^x_n]$.
2. $\mathcal{R}[r, Poss(t)]^x_n = \mathcal{R}[r, \pi^a_t]^x_n$ (definition of $\mathcal{R}$) $= \mathcal{R}[r^x_n, (\pi^a_t)^x_n]$ (by induction, as π^a_t is of length $m-1$) $= \mathcal{R}[r^x_n, (\pi^a_{t^x_n})]$ (since π does not mention x) $= \mathcal{R}[r^x_n, Poss(t)^x_n]$.
3. $\mathcal{R}[r, \forall y.\alpha]^x_n = (\forall y.\mathcal{R}[r,\alpha])^x_n = \forall y.\mathcal{R}[r,\alpha]^x_n$ (since $x \neq y$) $= \forall y.\mathcal{R}[r^x_n, \alpha^x_n]$ (by induction on $|\alpha|$) $= \mathcal{R}[r^x_n, (\forall y.\alpha)^x_n]$. □

Lemma 7. *Let α be any bounded, objective sentence in NF and $z \in \mathcal{Z}$.*
Then $w \models \mathcal{R}[z,\alpha]$ iff $w_\Sigma, z \models \alpha$.

Proof. As before, the proof is rather straightforward and uses the same induction scheme as Lemma 2. Assuming the lemma holds for z of length $k-1$, we only consider two cases, atoms with functional fluents and $\forall$.

1. Note that, by the definition of *NF*, ground atoms mentioning functional fluents have the form $f(n_1,\ldots,n_k) = n$, where n and n_i are standard names. Then:
   $w_\Sigma, z\cdot m \models f(n_1,\ldots,n_k) = n$ iff (by definition of w_Σ),
   $w_\Sigma, z \models \exists y.(\gamma_f)^{a\ v_1}_{mn_1}\cdots{}^{v_k}_{n_k} \wedge y = n$ iff (by induction),
   $w \models \mathcal{R}[z, \exists y.(\gamma_f)^{a\ v_1}_{mn_1}\cdots{}^{v_k}_{n_k} \wedge y = n]$ iff (by definition of $\mathcal{R}$),
   $w \models \mathcal{R}[z\cdot m, f(n_1,\ldots,n_k) = n]$.
2. $w \models \mathcal{R}[z, \forall x.\alpha]$ iff $w \models \forall x.\mathcal{R}[z,\alpha]$ iff $w \models \mathcal{R}[z,\alpha]^x_n$ for all n of the right sort iff (by Lemma 6), $w \models \mathcal{R}[z,\alpha^x_n]$ for all n iff (by sub-induction on $|\alpha|$), $w_\Sigma, z \models \alpha^x_n$ for all n iff $w_\Sigma, z \models \forall x.\alpha$. □

[11] As we only consider first-order sentences here, the second-order variable map u is dropped everywhere.

Theorem 3 *(Objective regression). Let* $\Sigma = \Sigma_{\text{una}} \cup \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$ *be a basic action theory and let* α *be an objective, bounded sentence. Then* $\mathcal{R}[\alpha]$ *is a fluent sentence and satisfies*

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

Proof. Suppose $\Sigma_{\text{una}} \cup \Sigma_0 \models \mathcal{R}[\alpha]$. We prove that $\Sigma \models \alpha$. Let w be any world such that $w \models \Sigma$. Then, $w \models \Sigma_{\text{una}} \cup \Sigma_0$, and so $w \models \mathcal{R}[\alpha]$. By Lemma 7, $w_\Sigma \models \alpha$. By Lemma 5, $w_\Sigma = w$, 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 $w \models \Sigma_{\text{una}} \cup \Sigma_0$. From Lemma 4, $w_\Sigma \models \Sigma$, and so $w_\Sigma \models \alpha$. By Lemma 7, $w \models \mathcal{R}[\alpha]$. □

Note that the conciseness of this proof depends crucially on the fact that Lemma 7 is proved by induction over *sentences*, which is possible only because quantification is interpreted substitutionally.

4.2. Regressing knowledge

Let us now turn to the more general case of regression for bounded sentences which may refer to the agent's knowledge. As we discussed in Section 3, this means that we need to consider two basic action theories Σ and Σ' for what is true in the world and for what the agent believes, respectively.

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

Proof. For both directions of the equivalence we will only consider the case where $\neg\mathit{SF}(n)$ holds for an arbitrary action name n. The other case is completely analogous.

To prove the only-if direction, let $e, w, z \models [n]\mathit{Know}(\alpha_n^a)$ for action name n. We write α' for α_n^a. Suppose $e, w, z \models \neg\mathit{SF}(n)$. It suffices to show that $e, w, z \models \mathit{Know}(\mathit{Poss}(n) \wedge \neg\mathit{SF}(n) \supset [n]\alpha')$. So suppose $w' \simeq_z w$, $w' \in e$, $w'[\mathit{Poss}(n), z] = 1$, and $w'[\mathit{SF}(n), z] = 0$. Thus $w'[\mathit{SF}(n), z] = w[\mathit{SF}(n), z]$ and, hence, $w' \simeq_{z\cdot n} w$. Since $e, w, z \models [n]\mathit{Know}(\alpha')$ by assumption, $e, w', z \cdot n \models \alpha'$, from which $e, w', z \models [n]\alpha'$ follows.

Conversely, let $e, w, z \models \neg\mathit{SF}(n) \wedge \mathit{Know}(\mathit{Poss}(n) \wedge \neg\mathit{SF}(n) \supset [n]\alpha')$. We need to show that $e, w, z \models [n]\mathit{Know}(\alpha')$, that is, $e, w, z \cdot n \models \mathit{Know}(\alpha')$. Let $w' \simeq_{z\cdot n} w$ and $w' \in e$. Then $w'[\mathit{Poss}(n), z] = 1$ and $w'[\mathit{SF}(n), z] = w[\mathit{SF}(n), z] = 0$ by assumption. Hence $e, w', z \models \mathit{Poss}(n) \wedge \neg\mathit{SF}(n)$. Therefore, by assumption, $e, w', z \cdot n \models \alpha'$, from which $e, w, z \models [n]\mathit{Know}(\alpha')$ 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. Like [39], it makes the simplifying assumption that all actions are known to the agent. Unlike [39], it is formalized without a fluent for the knowledge accessibility relation, which would have required situation terms in the language. 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 action being possible and the sensing information provided by a.[12] For example, if after doing *sonar* the robot knows it is close to the wall, then before doing *sonar*, the robot already knew a conditional: if the *sonar* returns a 1 on completion, then this indicates that the robot will be close to the wall.

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.[13] 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.

Rules 1–10 of the new regression operator $\mathcal{R}$ are exactly as before (Definition 6) except for the extra arguments Σ' and Σ. Then we add the following:

11. $\mathcal{R}[\Sigma', \Sigma, r, \mathit{Know}(\alpha)]$ *is defined inductively on* r *by*:
    (a) $\mathcal{R}[\Sigma', \Sigma, \langle\rangle, \mathit{Know}(\alpha)] = \mathit{Know}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$;
    (b) $\mathcal{R}[\Sigma', \Sigma, r \cdot t, \mathit{Know}(\alpha)] = \mathcal{R}[\Sigma', \Sigma, r, \beta_t^a]$, *where* β *is the right-hand side of the equivalence in Theorem* 4.

[12] Note that by this account, after performing an *impossible* action, the agent believes that the action was possible. This is perhaps undesirable, but this anomaly arises only in non-legal situations.

[13] This is like [19] but in contrast to Scherl and Levesque [39], who can only handle true belief. While we allow for false beliefs, we continue to use the terms knowledge and belief interchangeably.

For simplicity, we write $\mathcal{R}[\alpha]$ instead of $\mathcal{R}[\Sigma', \Sigma, \langle\rangle, \alpha]$. Next we extend Lemma 4 to knowledge, where $e_\Sigma = \{w_\Sigma \mid w \in e\}$ for a given epistemic state e and basic action theory Σ:

Lemma 8. *If* $e \models \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma_0)$ *then* $e_\Sigma \models \mathit{Know}(\Sigma)$.

Proof. Let $e \models \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma_0)$, that is, for all w, if $w \in e$ then $w \models \Sigma_{\text{una}} \cup \Sigma_0$. We need to show that for all w, if $w \in e_\Sigma$ then $w \models \Sigma$.

Let $w \in e_\Sigma$. By definition, there is a $w' \in e$ such that $w = w'_\Sigma$. Since $w' \models \Sigma_{\text{una}} \cup \Sigma_0$, by Lemma 4, $w'_\Sigma \models \Sigma$, that is, $w \models \Sigma$. □

We now turn to the generalization of Lemma 7 for knowledge.

Lemma 9. $e, w \models \mathcal{R}[\Sigma', \Sigma, z, \alpha]$ *iff* $e_{\Sigma'}, w_\Sigma, z \models \alpha$.

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

Let $z = \langle\rangle$. The proof for *Poss*, *SF*, fluent atoms, and the connectives $\neg$, $\wedge$, and $\forall$ is exactly analogous to Lemma 7.

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 $z = \langle\rangle$.

Now consider the case of $z \cdot n$, 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:

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

which completes the proof. □

Finally, here is the general regression theorem:

Theorem 5 *(Generalized regression). Let* Σ *and* Σ' *be basic action theories, and* α *be a bounded sentence. Then* $\mathcal{R}[\alpha]$ *is a static sentence and satisfies*

$$\Sigma \wedge \mathit{Know}(\Sigma') \models \alpha \quad \textit{iff} \quad \Sigma_{\text{una}} \cup \Sigma_0 \wedge \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0) \models \mathcal{R}[\alpha].$$

Proof. To prove the only-if direction, let us suppose that $\Sigma \wedge \mathit{Know}(\Sigma') \models \alpha$ and that $e, w \models \Sigma_{\text{una}} \cup \Sigma_0 \wedge \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0)$. Thus $w \models \Sigma_{\text{una}} \cup \Sigma_0$ and, by Lemma 4, $w_\Sigma \models \Sigma$. Also, $e \models \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0)$ and thus, by Lemma 8, $e_{\Sigma'} \models \mathit{Know}(\Sigma')$. (Note that Σ_{una} is the same for both Σ and Σ'.) Therefore, $e_{\Sigma'}, w_\Sigma \models \Sigma \wedge \mathit{Know}(\Sigma')$. By assumption, $e_{\Sigma'}, w_\Sigma \models \alpha$ and, by Lemma 9, $e, w \models \mathcal{R}[\alpha]$.

Conversely, suppose $\Sigma_{\text{una}} \cup \Sigma_0 \wedge \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0) \models \mathcal{R}[\alpha]$ and let $e, w \models \Sigma \wedge \mathit{Know}(\Sigma')$. Then $w \models \Sigma_{\text{una}} \cup \Sigma_0$ and $e \models \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0)$. Then, by assumption, $e, w \models \mathcal{R}[\alpha]$. Then $e_{\Sigma'}, w_\Sigma \models \alpha$ by Lemma 9. By Lemma 5, $w_\Sigma = w$ and, since $e \models \mathit{Know}(\Sigma')$, $e_{\Sigma'} = e$. Therefore, $e, w \models \alpha$. □

This theorem shows that determining what is true and what is known after any (bounded) number of actions have occurred can always be reduced to reasoning about what is true and known in the initial state.

In order to deal with our robot example, we need to go a little beyond this as we may want to make assumptions about what the robot does not know as in

$$\Sigma \wedge \mathit{Know}(\Sigma') \wedge \forall x.\neg \mathit{Know}(\mathit{distance} \neq x) \models [\mathit{sonar}][\mathit{backward}]\neg \mathit{Know}(\mathit{Close}).$$

The following corollary shows that we can deal with such cases without any problems.

Corollary 1. *Let γ be a static sentence which mentions neither Poss nor SF. Then*

$$\Sigma \wedge \mathit{Know}(\Sigma') \wedge \gamma \models \alpha \quad \mathit{iff} \quad \Sigma_{\text{una}} \cup \Sigma_0 \wedge \mathit{Know}(\Sigma_{\text{una}} \cup \Sigma'_0) \wedge \gamma \models \mathcal{R}[\alpha].$$

Proof. The proof makes use of the fact that for any static formula γ, $\mathcal{R}[\gamma] = \gamma$. $\Sigma \wedge \mathit{Know}(\Sigma') \wedge \gamma \models \alpha$ iff $\Sigma \wedge \mathit{Know}(\Sigma') \models \gamma \supset \alpha$ iff $\Sigma \wedge \mathit{Know}(\Sigma') \models \mathcal{R}[\gamma \supset \alpha]$ (by the regression theorem) iff $\Sigma \wedge \mathit{Know}(\Sigma') \models \gamma \supset \mathcal{R}[\alpha]$ (because $\mathcal{R}$ leaves γ as is) iff $\Sigma \wedge \mathit{Know}(\Sigma') \models \gamma \supset \mathcal{R}[\alpha]$. □

5. Mapping to the situation calculus

How do we know that the semantics of $\mathcal{ES}$ is correct? In this section, we argue that it is indeed correct by showing how formulas α in $\mathcal{ES}$ can be translated in a direct way to formulas α^* in the situation calculus as defined by Reiter. We assume that this language has functional and relational fluents, functions and predicates that are not fluents, the distinguished constant S_0, function *do*, binary predicate $\sqsubseteq$ over situations, predicates *Poss* and *SF*, and a binary predicate K for knowledge. We take $\mathit{Knows}(\alpha, \sigma)$ in the situation calculus as an abbreviation for the formula $\forall s(K(s, \sigma) \supset \alpha_s^{now})$, where α_s^{now} is the result of replacing by s in α every occurrence of *now* that is not within the scope of a further *Knows*.[14]

Perhaps the most desirable and simplest outcome of a translation from $\mathcal{ES}$ to the situation calculus would be that

$$\models \alpha \quad \text{iff} \quad \Sigma \models_{\text{FOL}} \alpha^*,$$

where $\models$ is validity in $\mathcal{ES}$, Σ is the set of foundational axioms of the situation calculus, and $\models_{\text{FOL}}$ is ordinary classical logical consequence. Unfortunately, we do not get exactly this correspondence for a variety of reasons we will discuss below. But we do get something close:

$$\models \alpha \quad \text{iff} \quad \Sigma \cup \Upsilon \models_{\text{FOL}} \alpha^*,$$

where Υ is a set of five axioms that we will justify separately.

Somewhat surprisingly, it turns out that the foundational axioms Σ are actually completely irrelevant as far as the fragment of the situation calculus as defined by our translation is concerned. In other words, we also obtain

$$\models \alpha \quad \text{iff} \quad \Upsilon \models_{\text{FOL}} \alpha^*.$$

Below, we will first prove this result and then show that only small modifications are needed to obtain a proof for the case when foundational axioms are assumed as well. To establish these results it will be necessary to work with ordinary Tarski models of sentences of the situation calculus. As we argued in the beginning, this is difficult and painstaking, and is indeed one of the main reasons to prefer $\mathcal{ES}$ over the situation calculus. But here there is no alternative. So while the proof of the theorems is quite laborious, we remind the reader that this can be thought of as a final reckoning for a formalism that is unworkable semantically.

5.1. The translation

Before describing Υ, we present the translation from $\mathcal{ES}$ into the situation calculus. In the simplest case, the idea is that a formula like $\mathit{distance} = 6$, where *distance* is a fluent, will be mapped to the situation calculus formula $\mathit{distance}(S_0) = 6$, where we have restored the distinguished situation term S_0 for the fluent. Similarly, the formula $[\mathit{forward}]\neg(\mathit{distance} = 6)$ will be mapped to $\neg(\mathit{distance}(\mathit{do}(\mathit{forward}, S_0)) = 6)$, and $\Box(\mathit{distance} > 0)$ will be mapped to $\forall s'(S_0 \sqsubseteq s' \supset \mathit{distance}(s') > 0)$. For knowledge, $\mathit{Know}(\mathit{distance} > 0)$ will be mapped to $\mathit{Knows}(\mathit{distance}(\mathit{now}) > 0, S_0)$ which is an abbreviation for $\forall s'(K(s', S_0) \supset \mathit{distance}(s') > 0)$. So $\mathcal{ES}$ formulas can be thought of as "situation-suppressed" (in situation-calculus terminology) and the * mapping we will define restores the situation argument to the fluents, leaving the rigids unchanged.

More precisely, we have the following:

Definition 8. Let α be any term or formula of $\mathcal{ES}$ without standard names. The expression α^* is defined as $\alpha[S_0]$ where, for any situation term σ, $\alpha[\sigma]$ is defined inductively by:

1. $v[\sigma]$, where v is a first-order variable, is v;
2. $g(t_1, \ldots, t_k)[\sigma]$, where g is a rigid function, predicate, or second-order variable, is $g(t_1[\sigma], \ldots, t_k[\sigma])$;
3. $f(t_1, \ldots, t_k)[\sigma]$, where f is a fluent function, predicate, or second-order variable is $f(t_1[\sigma], \ldots, t_k[\sigma], \sigma)$;
4. $(t_1 = t_2)[\sigma]$ is $(t_1[\sigma] = t_2[\sigma])$;
5. $([t]\alpha)[\sigma]$ is $\alpha[\mathit{do}(t[\sigma], \sigma)]$;
6. $(\alpha \wedge \beta)[\sigma]$ is $(\alpha[\sigma] \wedge \beta[\sigma])$;

[14] In some versions of the situation calculus, the argument to *Knows* is a formula α where situations are suppressed. In that case, α_s^{now} should be understood as restoring s as the situation argument.

7. $(\neg\alpha)[\sigma]$ is $\neg\alpha[\sigma]$;
8. $(\forall v.\alpha)[\sigma]$ is $\forall v.\alpha[\sigma]$;
9. $(\forall V.\alpha)[\sigma]$ is $\forall V.\alpha[\sigma]$;
10. $(\Box\alpha)[\sigma]$ is $\forall s'(\sigma \sqsubseteq s' \supset \alpha[s'])$;
11. $\mathit{Know}(\alpha)[\sigma]$ is $\mathit{Knows}(\alpha[\mathit{now}], \sigma)$.

Note that the translation of $\Box\alpha$ introduces quantification over situations, where the introduced variable s' is assumed to be one that does not appear in situation term σ.

5.2. The axioms and the embedding theorem

The axioms we assume in Υ are the following:

1. Domain of objects is countably infinite[15];
2. Domain of actions is countably infinite (as above);
3. Equality is the identity relation[16]:

$$\forall x \forall y.(x = y) \equiv \forall Q\big(Q(x) \equiv Q(y)\big);$$

4. Less-than over situations: $\forall s \forall s'.(s \sqsubseteq s') \equiv \forall P(\cdots \supset P(s, s'))$, where the ellipsis stands for the universal closure of

$$\begin{aligned}&\big[P(s_1, s_1)\big] \quad \wedge\\ &\big[P(s_2, s_3) \supset P\big(s_2, do(a, s_3)\big)\big];\end{aligned}$$

5. The K predicate: $\forall s' \forall s.K(s', s) \equiv \forall P(\cdots \supset P(s', s))$, where the ellipsis stands for the universal closure of

$$\begin{aligned}&\big[K(s_1, S_0) \supset P(s_1, S_0)\big] \quad \wedge\\ &\big[P(s_1, s_3) \wedge P(s_2, s_3) \supset P(s_1, s_2)\big] \quad \wedge\\ &\big[P(s_1, s_2) \wedge \mathit{Poss}(a, s_1) \wedge \mathit{SF}(a, s_1) \equiv \mathit{SF}(a, s_2) \supset P\big(do(a, s_1), do(a, s_2)\big)\big].\end{aligned}$$

Axioms (1) and (2) talk about the cardinality of the set of objects and actions respectively: they are both countable and infinite. The countability aspect is not very controversial. In the first-order case, every satisfiable set of sentences is satisfiable in a countable domain, and we do not expect users of the situation calculus to use second-order logic to defeat this. Note that this does not rule out having theories that talk about real numbers or other continuous phenomena; it simply rules out using second-order logic to force the interpretations of these theories to be uncountable. We can, however, imagine contexts where finiteness might be desirable. In such cases, we can introduce a new predicate O and instead of asserting that there are finitely many objects, assert that there are finitely many objects in O.

As for axiom (3), it is hard imagining anyone taking the negation of this one seriously. The usual first-order axiomatization of equality is often enough, but the intent is invariably for the equality symbol to be understood as the identity relation, which this second-order axiom ensures.

Axiom (4) uses second-order logic to define the $\sqsubseteq$ relation as reachability using do. It does not say anything about S_0 nor about situations that cannot be reached using do. As it turns out, we do not need to stipulate anything about them. (See the conclusion for more on this.)

Finally axiom (5) is a second order definition of the K predicate in terms of the value it has at S_0. This is just another way of capturing the successor state axiom for K introduced by Scherl and Levesque [39], and the added machinery to make *Knows* be a weak-S5 operator [14]. Other knowledge operators are possible in the situation calculus, but weak-S5 and its extensions (such as strong-S5) are the most often used.

The last missing piece are the axioms asserting the countability of objects and actions. Here is one way of specifying these for objects:

$$\begin{aligned}&\exists Q.\forall x Q(x) \wedge \mathit{Inf}(Q) \wedge \mathit{Cnt}(Q) \quad \text{where}\\ &\quad \mathit{Cnt}(Q) \stackrel{def}{=} \forall Q'\big(Q' \subsetneq Q \wedge \mathit{Inf}\big(Q'\big) \supset Q \leqslant Q'\big)\\ &\quad \mathit{Inf}(Q) \stackrel{def}{=} \exists Q'.Q' \subsetneq Q \wedge Q \leqslant Q'\end{aligned}$$

[15] See the precise logical rendering of this below.

[16] It is not hard to show that this second-order definition entails all the usual properties like reflexivity and substitution of equals for equals.

$$Q \subsetneq Q' \stackrel{def}{=} \forall x\big(Q(x) \supset Q'(x)\big) \wedge \exists x\big(Q(x) \wedge \neg Q'(x)\big)$$

$$Q \leqslant Q' \stackrel{def}{=} \exists R.\forall x\big(Q(x) \supset \exists y\, Q'(y) \wedge R(x,y)\big) \wedge \forall x, x', y.R(x,y) \wedge R\big(x', y\big) \supset x = x'.$$

Starting from the bottom, $Q \leqslant Q'$ says that the set Q' is no smaller than Q, that is, there is a 1–1 mapping from Q to Q'; $Q \subsetneq Q'$ says that Q is a proper subset of Q'; $Inf(Q)$ says that Q is infinite if it contains a proper subset Q' which is no smaller than Q itself; $Cnt(Q)$ says that Q is countable if every infinite proper subset is no smaller than Q; finally, the first line says that the set of all objects is countably infinite (here x is assumed to be of type object). To assert the same for actions we simply add another axiom of the form

$$\exists Q.\forall a\, Q(a) \wedge Inf(Q) \wedge Cnt(Q) \quad \text{where } a \text{ is of type action.}$$

With all the axioms in place we can now state our first embedding theorem:

Theorem 6. *Let α be any sentence of $\mathcal{ES}$ without standard names. Then*

$$\alpha \text{ is valid} \quad \textit{iff} \quad \Upsilon \models_{\text{FOL}} \alpha^*.$$

The long and arduous proof of the theorem is left to Appendix A. If nothing else, it provides further evidence that ordinary Tarski models of sentences of the situation calculus are cumbersome to work with.

An interesting aspect of the theorem is that it requires few assumptions about the nature of situations. For example, Υ admits models where there are situations other than those reachable from S_0 or the situations which are K-accessible from S_0. In contrast, Reiter's foundational axioms Σ, which we will present in a moment, rule out such non-standard models. In order to show that $\mathcal{ES}$ can be fully embedded in Reiter's situation calculus, which requires the foundational axioms, we still need to show that the above theorem continues to hold if we add Σ as additional assumptions on the right-hand side of the theorem. Note that this is not immediately obvious as extra assumptions normally lead to more entailments.

To see why this is not the case here, let us first review the axioms Σ for the epistemic situation calculus from [36]:

1. $do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2$;
2. $\forall Q.\forall s.[Ini(s) \supset P(s)] \wedge \forall a, s.[Q(s) \supset Q(do(a,s))] \supset \forall s.Q(s)$, where $Ini(s) \stackrel{def}{=} \forall a \forall s'.s \neq do(a, s')$;
3. $s \sqsubset do(a, s') \equiv s \sqsubseteq s'$;
4. $\neg s \sqsubset S_0$;
5. $K(s', s) \supset [Ini(s) \equiv Ini(s')]$.

(1) is a unique names axiom for situations; the second-order axiom (2) defines the set of all situations to be those reachable from an initial situation ($Ini(s)$) by a sequence of actions; (3) defines $\sqsubset$ as reachability (by a sequence of actions) between situations; (4) says that no situation precedes S_0; (5) says that an initial situation is K-accessible only from another initial situation. Note that $Ini(S_0)$ is a logical consequence of Σ.

The intuitive reason why we can add these axioms to Theorem 6 without invalidating it is because they assert properties of situations which we cannot even express in $\mathcal{ES}$, and hence they are not in the image of the translation. For example, since S_0 is not part of the language, we simply cannot say that nothing precedes S_0. In other words, while we certainly have $\Sigma \cup \Upsilon \models_{\text{FOL}} \forall s.\neg s \sqsubset S_0$ and $\Upsilon \not\models_{\text{FOL}} \forall s.\neg s \sqsubset S_0$, this does not matter for the theorem, as there is no $\alpha \in \mathcal{ES}$ such that α^* is equivalent to $\forall s.\neg s \sqsubset S_0$. Hence we obtain:

Theorem 7. *Let α be any sentence of $\mathcal{ES}$ without standard names. Then*

$$\alpha \text{ is valid} \quad \textit{iff} \quad \Sigma \cup \Upsilon \models_{\text{FOL}} \alpha^*.$$

The theorem establishes that $\mathcal{ES}$ is indeed a fragment of Reiter's situation calculus. As $\mathcal{ES}$ itself seems to be much more workable, this then is perhaps the main significance of the theorem: any property which we obtain for $\mathcal{ES}$ automatically holds for the fragment of the original situation calculus given by our translation. And this fragment covers the main uses of the situation calculus, in particular, Reiter's basic action theories [36], and those dealing with knowledge [39]. We saw an example in Section 3. In [24] we also showed that $\mathcal{ES}$ is expressive enough to capture the action language Golog [27].

What do we give up? As we have seen, the foundational axioms themselves cannot be expressed in $\mathcal{ES}$. Another sentence which has no straightforward counterpart in is

$$\exists s \exists s'.S_0 \sqsubseteq s \wedge S_0 \sqsubseteq s' \wedge \big(s \neq s'\big) \wedge F(s) \equiv F\big(s'\big),$$

which says that two distinct situations are reachable that agree on the truth value of F. It's the equality between situations that presents a problem here. However, as we showed in [24], sentences like these *can* be expressed if we add to $\mathcal{ES}$ an explicit encoding of action sequences. Indeed, with this trick we were able to come up with a backward translation from the situation calculus to $\mathcal{ES}$, which covers the entire *rooted* situation calculus with knowledge. Here rooted means that

quantification over situations is restricted to formulas of the form $\forall s.\sigma \sqsubseteq s \supset \alpha$, where σ is a situation term. Note, in particular, that the rooted situation calculus without knowledge is equivalent to Reiter's original version (simply take σ to be S_0). So, with a little bit of extra effort, $\mathcal{ES}$ has almost the same expressive power as all of the situation calculus.

6. Related work

While the situation calculus has received a lot of attention in the reasoning about action community, there are, of course, a number of alternative formalisms, including close relatives like the fluent calculus [13,40] and more distant cousins described in [16,9,38].

The closest approaches to ours are perhaps those concerned with reasoning about action based on dynamic logic [10]. For example, De Giacomo and Lenzerini [4] and later Demolombe et al. [6], whose work extends [3], show how to import Reiter's solution to the frame problem into dynamic logic. There are also epistemic extensions of dynamic logic such as [11] and [5]. In the language of [11], it is possible to express things like $[\mathit{forward}][\mathit{sonar}]\,\mathit{Know}(\mathit{Close})$ using an almost identical syntax and where *Know* also has a possible-world semantics. While most approaches remain propositional, there are some first-order treatments such as [5,6], which, like $\mathcal{ES}$, are inspired by the desire to capture fragments of the situation calculus in modal logic.

Although they do not consider epistemic notions, the work by [1] is relevant as it reconstructs a version of the situation calculus in Hybrid Logic [1], a variant of modal logic which was inspired by the work on tense logic by Prior [34]. 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.

7. Only knowing and actions

In previous work, for example [21,22], we have used the concept of *only-knowing* as a way of capturing the idea that a sentence is not only believed by an agent, but *all* that is believed by the agent. This idea has a number of applications, including reconstructing some aspects of nonmonotonic reasoning [25].

One application of only-knowing for our purposes is to provide a convenient way of specifying what is not known: instead of saying that an agent believes some sentences (such as a basic action theory) and also stipulating that she does not believe certain other sentences (as we did in the robot example in Section 3), it will be sufficient to say that the basic action theory is all that the agent believes. It will then follow logically that certain other sentences are not believed. So for example, in the robot example, instead of saying

> The following are logical entailments of
>
> $$\Sigma \wedge \mathit{Know}(\Sigma') \wedge \forall x.\neg \mathit{Know}(\mathit{distance} \neq x),$$

where we had to say explicitly that the agent did not know anything about the distance to the wall, it will be sufficient to say

> The following are logical entailments of
>
> $$\big(\Sigma \wedge \mathit{OKnow}(\Sigma')\big),$$

where $\mathit{OKnow}(\Sigma')$ will be how we say in the language that Σ' is all that is known. Once this operator is properly defined, we will then obtain as a property that

$$\models \mathit{OKnow}(\Sigma') \supset \forall x.\neg \mathit{Know}(\mathit{distance} \neq x),$$

which then allows us to carry out the example. In fact, for any two fluent sentences ϕ and ψ such that $\not\models (\phi \supset \psi)$, we will have that $\models (\mathit{OKnow}(\phi) \supset \neg \mathit{Know}(\psi))$.

But how should this operator be generalized from our previous work on only-knowing in the static case? In an earlier version of this paper [23], we proposed a definition that had some interesting properties, but some drawbacks as well. We now believe that the definition should be the following:

- $e, w, z, u \models \mathit{OKnow}(\alpha)$ iff for every w', $w' \in e_z^w$ iff $e_z^w, w', \langle\rangle, u \models \alpha$,

where

e_z^w is defined as $\{w'_z \mid w' \in e \text{ and } w' \simeq_z w\}$, and
w_z is defined by $w_z[H(\vec{n}), z'] = w[H(\vec{n}), z \cdot z']$.

This semantic characterization appears to be the correct generalization of only-knowing for a dynamic language like $\mathcal{ES}$. In addition, it appears to have some very nice connections to the concept of the progression of basic action theories [20,28,41]. One complication here is that rigid predicates (predicates whose truth values are unchanging and known) appear to present problems for this otherwise well-behaved definition. However, we leave this exploration for future research, and do not pursue the matter here.

8. Conclusions

In this paper, we have isolated a fragment of the situation calculus with knowledge (presented using a modal syntax) and showed it to have a relatively simple model theoretic semantics based on possible worlds. We showed that this semantics allowed clear and direct proofs of certain properties of interest. We also showed that it coincided with the use of the situation calculus by Reiter provided we made five assumptions: there are countably infinitely many actions and objects, the $=$ symbol denotes identity, the $\sqsubseteq$ symbol denotes reachability using *do*, and knowledge satisfies the properties of a quantified weak S5. Given these assumptions, practitioners are then free to use either the modal syntax we have presented here or the original situation calculus syntax introduced by McCarthy and can expect to reap the same semantic rewards.

The subset of the situation calculus studied here might be thought to owe more to the dialect studied by Ray Reiter and coworkers [36] than to the less constrained dialect first envisioned by John McCarthy [29]. For example, we lean heavily on the use of basic action theories and regression, as popularized by Reiter. However, this is not quite right. A careful reading of the embedding theorem, which shows how a fragment of the classical situation calculus can be realized in $\mathcal{ES}$, demonstrates quite clearly that we get an exact correspondence without requiring all the foundational axioms proposed by Reiter. These axioms (including the second-order induction axiom) were used by Reiter to constrain the space of all situations to be a tree rooted at S_0. In our embedding, we do not rule out "non-standard models" where, for example, S_0 is the result of performing some action. We can get away with this heresy because, without situation terms in the language, such non-standard anomalies cannot be expressed! Thus, from the point of view of the situation calculus, we can leave the space of situations unconstrained, just as McCarthy did, without jeopardizing the advantages of the Reiter account.

So in the end, despite the modal syntax which is admittedly somewhat at odds with McCarthy's aesthetic, the situation calculus dialect presented here is in fact closer semantically to the vision first presented by McCarthy and subsequently found to be so useful by so many.

Appendix A. Proof of the Embedding Theorem

We need some notation for talking about classical Tarski structures and truth. Suppose we are given a Tarski structure $\mathcal{M}$ defined by a domain $\mathcal{D}$ and interpretations for the constant, function and predicate symbols of the situation calculus language. We assume that $\mathcal{D} = \mathcal{D}_{sit} \cup \mathcal{D}_{act} \cup \mathcal{D}_{obj}$, where $\mathcal{D}_{sit}$ is the domain of situations, $\mathcal{D}_{act}$ is the domain of actions, and $\mathcal{D}_{obj}$ is the domain of objects. We use the following notation: if c is a constant symbol, then $c^{\mathcal{M}}$ is the element of $\mathcal{D}$ denoted by c; $h^{\mathcal{M}}$ is the function from the Cartesian product over $\mathcal{D}$ of the appropriate arity to $\mathcal{D}$ denoted by h; and $H^{\mathcal{M}}$ is the relation (subset of the Cartesian product) over $\mathcal{D}$ of the appropriate arity denoted by H.

Let μ be a mapping from ordinary variables to $\mathcal{D}$ and from second-order variables to relations over $\mathcal{D}$. For any ordinary variable v and element d of $\mathcal{D}$, $\mu\{x/d\}$ is the variable map just like μ except that x is mapped to d. Similarly, for any second-order variable P and relation Z over $\mathcal{D}$ of the right arity, $\mu\{P/Z\}$ is the variable map just like μ except that P is mapped to Z.

More notation: For any term t, $\|t\|_{\mu}^{\mathcal{M}}$ is the element of $\mathcal{D}$ denoted by t in the classical sense, and $\mathcal{M}, \mu \models \alpha$ means that α is true in the classical sense for structure $\mathcal{M}$, when the free variables are interpreted by μ. We will omit the μ when nothing depends on the variable map.

Let $\mathcal{M}$ be a Tarski structure satisfying the axioms in Υ over the domain $\mathcal{D} = \mathcal{D}_{sit} \cup \mathcal{D}_{act} \cup \mathcal{D}_{obj}$. Let $\iota_0 = S_0^{\mathcal{M}}$ and $\mathcal{D}_{strt} = \{\iota_0\} \cup \{d_s \in \mathcal{D}_{sit} \mid \mathcal{M}, \mu\{s/d_s\} \models K(s, S_0)\}$. $\mathcal{D}_{strt}$ should be thought of as starting situations consisting of S_0 and whatever is K-accessible from S_0. These play the role of the initial situations in Reiter's situation calculus except that they may have predecessors because we do not require Reiter's foundational axioms. Let e be an epistemic state, and assume that we are given three mappings $\omega \in [\mathcal{D}_{strt} \to W]$, $\theta \in [\mathcal{N} \to \mathcal{D}_{act} \cup \mathcal{D}_{obj}]$, and $\pi \in [\mathcal{Z} \times \mathcal{D}_{strt} \to \mathcal{D}_{sit}]$. Suppose further that the following properties are satisfied:

1. θ is 1–1, onto, and sort preserving;
2. ω is onto e (that is, $e \subseteq$ the image of ω);
3. for any rigid function symbol g,
$$g^{\mathcal{M}}\big(\theta(n_1), \ldots, \theta(n_k)\big) = \theta\big(\omega(\iota)\big[g(n_1, \ldots, n_k), \langle\rangle\big]\big);$$
4. for any fluent function symbol f,
$$f^{\mathcal{M}}\big(\theta(n_1), \ldots, \theta(n_k), \pi(z, \iota)\big) = \theta\big(\omega(\iota)\big[f(n_1, \ldots, n_k), z\big]\big);$$
5. for any rigid predicate symbol G,
$$G^{\mathcal{M}} = \big\{\big\langle\theta(n_1), \ldots, \theta(n_k)\big\rangle \mid \omega(\iota)\big[G(n_1, \ldots, n_k), \langle\rangle\big] = 1\big\};$$

6. for any fluent predicate symbol F (including *Poss* and *SF*),
$$F^{\mathcal{M}} = \{\langle \theta(n_1), \ldots, \theta(n_k), \pi(z, \iota)\rangle \mid \omega(\iota)[F(n_1, \ldots, n_k), z] = 1\};$$
7. $do^{\mathcal{M}}(\theta(n), \pi(z, \iota)) = \pi(z \cdot n, \iota)$;
8. $=^{\mathcal{M}}$ is identity;
9. $\sqsubset^{\mathcal{M}} = \{(\pi(z, \iota), \pi(z \cdot z', \iota))\}$;
10. $K^{\mathcal{M}} = \{(\pi(z, \iota'), \pi(z, \iota)) \mid \omega(\iota') \in e \text{ and } \omega(\iota') \simeq_z \omega(\iota)\}$.

In what follows, we will consider expressions (terms or formulas) of $\mathcal{ES}$ whose free first-order variables appear in the list $x_1, \ldots, x_m$. We will consider substituting these variables by standard names $n_1, \ldots, n_m$ of the right sort. For ease of reading, for any term or formula α, we write α^+ to mean $\alpha^{x_1 \ldots x_m}_{n_1 \ldots n_m}$. In the situation calculus, for a variable map μ, we write μ^+ to mean $\mu\{x_1/\theta(n_1), \ldots, x_m/\theta(n_m), s/\pi(z, \iota)\}$, where the variable s, the $z \in \mathcal{Z}$ and the $\iota \in \mathcal{D}_{strt}$ will be determined by context. Then we get the following:

Lemma 10. *Let t be a term of $\mathcal{ES}$ without standard names whose free variables are among the $x_1, \ldots, x_m$. Let $\iota \in \mathcal{D}_{strt}$, $z \in \mathcal{Z}$, and $w = \omega(\iota)$. Then, given the properties above,*
$$\|t[s]\|_{\mu^+} = \theta(|t^+|^z_w).$$

Proof. By induction on t.
If t is the variable x_i, then $\|t[s]\|_{\mu^+} = \|x_i\|_{\mu^+} = \theta(n_i) = \theta(|t^+|^z_w)$.
If t is of the form $g(t_1, \ldots, t_k)$ then we have the following:
$$\begin{aligned}
&\|t[s]\|_{\mu^+} = (\text{by definition of } t[s])\\
&\|g(t_1[s], \ldots, t_k[s])\|_{\mu^+} = (\text{by definition of denotation})\\
&g^{\mathcal{M}}(\|t_1[s]\|_{\mu^+}, \ldots, \|t_k[s]\|_{\mu^+}) = (\text{by induction})\\
&g^{\mathcal{M}}(\theta(|t_1^+|^z_w), \ldots, \theta(|t_k^+|^z_w)) = (\text{by Property 3})\\
&\theta(w[g(|t_1^+|^z_w, \ldots, |t_k^+|^z_w), \langle\rangle]) = (\text{by the rigidity constraint})\\
&\theta(w[g(|t_1^+|^z_w, \ldots, |t_k^+|^z_w), z]) = (\text{by definition of coreference})\\
&\theta(|g(t_1, \ldots, t_k)^+|^z_w).
\end{aligned}$$
If t is of the form $f(t_1, \ldots, t_k)$ then we have the following:
$$\begin{aligned}
&\|t[s]\|_{\mu^+} = (\text{by definition of } t[s])\\
&\|f(t_1[s], \ldots, t_k[s], s)\|_{\mu^+} = (\text{by definition of denotation})\\
&f^{\mathcal{M}}(\|t_1[s]\|_{\mu^+}, \ldots, \|t_k[s]\|_{\mu^+}, \pi(z, \iota)) = (\text{by induction})\\
&f^{\mathcal{M}}(\theta(|t_1^+|^z_w), \ldots, \theta(|t_k^+|^z_w), \pi(z, \iota)) = (\text{by Property 4})\\
&\theta(w[f(|t_1^+|^z_w, \ldots, |t_k^+|^z_w), z]) = (\text{by definition of coreference})\\
&\theta(|f(t_1, \ldots, t_k)^+|^z_w). \qquad \square
\end{aligned}$$

Lemma 11. *Let t be a term of $\mathcal{ES}$ without standard names whose free variables are among the $x_1, \ldots, x_m$. Let $\iota \in \mathcal{D}_{strt}$, $z \in \mathcal{Z}$, and $w = \omega(\iota)$. Then, given the properties above,*
$$\|do(t[s], s)\|_{\mu^+} = \pi(z \cdot |t^+|^z_w, \iota).$$

Proof. The proof is immediate from Property 7 and Lemma 10. $\square$

Definition 9. For a given variable map μ let u_μ be any $\mathcal{ES}$ variable map which satisfies the following:

1. for any rigid second-order variable V and any w and z, let
$$u_\mu[w, V(n_1, \ldots, n_k), z] = 1 \quad \text{iff} \quad \langle \theta(n_1), \ldots, \theta(n_k)\rangle \in \mu(V);$$

2. for any fluent second-order variable V and any ι and z, let

$$u_\mu\big[\omega(\iota), V(n_1,\ldots,n_k), z\big] = 1 \quad \text{iff} \quad \big\langle \theta(n_1),\ldots,\theta(n_k),\pi(z,\iota)\big\rangle \in \mu(V).$$

Note that u_μ is not necessarily completely specified for fluent variables as there may be worlds in W which are not in the image of ω. Let us call a world which is in the image of ω an ω-world. For example, all worlds in e are ω-worlds because ω is assumed to be onto e (Property 2). Then we have the following:

Lemma 12. *Let $w = \omega(\iota)$ and let u_1 and u_2 be variable maps which agree on all values where the first argument is an ω-world. Then for all basic sentences α,*

$$e, w, z, u_1 \models \alpha \quad \textit{iff} \quad e, w, z, u_2 \models \alpha.$$

Proof. The proof is by induction on the structure of α. All cases except second-order variables are immediate since w and all worlds in e are ω-worlds.

Let $e, w, z, u_1 \models \forall V.\alpha$. Then for all u', if $u' \sim_V u_1$ then $e, w, z, u' \models \alpha$. Now consider any $u'' \sim_V u_2$. Then there clearly is a $u' \sim_V u_1$ which agrees with u'' on V. Also, by definition, this u' agrees with u_1 on all other values and hence agrees with u_2 and therefore u'' on all values for ω-worlds. Thus, by induction, $e, w, z, u'' \models \alpha$. Since this holds for all $u'' \sim_V u_2$, we obtain $e, w, z, u_2 \models \forall V.\alpha$. The other direction is completely symmetric. □

Lemma 13. *$e, w, z, u_\mu \models \forall V.\alpha$ iff for all relations Z over $\mathcal{D}$ (with arity and sorts given by V), $e, w, z, u_{\mu\{V/Z\}} \models \alpha$.*

Proof. We only consider fluent variables V. (Rigids are a simpler special case.) For the only-if direction, suppose $u_{\mu\{V/Z\}}$ is given. Then there is a $u' \sim_V u_\mu$ such that u' agrees with $u_{\mu\{V/Z\}}$ on V and, by definition of $\sim_V$, also agrees with $u_{\mu\{V/Z\}}$ on all variables other than V on all ω-worlds. Hence $u_{\mu\{V/Z\}}$ agrees with u' on all ω-worlds and thus, by assumption and Lemma 12, $e, w, z, u_{\mu\{V/Z\}} \models \alpha$.

Conversely, let $u' \sim_V u_\mu$. Consider

$$Z = \Big\{\big\langle \theta(n_1),\ldots,\theta(n_k),\pi(z,i)\big\rangle \;\Big|\; u'\big[\omega(i), V(n_1,\ldots,n_k), z\big] = 1 \text{ for all } \iota \text{ and } z\Big\}.$$

Then u' agrees with $u_{\mu\{V/Z\}}$ on all ω-worlds and hence $e, w, z, u' \models \alpha$ by assumption and Lemma 12. Since this holds for any $u' \sim_V u_\mu$, we obtain $e, w, z, u_\mu \models \forall V.\alpha$. □

Now we can put all the results together and prove the main lemma:

Lemma 14. *Let α be a basic formula of $\mathcal{ES}$ with no standard names and whose free variables are among the $x_1,\ldots,x_m$. Then, given the properties above, for any variable map μ, any $\mathcal{ES}$ variable map u_μ satisfying Definition 9, any situation variable s, any $z \in \mathcal{Z}$, any $\iota \in \mathcal{D}_{strt}$ and $w = \omega(\iota)$,*

$$e, w, z, u_\mu \models \alpha^+ \quad \textit{iff} \quad \mathcal{M}, \mu^+ \models \alpha[s].$$

Proof. The proof is by induction on the structure of α. There are 12 cases:

1. For a formula of the form $F(t_1,\ldots,t_k)$:
   $e, w, z, u_\mu \models F(t_1,\ldots,t_k)^+$ iff (by definition of satisfaction)
   $w[F(|t_1^+|^z_w,\ldots,|t_k^+|^z_w), z] = 1$ iff (by Property 6)
   $\langle \theta(|t_1^+|^z_w),\ldots,\theta(|t_k^+|^z_w),\pi(z,\iota)\rangle \in F^{\mathcal{M}}$ iff (by Lemma 10)
   $\big\langle \|t_1[s]\|_{\mu^+},\ldots,\|t_k[s]\|_{\mu^+},\pi(z,\iota)\big\rangle \in F^{\mathcal{M}}$ iff (by definition of satisfaction)
   $\mathcal{M}, \mu^+ \models F(t_1[s],\ldots,t_k[s],s)$.

2. For a formula of the form $G(t_1,\ldots,t_k)$:
   $e, w, z, u_\mu \models G(t_1,\ldots,t_k)^+$ iff (by definition of satisfaction)
   $w[G(|t_1^+|^z_w,\ldots,|t_k^+|^z_w), z] = 1$ iff (by the rigidity constraint)
   $w[G(|t_1^+|^z_w,\ldots,|t_k^+|^z_w), \langle\rangle] = 1$ iff (by Property 5)
   $\langle \theta(|t_1^+|^z_w),\ldots,\theta(|t_k^+|^z_w)\rangle \in G^{\mathcal{M}}$ iff (by Lemma 10)
   $\langle \|t_1[s]\|_{\mu^+},\ldots,\|t_k[s]\|_{\mu^+}\rangle \in G^{\mathcal{M}}$ iff (by definition of satisfaction)
   $\mathcal{M}, \mu^+ \models G(t_1[s],\ldots,t_k[s],s)$.

3. For a formula of the form $P(t_1,\ldots,t_k)$:
   $e, w, z, u_\mu \models P(t_1,\ldots,t_k)^+$ iff (by definition of satisfaction)
   $u_\mu[w, P(|t_1^+|^z_w,\ldots,|t_k^+|^z_w), z] = 1$ iff (by definition of u_μ)
   $\langle \theta(|t_1^+|^z_w),\ldots,\theta(|t_k^+|^z_w),\pi(z,\iota)\rangle \in \mu^+[P]$ iff (by Lemma 10)

$\langle \|t_1[s]\|_{\mu^+}, \ldots, \|t_k[s]\|_{\mu^+}, \pi(z,\iota)\rangle \in \mu^+[P]$ iff (by definition of satisfaction)
$\mathcal{M}, \mu^+ \models P(t_1[s], \ldots, t_k[s], s)$.

4. For a formula of the form $Q(t_1, \ldots, t_k)$:
$e, w, z, u_\mu \models Q(t_1, \ldots, t_k)^+$ iff (by definition of satisfaction)
$u_\mu[w, Q(|t_1^+|_w^z, \ldots, |t_k^+|_w^z), z] = 1$ iff (by definition of u_μ)
$\langle \theta(|t_1^+|_w^z), \ldots, \theta(|t_k^+|_w^z), \pi(z,\iota)\rangle \in \mu^+[Q]$ iff (by Lemma 10)
$\langle \|t_1[s]\|_{\mu^+}, \ldots, \|t_k[s]\|_{\mu^+}, \pi(z,\iota)\rangle \in \mu^+[Q]$ iff (by definition of satisfaction)
$\mathcal{M}, \mu^+ \models Q(t_1[s], \ldots, t_k[s], s)$.

5. For a formula of the form $(t_1 = t_2)$:
$e, w, z, u_\mu \models (t_1 = t_2)^+$ iff (by definition of satisfaction)
$|t_1^+|_w^z = |t_2^+|_w^z$ iff (by Property 1)
$\theta(|t_1^+|_w^z) = \theta(|t_2^+|_w^z)$ iff (by Lemma 10)
$\|t_1[s]\|_{\mu^+} = \|t_2[s]\|_{\mu^+}$ iff (by definition of satisfaction and Property 8)
$\mathcal{M}, \mu^+ \models (t_1[s] = t_2[s])$.

6. For a formula of the form $[t]\alpha$:
$e, w, z, u_\mu \models ([t]\alpha)^+$ iff (by definition of satisfaction)
$e, w, z \cdot |t^+|_w^z, u_\mu \models \alpha^+$ iff (by induction)
$\mathcal{M}, \mu^+\{s'/\pi(z \cdot |t^+|_w^z, \iota)\} \models \alpha[s']$ iff
for every $\tau \in \mathcal{D}_{sit}$, if $\tau = \pi(z \cdot |t^+|_w^z, \iota)$, then $\mathcal{M}, \mu^+\{s'/\tau\} \models \alpha[s']$ iff (by Lemma 11)
for every $\tau \in \mathcal{D}_{sit}$, if $\tau = \| do(t[s], s)\|_{\mu^+}$, then $\mathcal{M}, \mu^+\{s'/\tau\} \models \alpha[s']$ iff
for every $\tau \in \mathcal{D}_{sit}$, $\mathcal{M}, \mu^+\{s'/\tau\} \models (s' = do(t[s], s) \supset \alpha[s'])$ iff (by definition of satisfaction)
$\mathcal{M}, \mu^+ \models \forall s'(s' = do(t[s], s) \supset \alpha[s'])$ iff
$\mathcal{M}, \mu^+ \models \alpha[do(t[s], s)]$ iff
$\mathcal{M}, \mu^+ \models ([t]\alpha)[s]$.

7. For a formula of the form $(\alpha \wedge \beta)$:
$e, w, z, u_\mu \models (\alpha \wedge \beta)^+$ iff (by induction)
$\mathcal{M}, \mu^+ \models (\alpha \wedge \beta)[s]$.

8. For a formula of the form $\neg\alpha$:
$e, w, z, u_\mu \models \neg\alpha^+$ iff (by induction)
$\mathcal{M}, \mu^+ \models \neg\alpha[s]$.

9. For a formula of the form $\forall v.\alpha$:
$e, w, z, u_\mu \models (\forall v.\alpha)^+$ iff (by definition of satisfaction)
for all names n of the right sort, $e, w, z, u_\mu \models \alpha^+{}_n^v$ iff (by induction)
for all names n of the right sort, $\mathcal{M}, \mu^+\{v/\theta(n)\} \models \alpha[s]$ iff (by Property 1)
for all $d \in \mathcal{D}$ of the right sort, $\mathcal{M}, \mu^+\{v/d\} \models \alpha[s]$ iff
(by definition of satisfaction) $\mathcal{M}, \mu^+ \models \forall v.\alpha[s]$.

10. For a formula of the form $\forall V.\alpha$:
$e, w, z, u_\mu \models (\forall V.\alpha)^+$ iff (by Lemma 13)
for all relations Z over $\mathcal{D}$, $e, w, z, u_{\mu\{V/Z\}} \models \alpha^+$ iff (by induction)
for all rel. Z over $\mathcal{D}$, $\mathcal{M}, \mu^+\{V/Z\} \models \alpha[s]$ iff (by definition of satisfaction)
$\mathcal{M}, \mu^+ \models \forall V.\alpha[s]$.

11. For a formula of the form $\Box\alpha$:
$e, w, z, u_\mu \models \Box\alpha^+$ iff (by definition of satisfaction)
for all $z' \in \mathcal{Z}$, $e, w, z \cdot z', u_\mu \models \alpha^+$ iff (by induction)
for all $z' \in \mathcal{Z}$, $\mathcal{M}, \mu^+\{s'/\pi(z \cdot z', \iota)\} \models \alpha[s']$ iff
for all $\tau \in \mathcal{D}_{sit}$, if there exists $z' \in \mathcal{Z}$ such that $\tau = \pi(z \cdot z', \iota)$, then $\mathcal{M}, \mu^+\{s'/\tau\} \models \alpha[s']$ iff (by Property 9)
for all $\tau \in \mathcal{D}_{sit}$, $\mathcal{M}, \mu^+\{s'/\tau)\} \models (s \sqsubseteq s' \supset \alpha[s'])$ iff
(by definition of satisfaction) $\mathcal{M}, \mu^+ \models \forall s'(s \sqsubseteq s' \supset \alpha[s'])$.

12. For a formula of the form $Know(\alpha)$:
$e, w, z, u_\mu \models Know(\alpha)^+$ iff (by definition of satisfaction)
for all $w' \in e$, if $w' \simeq_z w$, then $e, w', z, u_\mu \models \alpha^+$ iff (by Property 2)
for all $\omega(\iota') \in e$, if $\omega(\iota') \simeq_z w$, then $e, \omega(\iota'), z, u_\mu \models \alpha^+$ iff (by induction)
for all $\omega(\iota') \in e$, if $\omega(\iota') \simeq_z w$, then $\mathcal{M}, \mu^+\{s'/\pi(z, \iota')\} \models \alpha[s']$ iff
for all $\tau \in \mathcal{D}_{sit}$, if there exists $\iota' \in \mathcal{D}_{strt}$ such that $\tau = \pi(z, \iota')$ where $\omega(\iota') \in e$ and $\omega(\iota') \simeq_z w$,
then $\mathcal{M}, \mu^+\{s'/\tau\} \models \alpha[s']$ iff (by Property 10)
for all $\tau \in \mathcal{D}_{sit}$, $\mathcal{M}, \mu^+\{s'/\tau\} \models (K(s', s) \supset \alpha[s'])$ iff (by definition of satisfaction)

$\mathcal{M}, \mu^+ \models \forall s'(K(s', s) \supset \alpha[s'])$ iff
$\mathcal{M}, \mu^+ \models \mathit{Knows}(\alpha[\mathit{now}], s)$ iff
$\mathcal{M}, \mu^+ \models \mathit{Know}(\alpha)[s]$.

This completes the proof. □

With this lemma in place, we can now prove the correctness of the embedding of $\mathcal{ES}$ into the situation calculus:

Theorem 6. *Let α be any basic sentence of $\mathcal{ES}$ without standard names. Then*

$$\alpha \text{ is valid} \quad \text{iff} \quad \Upsilon \models_{\mathrm{FOL}} \alpha^*,$$

Proof. First assume that α is not valid. Then there is an e, w_0 such that $e, w_0 \not\models \alpha$. Define a Tarski structure as follows:

- The domain of $\mathcal{M}$ is $\mathcal{D} = \mathcal{D}_{obj} \cup \mathcal{D}_{act} \cup \mathcal{D}_{sit}$, where $\mathcal{D}_{obj}$ (resp. $\mathcal{D}_{act}$) is the set of standard names of objects (resp. actions), and $\mathcal{D}_{sit} = \mathcal{Z}$ times W;
- The fixed vocabulary of $\mathcal{M}$ is defined by:
  - $=^{\mathcal{M}}$ is the identity relation over $\mathcal{D}$,
  - $S_0^{\mathcal{M}} = (\langle\rangle, w_0)$,
  - $do^{\mathcal{M}}(n, (z, w)) = (z \cdot n, w)$,
  - $\sqsubset^{\mathcal{M}} = \{((z, w), (z \cdot z', w))\}$,
  - $K^{\mathcal{M}} = \{((z, w), (z, w_0)) \mid w \in e \text{ and } w \simeq_z w_0\}$;
- for every rigid predicate symbol G,
$$G^{\mathcal{M}} = \big\{\langle n_1, \ldots, n_k\rangle \;\big|\; w_0\big[G(n_1, \ldots, n_k), \langle\rangle\big] = 1\big\};$$
- for every fluent predicate symbol F (including *Poss* and *SF*),
$$F^{\mathcal{M}} = \big\{\langle n_1, \ldots, n_k, (z, w)\rangle \;\big|\; w\big[F(n_1, \ldots, n_k), z\big] = 1\big\};$$
- for every rigid function symbol g,
$$g^{\mathcal{M}}(n_1, \ldots, n_k) = w_0\big[g(n_1, \ldots, n_k), \langle\rangle\big];$$
- for every fluent function symbol f,
$$f^{\mathcal{M}}\big(n_1, \ldots, n_k, (z, w)\big) = w\big[g(n_1, \ldots, n_k), z\big].$$

These definitions ensure that $\mathcal{M}$ satisfies Υ. Next, define the mappings θ, π, and ω by letting $\theta(n) = n$, and for any initial $\iota = (\langle\rangle, w)$, letting $\pi(z, \iota) = (z, w)$, and $\omega(\iota) = w$. This ensures that the properties needed for Lemma 14 are satisfied, and so $\mathcal{M} \not\models \alpha^*$. Consequently, $\Upsilon \not\models_{\mathrm{FOL}} \alpha^*$.

Conversely, assume that $\Upsilon \not\models_{\mathrm{FOL}} \alpha^*$. Then there is a Tarski structure $\mathcal{M}$ that satisfies Υ but such that $\mathcal{M} \not\models \alpha^*$. The domain $\mathcal{D}$ must be $\mathcal{D}_{sit} \cup \mathcal{D}_{act} \cup \mathcal{D}_{obj}$, with $\mathcal{D}_{strt} \subseteq \mathcal{D}_{sit}$ as the set of starting situations, and with $\iota_0 = S_0^{\mathcal{M}} \in \mathcal{D}_{strt}$. Since $\mathcal{M} \models \Upsilon$, both $\mathcal{D}_{obj}$ and $\mathcal{D}_{act}$ are countably infinite, say $\mathcal{D}_{obj} = \{\delta_1, \delta_2, \ldots\}$, and $\mathcal{D}_{act} = \{\lambda_1, \lambda_2, \ldots\}$. We define the mappings θ, π and ω by the following:

- θ maps the i-th standard name for objects to δ_i, and the i-th standard name for actions to λ_i;
- for any $z \in \mathcal{Z}$ and $\iota \in \mathcal{D}_{strt}$, we define $\pi(z, \iota)$ by:
$$\pi(\langle\rangle, \iota) = \iota,$$
$$\pi(z \cdot n, \iota) = do^{\mathcal{M}}\big(\theta(n), \pi(z, \iota)\big);$$
- for any $\iota \in \mathcal{D}_{strt}$, we let $\omega(\iota)$ be the world w defined by the following:
$$w\big[G(n_1, \ldots, n_k), z\big] = 1 \quad \text{iff} \quad \big\langle\theta(n_1), \ldots, \theta(n_k)\big\rangle \in G^{\mathcal{M}},$$
$$w\big[F(n_1, \ldots, n_k), z\big] = 1 \quad \text{iff} \quad \big\langle\theta(n_1), \ldots, \theta(n_k), \pi(z, \iota)\big\rangle \in F^{\mathcal{M}},$$
$$w\big[g(n_1, \ldots, n_k), z\big] = \theta^{-1}\big(g^{\mathcal{M}}\big(\theta(n_1), \ldots, \theta(n_k)\big)\big),$$
$$w\big[f(n_1, \ldots, n_k), z\big] = \theta^{-1}\big(f^{\mathcal{M}}\big(\theta(n_1), \ldots, \theta(n_k), \pi(z, \iota)\big)\big).$$

For any $\iota \in \mathcal{D}_{strt}$, $\omega(\iota)$ obviously satisfies the rigidity constraint. Finally, we let

- $e = \{\omega(\iota) \mid (\iota, \iota_0) \in K^{\mathcal{M}}\}$;
- $w_0 = \omega(\iota_0)$.

These definitions ensure that all of the properties needed for Lemma 14 are satisfied, and so $e, w_0 \not\models \alpha$. Consequently, α is not valid. □

Theorem 7. *Let α be any sentence of $\mathcal{ES}$ without standard names. Then*

$$\alpha \text{ is valid} \quad \text{iff} \quad \Sigma \cup \Upsilon \models_{\text{FOL}} \alpha^*.$$

Proof. The proof is almost the same as the previous one because we can re-use the same model constructions.

Let us first assume that α is not valid. Then there is an e, w_0 such that $e, w_0 \not\models \alpha$. Now define a Tarski structure $\mathcal{M}$ exactly as in the first part of the proof of Theorem 6. It is easy to see that Σ (as well as Υ) is satisfied by $\mathcal{M}$. In particular, the construction ensures that situations have unique names and that neither S_0 nor any situation K-accessible from S_0 has a predecessor, that is, these are truly initial situations. Furthermore, by construction, the situations reachable from the initial situations are the only situations of the model so that the induction axiom for situations (Axiom 2) is also satisfied. The rest of the argument is exactly as in Theorem 6, from which $\Sigma \cup \Upsilon \not\models_{\text{FOL}} \alpha^*$ follows.

Conversely, assume that $\Sigma \cup \Upsilon \not\models_{\text{FOL}} \alpha^*$. Then there is a Tarski structure $\mathcal{M}$ that satisfies $\Sigma \cup \Upsilon$ but such that $\mathcal{M} \not\models \alpha^*$. We define the mappings θ, π and ω as well as w_0 and e exactly as in Theorem 6. As before, all the properties of Lemma 14 are satisfied and, hence, $e, w_0 \not\models \alpha$, which proves that α is not valid. □

References

[1] P. Blackburn, J. Kamps, M. Marx, Situation calculus as hybrid logic: first steps, in: P. Brazdil, A. Jorge (Eds.), Progress in Artificial Intelligence, in: Lecture Notes in Artificial Intelligence, vol. 2258, Springer-Verlag, 2001, pp. 253–260.
[2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot, Artificial Intelligence 114 (1–2) (2000).
[3] M.A. Castilho, O. Gasquet, A. Herzig, Formalizing action and change in modal logic I. The frame problem, Journal of Logic and Computation 9 (5) (1999) 701–735.
[4] G. De Giacomo, M. Lenzerini, PDL-based framework for reasoning about actions, in: Proc. of AI*IA, in: Lecture Notes in Artificial Intelligence, vol. 992, 1995, pp. 103–114.
[5] R. Demolombe, Belief change: from situation calculus to modal logic, in: IJCAI Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'03), Acapulco, Mexico, 2003.
[6] R. Demolombe, A. Herzig, I.J. Varzinczak, Regression in modal logic, Journal of Applied Non-Classical Logics 13 (2) (2003) 165–185.
[7] R. Fagin, J. Halpern, Y. Moses, M. Vardi, Reasoning about Knowledge, MIT Press, Cambridge, 1995.
[8] A. Finzi, F. Pirri, R. Reiter, Open world planning in the situation calculus, in: Proc. of the 7th Conference on Artificial Intelligence (AAAI-00), AAAI Press, 2000, pp. 754–760.
[9] Michael Gelfond, Vladimir Lifschitz, Representing action and change by logic programs, Journal of Logic Programming 17 (1993) 301–321.
[10] D. Harel, Dynamic logic, in: D. Gabbay, F. Guenther (Eds.), Handbook of Philosophical Logic, vol. 2, D. Reidel Publishing Company, 1984, pp. 497–604.
[11] A. Herzig, J. Lang, D. Longin, T. Polacsek, A logic for planning under partial observability, in: Proc. of the Seventeenth National Conference on Artificial Intelligence (AAAI-00), AAAI Press, 2000, pp. 768–773.
[12] J. Hintikka, Knowledge and Belief, Cornell University Press, Ithaca, 1962.
[13] S. Hölldobler, J. Schneeberger, A new deductive approach to planning, New Generation Computing 8 (1990) 225–244.
[14] G. Hughes, M. Cresswell, An Introduction to Modal Logic, Methuen and Co., London, 1968.
[15] D. Kaplan, Quantifying-in, in: L. Linsky (Ed.), Reference and Modality, Oxford University Press, Oxford, 1971, pp. 112–144.
[16] R. Kowalski, M. Sergot, A logic based calculus of events, New Generation Computing 4 (1986) 67–95.
[17] S.A. Kripke, Semantical considerations on modal logic, Acta Philosophica Fennica 16 (1963) 83–94.
[18] S.A. Kripke, Is there a problem with substitutional quantification?, in: G. Evans, J. McDowell (Eds.), Truth and Meaning, Clarendon Press, Oxford, 1976, pp. 325–419.
[19] G. Lakemeyer, H.J. Levesque, $\mathcal{AOL}$: a logic of acting, sensing, knowing, and only knowing, in: Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, 1998, pp. 316–327.
[20] G. Lakemeyer, H.J. Levesque, A semantical account of progression in the presence of defaults, in: Proc. of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, USA, 2009, pp. 842–847.
[21] H.J. Levesque, All I Know: A Study in Autoepistemic Logic, Artificial Intelligence, vol. 42, North Holland, 1990, pp. 263–309.
[22] H.J. Levesque, G. Lakemeyer, The Logic of Knowledge Bases, MIT Press, 2001.
[23] G. Lakemeyer, H.J. Levesque, Situations si, situation terms no, in: Proc. of the Ninth Conference on Principles of Knowledge Representation and Reasoning (KR2004), 2004.
[24] G. Lakemeyer, H.J. Levesque, A useful fragment of the situation calculus, in: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), 2005, pp. 490–496.
[25] G. Lakemeyer, H.J. Levesque, Towards an axiom system for default logic, in: Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06), 2006.
[26] H.J. Levesque, F. Pirri, R. Reiter, Foundations for the situation calculus, Linköping Electronic Articles in Computer and Information Science 3 (18) (1998).
[27] H.J. Levesque, R. Reiter, Y. Lespérance, F. Lin, R.B. Scherl, Golog: a logic programming language for dynamic domains, Journal of Logic Programming 31 (1997) 59–84.
[28] F. Lin, R. Reiter, How to progress a database, Artificial Intelligence 92 (1997) 131–167.
[29] J. McCarthy, Situations, actions and causal laws, Technical report, Stanford University, 1963. Also in: M. Minsky (Ed.), Semantic Information Processing, MIT Press, Cambridge, MA, 1968, pp. 410–417.
[30] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Mitchie, M. Swann (Eds.), Machine Intelligence, vol. 4, Edinburgh University Press, 1969, pp. 463–502.
[31] S. McIlraith, T.C. Son, Adapting Golog for composition of semantic web services, in: Proc. of the Eighth International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, 2002, pp. 482–493.
[32] R.C. Moore, A formal theory of knowledge and action, in: J.R. Hobbs, R.C. Moore (Eds.), Formal Theories of the Commonsense World, Ablex, Norwood, NJ, 1985, pp. 319–358.

[33] V.R. Pratt, Semantical considerations on Floyd–Hoare logic, in: Proc. 17th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1976, pp. 109–121.
[34] A. Prior, Past, Present and Future, Oxford University Press, 1967.
[35] 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.
[36] R. Reiter, Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems, MIT Press, 2001.
[37] R. Reiter, On knowledge-based programming with sensing in the situation calculus, ACM Transactions on Computational Logic (2001) 433–457.
[38] Erik Sandewall, Features and Fluents. The Representation of Knowledge about Dynamical Systems, Oxford University Press, 1994.
[39] R.B. Scherl, H.J. Levesque, Knowledge, action, and the frame problem, Artificial Intelligence 144 (1–2) (2003) 1–39.
[40] Michael Thielscher, From situation calculus to fluent calculus: state update axioms as a solution to the inferential frame problem, Artificial Intelligence 111 (1–2) (1999) 277–299.
[41] S. Vassos, H.J. Levesque, On the progression of situation calculus basic action theories: resolving a 10-year-old conjecture, in: Proc. of the 20th AAAI Conference on Artificial Intelligence (AAAI-08), 2008.