A Semantical Account of Progression in the Presence of Defaults

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

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

Abstract

In previous work, we proposed a modal fragment of the situation calculus called $\mathcal{ES}$, which fully captures Reiter's basic action theories. $\mathcal{ES}$ also has epistemic features, including only-knowing, which refers to all that an agent knows in the sense of having a knowledge base. While our model of only-knowing has appealing properties in the static case, it appears to be problematic when actions come into play. First of all, its utility seems to be restricted to an agent's initial knowledge base. Second, while it has been shown that only-knowing correctly captures default inferences, this was only in the static case, and undesirable properties appear to arise in the presence of actions. In this paper, we remedy both of these shortcomings and propose a new dynamic semantics of only-knowing, which is closely related to Lin and Reiter's notion of progression when actions are performed and where defaults behave properly.

Introduction

In previous work, Lakemeyer and Levesque (2004; 2005) proposed a modal fragment of the situation calculus called $\mathcal{ES}$, which fully captures Reiter's basic action theories and regression-based reasoning, including reasoning about knowledge. So, for example, the language allows us to formulate Reiter-style successor state axioms such as this one:

$$\begin{aligned}&\forall a, x.\Box([a]\mathit{Broken}(x) \ \equiv\\ &\quad (a = \mathit{drop}(x) \land \mathit{Fragile}(x)) \ \lor\\ &\quad (\mathit{Broken}(x) \land a \neq \mathit{repair}(x)))\end{aligned}$$

In English: after any sequence of actions ($\Box$), an object x will be broken after doing action a ($[a]\mathit{Broken}(x)$) 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. Here we assume that *Fragile* is a predicate which is not affected by any action so that its successor state axiom would be

$$\forall a, x.\Box([a]\mathit{Fragile}(x) \ \equiv \ \mathit{Fragile}(x)).$$

Let us call the conjunction of these two axioms SSA_{BF}. In addition to action and change, the language $\mathcal{ES}$ also addresses what an agent knows and only-knows. The latter is intended to capture all an agent knows in the sense of having a knowledge base. For illustration, consider the following sentence, which is logically valid in $\mathcal{ES}$:

$$\begin{aligned}&\boldsymbol{O}(\mathit{Fragile}(o) \land \neg\mathit{Broken}(o) \land \mathit{SSA}_{BF}) \ \supset\\ &\qquad [\mathit{drop}(o)]\,(\boldsymbol{K}(\mathit{Broken}(o)) \land \neg\boldsymbol{K}(\mathit{Glass}(o)))\ .\end{aligned}$$

In English: if all the agent knows is that *o* is fragile and not broken and that the successor state axioms for *Broken* and *Fragile* hold, then after dropping *o*, the agent knows that *o* is broken, but does not know that *o* is made of glass.

Let us now consider what the agent should only-know after the drop action has occurred. Intuitively, the agent's knowledge should change in that it now believes that *o* is broken, with everything else remaining the same. Formally,

$$[\mathit{drop}(o)]\,\boldsymbol{O}(\mathit{Fragile}(o) \land \mathit{Broken}(o) \land \mathit{SSA}_{BF}).$$

In fact this view corresponds essentially to what Lin and Reiter (LR) [1997] call the *progression* of a database wrt an action. It turns out, however, that the semantics of only-knowing as proposed in (Lakemeyer and Levesque 2004) differs from this in that the last formula above is *not* entailed. The reason is that their version, unlike progression, does not forget what was true initially (like whether or not *o* was already broken), and so more ends up being known.

The LR notion of progression allows for efficient implementations under certain restrictions (Lin and Reiter 1997; Liu and Levesque 2005; Vassos and Levesque 2007), and being able to forget the past seems essential for this. Hence the previous semantics of only-knowing may not be very useful, except perhaps in the initial state. In this paper, we present a new semantics of only-knowing which avoids this pitfall and is fully compatible with LR's idea of progression.

Levesque (1990) showed that only-knowing in the static case also accounts for default reasoning in the sense of autoepistemic logic (Moore 1985). For example, the default that objects are fragile unless known otherwise can be written as

$$\forall x \neg\boldsymbol{K}\neg\mathit{Fragile}(x) \supset \mathit{Fragile}(x).$$

If the agent uses this default instead of the fact that *o* is fragile then it would still conclude, this time by default, that *o* is fragile and hence believe that it is broken after dropping it. But suppose that *o* is actually *not* fragile. What should the agent believe after *sensing* the status of *o*'s fragility? Clearly, it should then believe that *o* is indeed not fragile and it should not believe that dropping *o* will break it. That is, the default should no longer apply. Unfortunately, the previous definition of only-knowing does not do this. The problem, roughly, is that the initial default conclusion that *o* is fragile cannot be distinguished from a hard fact. Subsequently sensing the opposite then leads to an inconsistency.

In this paper we will fix this problem by proposing a semantics which separates conclusions based on facts from those based on defaults. To this end, we will distinguish between what is known for sure (using the modality $\boldsymbol{K}$) and what is believed after applying defaults (using another modality $\boldsymbol{B}$). In fact, defaults themselves will be formulated using $\boldsymbol{B}$ instead of $\boldsymbol{K}$. All this will be integrated with progression in the sense that defaults will be applied to the progressed knowledge base.

For space reasons, the paper, which also appears in (Lakemeyer and Levesque 2009a) in almost identical form, contains no proofs. These and a comparison between the old and new semantics of only-knowing and between our notion of progression and that of LR can be found in (Lakemeyer and Levesque 2009b).

The rest of the paper is organized as follows. In the next section, we introduce the logic $\mathcal{ES}_{\text{O}}$, which is like the old $\mathcal{ES}$ except for the new semantics of only-knowing and defaults. This semantics agrees with the previous one in the static case. After that, we consider only-knowing in the context of basic action theories. In particular, we show that what is only-known after an action extends LR's original idea of progression, and how reasoning about defaults fits into the picture. We then address related work and conclude.

The Logic $\mathcal{ES}_{\text{O}}$

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:

- *rigid terms*: The ground terms of the language are taken to be isomorphic to the domain of discourse. This allows first-order quantification to be understood substitutionally. Equality can also be given a very simple treatment: two ground terms are equal only if they are identical.
- *knowledge and truth*: The language includes modal operators $\boldsymbol{K}$ and $\boldsymbol{B}$ for knowledge and belief. The $\boldsymbol{K}$ operator allows us to distinguish between sentences that are true and sentences that are known (by some implicit agent). The $\boldsymbol{B}$ operator allows an agent to have false beliefs about its world or how its world changes. For example, we can model situations where an object is not fragile but the agent does not know it, yet may believe that it is fragile by default.
- *sensing*: 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).[1] Just as an action theory may contain precondition axioms characterizing the conditions under which *Poss* holds, it can contain axioms characterizing the conditions under which *SF* holds.

The language

The symbols of $\mathcal{ES}_{\text{O}}$ consist of first-order variables, second-order predicate variables of every arity, rigid functions of every arity, fluent predicate symbols of every arity, as well as these connectives and other symbols: $=$, $\wedge$, $\neg$, $\forall$, $\boldsymbol{K}$, $\boldsymbol{B}$, $\boldsymbol{O}$, $\boldsymbol{\Omega}$, $\square$, round and square parentheses, period, comma. We assume two special fluent predicates *Poss* and *SF* (for sensing). $\boldsymbol{K}$, $\boldsymbol{B}$, $\boldsymbol{O}$, and $\boldsymbol{\Omega}$ are called epistemic operators.

The *terms* of the language are formed in the usual way from first-order variable and rigid functions.

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

1. If $t_1, \ldots, t_k$ are terms, F is a k-ary predicate symbol, and V is a k-ary second-order variable, then $F(t_1, \ldots, t_k)$ and $V(t_1, \ldots, t_k)$ are (atomic) formulas;
2. If t_1 and t_2 are terms, then $(t_1 = t_2)$ is a formula;
3. If α and β are formulas, v is a first-order variable, V is a second-order variable, and t is a term, then the following are also formulas: $(\alpha \wedge \beta)$, $\neg\alpha$, $\forall v.\, \alpha$, $\forall V.\, \alpha$, $[t]\alpha$, $\square\alpha$, $\boldsymbol{K}\alpha$, $\boldsymbol{B}\alpha$, $\boldsymbol{O}\alpha$, and $\boldsymbol{\Omega}\alpha$, where the formulas following $\boldsymbol{O}$ and $\boldsymbol{\Omega}$ are restricted further below.

We read $[t]\alpha$ as "α holds after action t", and $\square\alpha$ as "α holds after any sequence of actions," and $\boldsymbol{K}\alpha$ ($\boldsymbol{B}\alpha$) as "the agent knows (believes) α." $\boldsymbol{O}\alpha$ may be read as "the agent only-knows α" and is intended to capture all the agent knows about what the world is like now and how it evolves as a result of actions. Here no defaults are taken into account, just facts which, as we will see later, come in the form of a basic action theory similar to those proposed by Reiter (2001a). Therefore, we restrict $\boldsymbol{O}$ to apply to so-called *objective formulas* only, which are those mentioning no epistemic operators. Finally, $\boldsymbol{\Omega}\alpha$ is meant to capture all and only the defaults believed by the agent. For that, α is restricted to what we call *static belief formulas*, which mention neither $\square$ nor $[t]$ nor any epistemic operator except $\boldsymbol{B}$.

As usual, we treat $(\alpha \vee \beta)$, $(\alpha \supset \beta)$, $(\alpha \equiv \beta)$, $\exists v.\, \alpha$, and $\exists V.\, \alpha$ as abbreviations. We use α^x_t to mean formula α with all free occurrences of variable x replaced by term t. We call a formula without free variables a *sentence*.

We will also sometimes refer to *static objective formulas*, which are the objective formulas among the static belief formulas, and *fluent formulas*, which are formulas with no $\boldsymbol{K}$, $\boldsymbol{O}$, $\boldsymbol{B}$, $\boldsymbol{\Omega}$, $\square$, $[t]$, *Poss*, or *SF*.[3]

[1] For convenience, we assume that every action returns a (perhaps trivial) sensing result. Here, we restrict ourselves to binary values. See (Scherl and Levesque 2003) for how to handle arbitrary sensing results.

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

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

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. A world determines truth values for the ground atoms after any sequence of actions. An epistemic state is defined by a set of worlds, as in possible-world semantics.

More precisely, let $\mathcal{Z}$ be the set of all finite sequences of elements of $\mathcal{R}$ including $\langle\,\rangle$, the empty sequence. $\mathcal{Z}$ should be understood as the set of all finite sequences of actions. Then

- a world $w \in W$ is any function from $\mathcal{G}$ (the set of ground atoms) and $\mathcal{Z}$ to $\{0,1\}$.
- an epistemic state $e \subseteq W$ is any set of worlds.

To interpret formulas with free variables, we proceed as follows. First-order variables are handled substitutionally using the rigid terms $\mathcal{R}$. To handle the quantification over second-order variables, we use second-order *variable maps* defined as follows:

> The *second-order ground atoms* are formulas of the form $V(t_1, \ldots, t_k)$ where V is a second-order variable and all of the t_i are in $\mathcal{R}$. A *variable map* u is a function from second-order ground atoms to $\{0,1\}$.

Let u and u' be variable maps, and let V be a second-order variable; we write $u' \sim_V u$ to mean that u and u' agree except perhaps on the assignments 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 on the sensing throughout action sequence z) inductively by the following:

1. $w' \simeq_{\langle\,\rangle} w$ for all w';
2. $w' \simeq_{z \cdot t} w$ iff $w' \simeq_z w$, $w'[\mathit{Poss}(t), z] = 1$ and $w'[\mathit{SF}(t), z] = w[\mathit{SF}(t), z]$.

Note that $\simeq_z$ is not quite an equivalence relation because of the use of *Poss* here. 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, we now turn to the semantic definitions for sentences of $\mathcal{ES}_{\mathrm{O}}$. Given an epistemic state $e \subseteq W$, a world $w \in W$, an action sequence $z \in \mathcal{Z}$, and a second-order variable map u, we have:

1. $e, w, z, u \models F(t_1, \ldots, t_k)$ iff $w[F(t_1, \ldots, t_k), z] = 1$;
2. $e, w, z, u \models V(t_1, \ldots, t_k)$ iff $u[V(t_1, \ldots, t_k)] = 1$;
3. $e, w, z, u \models (t_1 = t_2)$ iff t_1 and t_2 are identical;
4. $e, w, z, u \models [t]\alpha$ iff $e, w, z \cdot t, u \models \alpha$;
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 $e, w, z, u \models \alpha^x_t$, for all $t \in \mathcal{R}$;
8. $e, w, z, u \models \forall V.\, \alpha$ iff $e, w, z, u' \models \alpha$, for all $u' \sim_V u$;
9. $e, w, z, u \models \Box\alpha$ iff $e, w, z \cdot z', u \models \alpha$, for all $z' \in \mathcal{Z}$;

To define the meaning of the epistemic operators, we need the following definition:

Definition 1 *Let w be a world and e a set of worlds, and z a sequence of actions. Then*

1. *w_z is a world such that $w_z[p, z'] = w[p, z \cdot z']$ for all ground atoms p and action sequences z';*
2. *$e^w_z = \{w'_z \mid w' \in e$ and $w' \simeq_z w\}$.*

Note that w_z is exactly like w after the actions z have occurred. So in a sense, w_z can be thought of as the progression of w wrt z. e^w_z then contains all those worlds of e which are progressed wrt z and which are compatible with (the real) world w in terms of the sensing results and where all the actions in z are executable. Note that when z is empty, $e^w_z = e$.

10. $e, w, z, u \models \boldsymbol{K}\alpha$ iff for all $w' \in e^w_z$, $e^w_z, w', \langle\rangle, u \models \alpha$;
11. $e, w, z, u \models \boldsymbol{O}\alpha$ iff for all w', $w' \in e^w_z$ iff $e^w_z, w', \langle\rangle, u \models \alpha$.

In other words, knowing α in e and w after actions z means that α is true in all the progressed worlds of e which are compatible with w. $\boldsymbol{O}\alpha$ is quite similar except for the "iff," whose effect is that e^w_z must contain every world which satisfies α.

$\boldsymbol{B}$ and $\boldsymbol{\Omega}$ are meant to capture what the agent believes in addition by applying defaults. Having more beliefs (as a result of defaults) is modeled by considering a subset of the worlds in e^w_z. For that purpose, we introduce a function δ which maps each set of worlds into a subset. In particular, we require that $\delta(e^w_z) \subseteq e^w_z$. As δ is now part of the model (just like w and e) we add it to the L.H.S. of the satisfaction relation with the understanding that the previous rules are retrofitted with δ as well. Then we have:

12. $e, w, z, u, \delta \models \boldsymbol{B}\alpha$ iff for all $w' \in \delta(e^w_z)$, $e^w_z, w', \langle\rangle, u, \delta \models \alpha$;
13. $e, w, z, u, \delta \models \boldsymbol{\Omega}\alpha$ iff for all $w' \in e^w_z$, $w' \in \delta(e^w_z)$ iff $e^w_z, w', \langle\rangle, u, \delta \models \alpha$.

Note that the only difference between $\boldsymbol{K}$ and $\boldsymbol{B}$ is that the latter considers $\delta(e^w_z)$ instead of e^w_z. Likewise, the definition of $\boldsymbol{\Omega}$ is similar to that of $\boldsymbol{O}$. The role of $\boldsymbol{\Omega}$ is to constrain δ to produce a special subset of e^w_z. Roughly, the effect of the definition of $\boldsymbol{\Omega}\alpha$ is that one starts with whatever facts are believed (represented by e^w_z) and then settles on a largest subset of e^w_z such that α (representing the defaults) is also believed.

We say that a sentence in $\mathcal{ES}_{\mathrm{O}}$ is true at a given e, w, and δ (written $e, w, \delta \models \alpha$) if $e, w, \langle\,\rangle, u, \delta \models \alpha$ for any second-order variable map u. If 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, w, and δ, if $e, w, \delta \models \alpha'$ for every

[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.

$\alpha' \in \Sigma$, then $e, w, \delta \models \alpha$. Finally, we write $\models \alpha$ (read: α is valid) to mean $\{\} \models \alpha$.

For reasons of space we cannot go into details about the general logical properties of the epistemic operators. To demonstrate that the operators are well-behaved, we only list some properties, which all have simple semantic proofs:
$\models \Box(\boldsymbol{K}\alpha \supset \boldsymbol{B}\alpha)$
$\models \Box(\boldsymbol{O}\alpha \supset \boldsymbol{K}\alpha)$
$\models \Box(\boldsymbol{\Omega}\alpha \supset \boldsymbol{B}\alpha)$

Moreover, $\boldsymbol{K}$ and $\boldsymbol{B}$ satisfy the usual $K45$ axioms of modal logic (Hughes and Cresswell 1968) and they are mutually introspective, e.g. $\models \Box(\boldsymbol{B}\alpha \supset \boldsymbol{KB}\alpha)$.

The Semantics of Progression and Defaults

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 (2001a) second-order induction axiom for situations. The treatment of defaults is deferred to Section .

Definition 2 *Given a set of fluents $\mathcal{F}$, a set $\Sigma \subseteq \mathcal{ES}_0$ of sentences is called a basic action theory over $\mathcal{F}$ iff* $\Sigma = \Sigma_0 \cup \Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$ *where*

1. *Σ_0 is any set of fluent sentences;*
2. *Σ_{pre} is a singleton sentence of the form $\Box Poss(a) \equiv \pi$, where π is a fluent formula;*[5]
3. *Σ_{post} is a set of sentences of the form $\Box[a]F(\vec{v}) \equiv \gamma_F$, one for each relational fluent F in $\mathcal{F}$, respectively, and where the γ_F are fluent formulas.*[6]
4. *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 in $\mathcal{F}$, which incorporate the solution to the frame problem proposed by Reiter (1991). Σ_{sense} characterizes the sensing results of actions. For actions like *drop*(o), which do not return any useful sensing information, *SF* can be defined to be vacuously true (see below for an example).

We will usually require that Σ_{pre}, Σ_{post}, and Σ_{sense} be first-order. However, Σ_0 may contain second-order sentences. As we will see, this is inescapable if we want to capture progression correctly. In the following, we assume that Σ (and hence $\mathcal{F}$) is finite and we will freely use Σ or its subsets as part of sentences with the understanding that we mean the conjunction of the sentences contained in the set.

Progression = Only-knowing after an action

Let us now turn to the first main result of this paper. The question we want to answer is this: suppose an agent is given a basic action theory as its initial knowledge base; how do we characterize the agent's knowledge after an action is performed? As hinted in the introduction, only-knowing will give us the answer.

In the following, for a given basic action theory Σ, we sometimes write ϕ for Σ_0 and $\Box\beta$ for the rest of the action theory $\Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$. We assume that π and φ refer to the right-hand sides of the definitions of *Poss* and *SF* in Σ, and γ_F is the right-hand side of the successor state axiom for fluent F. Also, let $\vec{F}$ consist of all the fluent predicate symbols in Σ, and let $\vec{P}$ be corresponding second-order variables, where each P_i has the same arity as F_i. Then $\alpha^{\vec{F}}_{\vec{P}}$ denotes the formula α with every occurrence of F_i replaced by P_i.

The following result characterizes in general terms all that is known after performing an action:

Theorem 1 *Let $\Sigma = \phi \wedge \Box\beta$ be a basic action theory and t an action term. Then*

$$\begin{aligned}\models \boldsymbol{O}(\phi \wedge \Box\beta) \supset\ & \\ & (SF(t) \supset [t]\boldsymbol{O}(\Psi \wedge \Box\beta)) \wedge \\ & (\neg SF(t) \supset [t]\boldsymbol{O}(\Psi' \wedge \Box\beta)),\end{aligned}$$

where $\Psi = \exists\vec{P}[(\phi \wedge \pi^a_t \wedge \varphi^a_t)^{\vec{F}}_{\vec{P}} \wedge \bigwedge \forall\vec{x}.F(\vec{x}) \equiv {\gamma_F}^{a}_{t}{}^{\vec{F}}_{\vec{P}}]$ *and*

$$\Psi' = \exists\vec{P}[(\phi \wedge \pi^a_t \wedge \neg\varphi^a_t)^{\vec{F}}_{\vec{P}} \wedge \bigwedge \forall\vec{x}.F(\vec{x}) \equiv {\gamma_F}^{a}_{t}{}^{\vec{F}}_{\vec{P}}].$$

What the theorem says is that if all the agent knows initially is a basic action theory, then after doing action t all the agent knows is another basic action theory, where the dynamic part ($\Box\beta$) remains the same and the initial database ϕ is replaced by Ψ or Ψ', depending on the outcome of the sensing. Note that the two sentences differ only in one place, φ^a_t vs. $\neg\varphi^a_t$. Roughly, Ψ and Ψ' specify how the truth value of each fluent F in $\mathcal{F}$ is determined by what was true previously (ϕ), taking into account that the action was possible (π^a_t) and that the sensing result was either true (φ^a_t) or false ($\neg\varphi^a_t$). Since after performing an action, the agent again only-knows a basic action theory, we can take this as its new initial theory and the process can iterate. We remark that our notion of progression is very closely related to progression as defined by (Lin and Reiter 1997), but extends it to handle sensing actions. Note that, while Lin and Reiter need to include the unique names axioms for actions in the progression, we do not, as these are built into the logic.

We mentioned above that after an action, the resulting knowledge base can be taken as the new initial knowledge base, and the progression can iterate. The following theorem shows that this view is justified in that the entailments about the future remain the same when we substitute what is known about the world initially by its progression. Here we only consider the case where *SF*(t) is true.

Theorem 2 $\models \boldsymbol{O}(\phi \wedge \Box\beta) \wedge SF(t) \supset [t]\boldsymbol{K}(\alpha)$ *iff*
$\models \boldsymbol{O}(\Psi \wedge \Box\beta) \supset \boldsymbol{K}(\alpha)$.

In English (roughly): It follows from your initial knowledge base that you will know α after doing action t iff knowing α follows from your progressed knowledge base.

[5] We assume that all free variables are implicitly universally quantified and that $\Box$ has lower syntactic precedence than the logical connectives, so that $\Box Poss(a) \equiv \pi$ stands for the sentence $\forall a.\Box(Poss(a) \equiv \pi)$.

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

Defaults for basic action theories

Here we restrict ourselves to static defaults like "birds normally fly." In an autoepistemic setting (Moore 1985; Levesque 1990), these have the following form:

$$\forall\vec{x}.\boldsymbol{B}\alpha \land \neg\boldsymbol{B}\neg\beta \supset \gamma,$$

which can be read as "if α is believed and β is consistent with what is believed then assume γ." Here the assumption is that α, β, and γ are static objective formulas.

Let Σ_{def} be the conjunction of all defaults of the above form held by an agent. For a given basic action theory Σ, as defined in Section , the idea is to apply the same defaults to what is known about the current situation after any number of actions have occurred, that is, for the purpose of default reasoning, we assume that $\Box\Omega\Sigma_{\text{def}}$ holds. The following theorem relates what is then believed after one action has occurred (where *SF* returns true) with stable expansions (Moore 1985).[7]

Theorem 3 *Let t be a ground action and $\Sigma = \phi \land \Box\beta$ a basic action theory such that $\models \boldsymbol{O}\Sigma \land \mathit{SF}(t) \supset [t]\boldsymbol{O}(\psi \land \Box\beta)$ and ψ is first order. Then for any static belief sentence γ,*

$$\models \boldsymbol{O}\Sigma \land \mathit{SF}(t) \land \Box\Omega\Sigma_{\text{def}} \supset [t]\boldsymbol{B}\gamma \quad \textit{iff}$$
$$\gamma \textit{ is in every stable expansion of } \psi \land \Sigma_{\text{def}}.$$

An example

To illustrate progression, let us consider the example of the introduction with two fluents *Broken* and *Fragile*, actions $\mathit{drop}(x)$, $\mathit{repair}(x)$, and $\mathit{senseF}(x)$ (for sensing whether x is fragile). First, we let the basic action theory Σ consist of the following axioms:

- $\Sigma_0 = \{\mathit{Fragile}(o), \neg\mathit{Broken}(o)\}$;
- $\Sigma_{\text{pre}} = \{\Box\mathit{Poss}(a) \equiv \mathit{true}\}$ (for simplicity);
- $\Sigma_{\text{post}} = \{\mathit{SSA}_{BF}\}$ (from the introduction);
- $\Sigma_{\text{sense}} = \{\Box\mathit{SF}(a) \equiv \exists x.a = \mathit{drop}(x) \land \mathit{true} \lor a = \mathit{repair}(x) \land \mathit{true} \ \lor \ a = \mathit{senseF}(x) \land \mathit{Fragile}(x)\}$.

As before, let $\Box\beta$ be $\Sigma_{\text{pre}} \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$. Then we have

$$\models \Sigma \land \boldsymbol{O}(\Sigma_0 \land \Box\beta) \supset [\mathit{drop}(o)]\boldsymbol{O}(\Psi \land \Box\beta),$$

where $\Psi = \exists P, P'.[\neg P(o) \land P'(o) \land$
$\quad \exists x.\mathit{drop}(o) = \mathit{drop}(x) \land \mathit{true} \lor$
$\quad\quad \mathit{drop}(o) = \mathit{repair}(x) \land \mathit{true} \lor$
$\quad\quad \mathit{drop}(o) = \mathit{senseF}(x) \land P'(x) \land$
$\quad \forall x.\, \mathit{Broken}(x) \equiv \mathit{drop}(o) = \mathit{drop}(x) \land P'(x) \lor$
$\quad\quad P(x) \land \mathit{drop}(o) \neq \mathit{repair}(x) \land$
$\quad \forall x.\, \mathit{Fragile}(x) \equiv P'(x)]$.

Using the fact that all actions are distinct, it is not difficult to see that Ψ can be simplified to

$$(\mathit{Fragile}(o) \land \mathit{Broken}(o)).$$

In other words, after dropping *o*, the agent's knowledge base is as before, except that *o* is now known to be broken.

To see how defaults work, we now let Σ be as before except that $\Sigma_0 = \{\neg\mathit{Broken}(o)\}$ and let $\Sigma' = \Sigma \cup \{\neg\mathit{Fragile}(o)\}$. Let $\Sigma_{\text{def}} = \{\forall x.\neg\boldsymbol{B}\neg\mathit{Fragile}(x) \supset \mathit{Fragile}(x)\}$. Then the following are logical consequences of

$$\Sigma' \land \boldsymbol{O}(\Sigma_0 \land \Box\beta) \land \Box\Omega\Sigma_{\text{def}} \ :$$

1. $\boldsymbol{B}\mathit{Fragile}(o)$;
2. $[drop(o)]\boldsymbol{B}\mathit{Broken}(o)$;
3. $[\mathit{senseF}(o)]\boldsymbol{K}\neg\mathit{Fragile}(o)$;
4. $[\mathit{senseF}(o)][drop(o)]\boldsymbol{K}\neg\mathit{Broken}(o)$.

(1) holds because of the default, since o's non-fragility is not yet known. Notice, in particular, the role of $\Omega\Sigma_{\text{def}}$: while the semantics of $\mathcal{ES}_{\mathrm{O}}$ puts no restrictions on δ other than $\delta(e) \subseteq e$,[8] it is $\Omega\Sigma_{\text{def}}$ which forces $\delta(e)$ to be the largest subset of e which is compatible with the default, that is, δ selects only worlds from e where o is fragile. (2) holds because the default also applies after $\mathit{drop}(o)$. In particular, Theorem 3 applies as $[drop(o)]\boldsymbol{O}(\mathit{Broken}(o) \equiv \mathit{Fragile}(o) \land \Box\beta)$ follows as well. Finally, in (3) and (4) the agent has found out that o is not fragile, blocking the default since $\models \Box(\boldsymbol{K}\alpha \supset \boldsymbol{B}\alpha)$.

As one of the reviewers remarked, from a commonsense point of view, it is also or perhaps more plausible to have a sensing action for broken instead of fragility. In other words, after dropping an object one would sense whether it is broken, and if not conclude that it must not be fragile. This can be modeled in our framework as well. In particular, in all situations which are compatible with sensing that the object o is not broken after dropping it the fluent $\mathit{Fragile}(o)$ is false.

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 (Thielscher 1999) and more distant cousins such as (Kowalski and Sergot 1986; Gelfond and Lifschitz 1993).

While $\mathcal{ES}_{\mathrm{O}}$ is intended to capture a fragment of the situation calculus, it is also related to the work formalizing action and change in the framework of dynamic logic (Harel 1984). Examples are (De Giacomo and Lenzerini 1995) and later (Herzig *et al* 2000), who also deal with belief. While these approaches remain propositional, there are also first-order treatments such as (Demolombe 2003; Demolombe, Herzig, and Varzinczak 2003; Blackburn *et al* 2001), which, like $\mathcal{ES}_{\mathrm{O}}$, are inspired by the desire to capture fragments of the situation calculus in modal logic. Demolombe (2003) even considers a form of only-knowing, which is related to the version of only-knowing in (Lakemeyer and Levesque 2004), which in turn derives from the logic $\mathcal{OL}$ (Levesque and Lakemeyer 2001).

The idea of progression is not new and lies at the heart of most planning systems, starting with STRIPS (Fikes and Nilsson 1971), but also in implemented agent programming languages like 3APL (Hindriks *et al* 1999). Lin and Reiter (1997) so far gave the most general account. Restricted forms of LR-progression, which are first-order definable, are

[7] Roughly, E is a stable expansion of α iff for all γ, $\gamma \in E$ iff γ is a first-order consequence of $\{\alpha\} \cup \{\boldsymbol{B}\beta \mid \beta \in E\} \cup \{\neg\boldsymbol{B}\beta \mid \beta \notin E\}$.

[8] Here e is the (unique) set of worlds which satisfies $\boldsymbol{O}(\Sigma_0 \land \Box\beta)$.

discussed in (Lin and Reiter 1997; Liu and Levesque 2005; Claßen *et al* 2007; Vassos and Levesque 2007).

Default reasoning has been applied to actions mostly to solve the frame problem (Shanahan 1993). Here, however, we use Reiter's monotonic solution to the frame problem (Reiter 1991) and we are concerned with the static "Tweety-flies" variety of defaults. Kakas et al. (2008) recently made a proposal that deals with these in the presence of actions, but only in a propositional setting of a language related to $\mathcal{A}$ (Gelfond and Lifschitz 1993).

Conclusion

The paper introduced a new semantics for the concept of only-knowing within a modal fragment of the situation calculus. In particular, we showed that, provided an agent starts with a basic action theory as its initial knowledge base, then all the agent knows after an action is again a basic action theory. The result is closely related to Lin and Reiter's notion of progression and generalizes it to allow for actions which return sensing results. We also showed how to handle static defaults in the sense that these are applied every time after an action has been performed. Because of the way only-knowing is modelled, defaults behave as in autoepistemic logic. In previous work we showed that by modifying the semantics of only-knowing in the static case, other forms of default reasoning like Reiter's default logic can be captured (Lakemeyer and Levesque 2006). We believe that these results will carry over to our dynamic setting as well.

References

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

Claßen, J.; Eyerich, P.; Lakemeyer, G.; and Nebel, B. 2007. Towards an integration of Golog and planning. In Veloso, M. M., ed., *Proc. of IJCAI-07,* 1846–1851.

Fikes, R., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

G. De Giacomo and M. Lenzerini, PDL-based framework for reasoning about actions, *Proc. of AI*IA'95,* LNAI 992, 103–114, 1995.

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

R. Demolombe, A. Herzig, and I.J. Varzinczak, Regression in modal logic, *J. of Applied Non-Classical Logics*, 13(2):165-185, 2003.

H. Enderton, *A Mathematical Introduction to Logic,* Academic Press, New York, 1972.

Michael Gelfond and Vladimir Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–321, 1993.

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

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

Hindriks, K. V.; De Boer, F. S.; Van der Hoek, W.; and Meyer, J.-J. C. 1999. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems* 2(4):357–401.

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

A. Kakas, L. Michael, and R. Miller, Fred meets Tweety. In *Proc. ECAI, IOS Press,* 747–748, 2008.

R. Kowalski and M. Sergot. A logic based calculus of events. *New Generation Computing,* 4:67–95, 1986.

G. Lakemeyer and H. J. Levesque, Situations, si! Situation Terms, no!. In *Ninth Conf. on Principles of Knowledge Representation and Reasoning,* AAAI Press, 2004.

G. Lakemeyer and H. J. Levesque, A useful fragment of the situation calculus. IJCAI-05, AAAI Press, 490–496, 2005.

G. Lakemeyer and H. J. Levesque, Towards an axiom system for default logic. In *Proc. of AAAI-06, AAAI Press,* 2006.

G. Lakemeyer and H. J. Levesque, A Semantical Account of Progression in the Presence of Defaults. in *Proc. IJCAI-09,* 2009.

G. Lakemeyer and H. J. Levesque, A Semantical Account of Progression in the Presence of Defaults (extended version). in *Conceptual Modeling: Foundations and Applications,* A. Borgida, V. Chaudhri, P. Giorgini, E. Yu (Eds.), Springer LNCS, 2009.

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

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

H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl., Golog: A logic programming language for dynamic domains. *Journal of Logic Programming,* **31**, 59–84, 1997.

F. Lin and R. Reiter, How to progress a database, *Artificial Intelligence*, Elsevier, 92, 131-167, 1997.

Y. Liu and H. J.Levesque, 2005. Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In *Proc. of IJCAI-05*.

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

R. C. Moore, Semantical considerations on nonmonotonic logic. *Artificial Intelligence* **25**, 75–94, 1985.

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

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

M. Shanahan. *Solving the Frame Problem.* MIT Press, 1997.

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

M. Thielscher. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence,* 111(1–2):277–299, 1999.

S. Vassos and H. J. Levesque, 2007. Progression of situation calculus action theories with incomplete information. In *Proc. IJCAI-07*.