Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

On the Progression of Knowledge in the Situation Calculus

Yongmei Liu
Dept. of Computer Science
Sun Yat-sen University
Guangzhou 510006, China
ymliu@mail.sysu.edu.cn

Ximing Wen[1,2]
[1]Dept. of Computer Science
Sun Yat-sen University
[2]Guangdong Institute of Public Administration
Guangzhou 510006, China
wenxim@student.sysu.edu.cn

Abstract

In a seminal paper, Lin and Reiter introduced the notion of progression for basic action theories in the situation calculus. Earlier works by Moore, Scherl and Levesque extended the situation calculus to account for knowledge. In this paper, we study progression of knowledge in the situation calculus. We first adapt the concept of bisimulation from modal logic and extend Lin and Reiter's notion of progression to accommodate knowledge. We show that for physical actions, progression of knowledge reduces to forgetting predicates in first-order modal logic. We identify a class of first-order modal formulas for which forgetting an atom is definable in first-order modal logic. This class of formulas goes beyond formulas without quantifying-in. We also identify a simple case where forgetting a predicate reduces to forgetting a finite number of atoms. Thus we are able to show that for local-effect physical actions, when the initial KB is a formula in this class, progression of knowledge is definable in first-order modal logic. Finally, we extend our results to the multi-agent case.

1 Introduction

In a seminal paper, Lin and Reiter [1997] introduced the notion of progression for basic action theories in the situation calculus. Roughly, progression concerns updating the representation of the current world state after an action is executed. Lin and Reiter showed that progression is not first-order (FO) definable in general, and identified two simple cases where progression is first-order definable and computable. They also conjectured that there is no alternative definition of progression that is always first-order definable. The conjecture was resolved by Vassos and Levesque [2008].

However, knowledge was not taken into account in Lin and Reiter's study of progression. To motivate progression of knowledge, consider the litmus testing example from [Moore, 1985]. Suppose there is an initial representation describing that the solution is acid and the agent knows the paper is not red. After the agent performs a litmus test on the solution, how should we update the representation to reflect the change to the world state and the agent's knowledge state? Intuitively, the new representation should entail that the agent knows the solution is acid. Moore [1985] adapted the possible-world model of knowledge to the situation calculus by introducing a special fluent $K(s', s)$, meaning that situation s' is accessible from situation s. Later, Scherl and Levesque [2003] proposed a solution to the frame problem for knowledge-producing actions by giving a successor state axiom for the K fluent.

Recently, Liu and Lakemeyer [2009] showed that for the so-called local-effect actions, progression is always first-order definable and computable. Their proof is a very simple one by applying Lin and Reiter's result that progression reduces to forgetting predicates in first-order logic. They identified a simple case where forgetting a predicate reduces to forgetting a finite number of atoms, and used it in their proof. Forgetting in first-order logic was studied by Lin and Reiter [1994]. In the past decades, forgetting has found many applications in the area of knowledge representation and reasoning. Recently, Zhang and Zhou [2009] studied forgetting in propositional S5 modal logic. They defined forgetting based on the notion of bisimulation in modal logic, and their definition coincides with the semantic definition of formula $\exists V \phi$ in [French, 2005] and the notion of uniform interpolation in [Ghilardi *et al.*, 2006]. A result of the latter paper illustrates that propositional S5 logic is closed under forgetting.

In this paper, we study progression of knowledge in the situation calculus. We first adapt the concept of bisimulation from modal logic and extend Lin and Reiter's definition of progression to accommodate knowledge. We show that for physical actions, progression of knowledge reduces to forgetting predicates in first-order modal logic. We study the problem of forgetting an atom or a predicate in first-order modal logic, and identify a class of first-order modal formulas for which forgetting an atom is definable in first-order modal logic, and forgetting a predicate is definable in second-order modal logic. This class of formulas goes beyond formulas without quantifying-in. We also identify a simple case where forgetting a predicate reduces to forgetting a finite number of atoms. Thus we are able to show that for local-effect physical actions, when the initial KB is a formula in this class, progression of knowledge is definable in first-order modal logic. As for sensing actions, we show how to do progression when the initial representation does not contain negative knowledge. Finally, we extend our results to the multi-agent case.

2 Preliminaries

2.1 First-order and second-order modal logic

The first-order modal language is obtained from the first-order language by adding a syntactic rule: if ϕ is a formula, then $K\phi$ (read as "knowing ϕ") is a formula, where K is a modal operator. We call $K\phi$ a knowledge atom.

There are variations in the definition of semantics for FO modal logic. *We take the constant domain variation in this paper*. A Kripke structure is a tuple $W = (S, R, D, \pi)$, where S is a set of states or possible worlds, R is a binary relation on S, called the accessibility relation, and π associates with each world s a first-order structure with D as the domain.

A variable assignment ν is a mapping from all variables to D. We let $\nu(x/d)$ denote the assignment that is the same as ν except that x is mapped to d. Given a Kripke structure W, a state s in W, a variable assignment ν, and a formula ϕ, $W, s, \nu \models \phi$ ("W, s, ν satisfies ϕ") is defined as follows:

1. $W, s, \nu \models P(\vec{\tau})$ iff $\pi(s), \nu \models P(\vec{\tau})$;
2. $W, s, \nu \models \phi \vee \psi$ iff $W, s, \nu \models \phi$ or $W, s, \nu \models \psi$;
3. $W, s, \nu \models \neg\phi$ iff $W, s, \nu \not\models \phi$;
4. $W, s, \nu \models \exists x\phi$ iff for some $d \in D$, $W, s, \nu(x/d) \models \phi$;
5. $W, s, \nu \models K\phi$ iff for all t such that sRt, $W, t, \nu \models \phi$.

A sentence ϕ is valid in W if $W, s \models \phi$ for all $s \in S$.

In modal logics, some commonly considered desired properties of knowledge are as follows:

A3 $K\phi \supset \phi$, the knowledge axiom;

A4 $K\phi \supset KK\phi$, the positive introspection axiom;

A5 $\neg K\phi \supset K\neg K\phi$, the negative introspection axiom.

These properties can be achieved by imposing various conditions on accessibility relations. For example, A3, A4, and A5 are valid in Kripke structures whose accessibility relations are reflexive, transitive, and Euclidean, respectively. A relation R is Euclidean if for any $(x, y) \in R$, $(x, y') \in R$, we must have $(y, y') \in R$. It is easy to show that a relation is an equivalence relation iff it is reflexive, transitive and Euclidean. *In this paper, we restrict our attention to structures whose accessibility relations are equivalence relations.* Such structures are called S5 structures in the literature.

We now extend FO modal logic to the second-order (SO) case. We add two syntactic rules, where P is a predicate variable: $P(\vec{\tau})$ is a formula; if ϕ is a formula, so is $\exists P\phi$. An n-ary *intension* wrt a Kripke structure $W = (S, R, D, \pi)$ is a function f mapping each $s \in S$ to an n-ary relation on D. Now a variable assignment ν maps each predicate variable to an intension, in addition to mapping each individual variable to an element of D. We add the following semantic rules:

1. $W, s, \nu \models P(\vec{\tau})$ iff $\pi(s), \nu' \models P(\vec{\tau})$, where ν' is the same as ν except that it maps each predicate variable Q to $\nu(Q)(s)$;
2. $W, s, \nu \models \exists P\phi$ iff for some intension I, $W, s, \nu(P/I) \models \phi$.

2.2 Situation calculus with knowledge

The situation calculus [Reiter, 2001] is a many-sorted first-order language suitable for describing dynamic worlds. There are three disjoint sorts: $action$ for actions, $situation$ for situations, and $object$ for everything else. A situation calculus language $\mathcal{L}_{sc}$ has the following components: a constant S_0 denoting the initial situation; a binary function $do(a, s)$ denoting the successor situation to s resulting from performing action a; a binary predicate $Poss(a, s)$ meaning that action a is possible in situation s; action functions, *e.g.*, $move(x, y)$; a finite number of relational fluents, *i.e.*, predicates taking a situation term as their last argument, *e.g.*, $ontable(x, s)$; and a finite number of situation-independent predicates and functions. We ignore functional fluents in this paper.

The situation calculus has been extended to accommodate sensing and knowledge. Assume that in addition to ordinary actions that change the world, there are sensing actions of the form $sense_{\psi}(\vec{x})$, which does not change the world but tells the agent whether some condition $\psi(\vec{x})$ holds in the current situation. Knowledge is modeled in the possible-world style by introducing a special fluent $K(s', s)$, meaning that situation s' is accessible from situation s. Then knowing ϕ at situation s is represented as follows:

$$\mathbf{Knows}(\phi, s) \stackrel{def}{=} \forall s'.K(s', s) \supset \phi[s'].$$

In the presence of sensing and knowledge, a domain of application is specified by a basic action theory of the form:

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{K}_{Init}, \text{ where}$$

1. Σ are the foundational axioms:
   (a) $do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2$
   (b) $(\neg s \sqsubset S_0) \wedge (s \sqsubset do(a, s') \equiv s \sqsubseteq s')$
   (c) $\forall P.\forall s[Init(s) \supset P(s)] \wedge$ $\forall a, s[P(s) \supset P(do(a, s))] \supset (\forall s)P(s)$, where $Init(s) \stackrel{def}{=} \neg(\exists a, s')s = do(a, s')$
   (d) $K(s, s') \supset [Init(s) \equiv Init(s')]$.

   A model of these axioms consists of a forest of isomorphic trees rooted at the initial situations, which can be K-related to only initial situations.
2. $\mathcal{D}_{ss}$ is a set of successor state axioms (SSAs) for fluents. The SSAs for ordinary fluents must satisfy the no-side-effect conditions, *i.e.*, they are not affected by sensing actions. The SSA for K is as follows:
   $$K(s', do(a, s)) \equiv \exists s^*.s' = do(a, s^*) \wedge K(s^*, s) \wedge$$
   $$\textstyle\bigwedge_i \forall \vec{x}_i[a = sense_{\psi_i}(\vec{x}_i) \supset \psi_i(\vec{x}_i, s^*) \equiv \psi_i(\vec{x}_i, s)].$$
   Intuitively, s' is accessible after a is done in s iff it is the result of doing a in some s^* which is accessible from s and which agrees with s on the formula being sensed.
3. $\mathcal{D}_{ap}$ is a set of action precondition axioms.
4. $\mathcal{D}_{una}$ is the set of unique names axioms for actions.
5. $\mathcal{K}_{Init}$ consists of axioms stating that K has certain properties in all initial situations. Such a property P must have the feature that, by virtue of the SSA for K, P holds in all situations as long as P holds in all initial situations. For example, the axiom for reflexivity is: $\forall s.Init(s) \supset K(s, s)$. *In this paper, we assume that $\mathcal{K}_{Init}$ consists of the axioms for the reflexive, transitive and Euclidean properties.*
6. $\mathcal{D}_{S_0}$, called the initial KB, is a set of first-order sentences uniform in S_0. We will make this more precise in Section 2.3.

2.3 Terminology and notation

Let $\mathcal{L}_{sc}$ be a situation calculus language. We use $\mathcal{L}$ to denote the situation-suppressed language of $\mathcal{L}_{sc}$, *i.e.*, the first-order language obtained from $\mathcal{L}_{sc}$ by removing the sort situation and the K fluent, and removing the situation argument from every ordinary fluent. We use $\mathcal{L}'$ to denote the primed version of $\mathcal{L}$, *i.e.*, for each ordinary fluent $F(\vec{x}, s)$, there is an $F'(\vec{x})$ predicate in $\mathcal{L}'$. We let $\mathcal{L}^*$ denote the union of $\mathcal{L}$ and $\mathcal{L}'$. Intuitively, we can use $\mathcal{L}$ to talk about a situation σ, and use $\mathcal{L}'$ to talk about a successor situation of σ. For a first-order language L, say $\mathcal{L}$, $\mathcal{L}'$, or $\mathcal{L}^*$, we use L_m to denote the first-order modal language based on L. For a language L, say $\mathcal{L}$ or $\mathcal{L}_m$, we let L^2 denote its second-order extension. By an L-formula we mean a formula of a language L, and by an L-structure we mean a structure of L.

Now let $\phi \in \mathcal{L}_m^2$, and σ a situation term. We define an $\mathcal{L}_{sc}^2$-formula $\phi[\sigma]$, which asserts that ϕ holds in σ.

Definition 2.1 $\phi[\sigma]$ is inductively defined as follows:

1. $P(\vec{\tau})[\sigma] = \begin{cases} P(\vec{\tau}, \sigma) & \text{if } P \text{ is a fluent} \\ P(\vec{\tau}) & \text{otherwise} \end{cases}$
2. $(\neg\phi)[\sigma] = \neg\phi[\sigma]$, $(\phi \vee \psi)[\sigma] = \phi[\sigma] \vee \psi[\sigma]$, $(\exists x\phi)[\sigma] = \exists x\phi[\sigma]$, $(\exists P\phi)[\sigma] = \exists P\phi[\sigma]$;
3. $(K\phi)[\sigma] = \forall s.K(s, \sigma) \supset \phi[s]$.

For example, let ϕ be the modal formula $KK\exists xF(x)$. Then $\phi[S_0]$ is $\forall s.K(s, S_0) \supset [\forall s'.K(s', s) \supset \exists xF(x, s)]$.

Similarly, we can define $\phi[\sigma]$ for $\phi \in \mathcal{L}_m'^2$, which asserts that the unprimed ϕ (the formula obtained from ϕ by replacing each F' with F) holds in σ.

Definition 2.2 We use $\mathcal{L}_\sigma$ ($\mathcal{L}_\sigma^2$ resp.) to denote the set of $\phi[\sigma]$ where $\phi \in \mathcal{L}_m$ ($\mathcal{L}_m^2$ resp.). A formula in $\mathcal{L}_\sigma$ or $\mathcal{L}_\sigma^2$ is said to be *uniform* in σ.

Let ϕ be a formula, μ and μ' two expressions. We denote by $\phi(\mu/\mu')$ the result of replacing every occurrence of μ in ϕ with μ'. For a structure M and a syntactic object o, we let o^M stand for the denotation of o in M. We use sit^M to denote the domain of an $\mathcal{L}_{sc}$-structure M for sort *situation*. Let α be a ground action. We denote by S_α the situation term $do(\alpha, S_0)$.

We say that a formula in $\mathcal{L}_m$ is *objective* if it does not contain any K operator. We introduce a notation $L\phi$ ("ϕ is possible"), which abbreviates for $\neg K\neg\phi$, and a notation $W\phi$ ("knowing whether ϕ"), which abbreviates for $K\phi \vee K\neg\phi$. We also use **KWhether**(ϕ, σ) to denote $W\phi[\sigma]$.

3 Progression of knowledge

In this section, we extend Lin and Reiter's definition of progression to accommodate knowledge.

Let $\mathcal{D}$ be a basic action theory. Intuitively, a progression of $\mathcal{D}_{S_0}$ wrt a ground action α should be a set of sentences $\mathcal{D}_{S_\alpha}$ with the following properties: First, just as $\mathcal{D}_{S_0}$ is uniform in S_0, $\mathcal{D}_{S_\alpha}$ should be uniform in S_α. Second, for all queries about the possible future of S_α, the old theory $\mathcal{D}$ is equivalent to the new theory $(\mathcal{D} - \mathcal{D}_{S_0}) \cup \mathcal{D}_{S_\alpha}$.

To define progression, we first adapt the concept of bisimulation from modal logic and define a similarity relation between $\mathcal{L}_{sc}$-structures. Roughly, a bisimulation is a relation between situations of two $\mathcal{L}_{sc}$-structures where related situations agree on all ordinary fluents and have matching accessibility possibilities.

Definition 3.1 Let M and M' be $\mathcal{L}_{sc}$-structures with the same domains for sorts *action* and *object*. Let $\gamma \in sit^M$ and $\gamma' \in sit^{M'}$. We write $M, \gamma \sim M', \gamma'$ if M and M' interpret all situation-independent predicate and function symbols identically, and there is a bisimulation relation $B \subseteq sit^M \times sit^{M'}$ such that $\gamma B \gamma'$, and whenever $\delta B \delta'$, we have:

1. $M, \delta \equiv M', \delta'$, denoting that M, δ and M', δ' agree on all ordinary fluents, that is, for every ordinary fluent F, and every variable assignment ν, $M, \nu(s/\delta) \models F(\vec{x}, s)$ iff $M', \nu(s/\delta') \models F(\vec{x}, s)$.
2. For all ρ s.t. $\delta K^M \rho$, there is ρ' s.t. $\delta' K^{M'} \rho'$ and $\rho B \rho'$ (the forth condition), here K^M is the denotation of the K fluent in M.
3. For all ρ' s.t. $\delta' K^{M'} \rho'$, there is ρ s.t. $\delta K^M \rho$ and $\rho B \rho'$ (the back condition).

We write $M \sim_{S_\alpha} M'$ if $M, S_\alpha^M \sim M', S_\alpha^{M'}$. Following the definition of progression in [Reiter, 2001], we have:

Definition 3.2 A set of sentences $\mathcal{D}_{S_\alpha}$ in $\mathcal{L}_{S_\alpha}^2$ is a progression of $\mathcal{D}_{S_0}$ wrt α if $\mathcal{D} \models \mathcal{D}_{S_\alpha}$, and for every model M of $(\mathcal{D} - \mathcal{D}_{S_0}) \cup \mathcal{D}_{S_\alpha}$, there is a model M' of $\mathcal{D}$ such that $M \sim_{S_\alpha} M'$.

By induction on the formula, it is easy to prove:

Proposition 3.3 *Let $M, \gamma \sim M', \gamma'$ and $\phi \in \mathcal{L}_m^2$. Then for any ν, $M, \nu(s/\gamma) \models \phi[s]$ iff $M', \nu(s/\gamma') \models \phi[s]$.*

Then it is straightforward to prove:

Theorem 3.4 *Let $\mathcal{D}_{S_\alpha}$ be a progression of $\mathcal{D}_{S_0}$ wrt α. Then for every $\phi \in \mathcal{L}_{S_\alpha}^2$, $\mathcal{D} \models \phi$ iff $(\mathcal{D} - \mathcal{D}_{S_0}) \cup \mathcal{D}_{S_\alpha} \models \phi$.*

Thus for any query about S_α, the old theory $\mathcal{D}$ and the new theory $(\mathcal{D} - \mathcal{D}_{S_0}) \cup \mathcal{D}_{S_\alpha}$ are equivalent.

4 Forgetting in first-order modal logic

In this section, we define forgetting in first-order modal logic, and analyze basic properties of forgetting. We identify a class of first-order modal formulas for which forgetting an atom is definable in first-order modal logic, and forgetting a predicate is definable in second-order modal logic. This class of formulas goes beyond formulas without quantifying-in.

We first review forgetting in first-order logic. We extend the definition to formulas with free variables.

Definition 4.1 Let $\phi(\vec{x}) \in \mathcal{L}$, and let μ be a ground atom or predicate. A formula $\psi(\vec{x})$ is a result of forgetting μ in $\phi(\vec{x})$, written forget$(\phi(\vec{x}), \mu) \Leftrightarrow \psi(\vec{x})$, if for any $\mathcal{L}$-structure U, and any variable assignment ν, $U, \nu \models \psi(\vec{x})$ iff there exists an $\mathcal{L}$-structure U' such that $U', \nu \models \phi(\vec{x})$ and $U \sim_\mu U'$, which denotes that U and U' agree on everything except possibly on the interpretation of μ.

A nice result about forgetting an atom in first-order logic is that it is always definable in first-order logic. As to forgetting a predicate, the result is definable in second-order logic. Let ϕ be a formula, $P(\vec{\tau})$ a ground atom, and v a truth value. We denote by $\phi^v_{P(\vec{\tau})}$ the result of replacing every occurrence of the form $P(\vec{\tau}')$ in ϕ by $\vec{\tau} = \vec{\tau}' \wedge v \vee \vec{\tau} \neq \vec{\tau}' \wedge P(\vec{\tau}')$.

Proposition 4.2

1. *forget*$(\phi, P(\vec{\tau})) \Leftrightarrow \phi^{true}_{P(\vec{\tau})} \vee \phi^{false}_{P(\vec{\tau})}$;
2. *forget*$(\phi, P) \Leftrightarrow \exists R.\phi(P/R)$, *where R is a second-order predicate variable.*

To define forgetting in first-order modal logic, we first define a similarity relation between pairs of the form (W, s) where W is a Kripke structure and s a state in M.

Definition 4.3 Let μ be a ground atom or predicate. Let $W = (S, R, D, \pi)$ and $W' = (S', R', D, \pi')$ be Kripke structures with the same domain. Let $s \in S$ and $s' \in S'$. We write $W, s \sim_\mu W', s'$ if there exists a bisimulation relation $B \subseteq S \times S'$ such that sBs', and whenever tBt', we have:

1. $\pi(t) \sim_\mu \pi'(t')$;
2. For all u s.t. tRu, there exists u' s.t. $t'R'u'$ and uBu';
3. For all u' s.t. $t'R'u'$, there exists u s.t. tRu and uBu'.

We define forgetting as follows:

Definition 4.4 Let $\phi(\vec{x}) \in \mathcal{L}_m$, and let μ be a ground atom or predicate. A formula $\psi(\vec{x})$ is a result of forgetting μ in $\phi(\vec{x})$, written kforget$(\phi(\vec{x}), \mu) \Leftrightarrow \psi(\vec{x})$, if for any S5 pair (W, s), and any ν, $W, s, \nu \models \psi(\vec{x})$ iff there exists an S5 pair (W', s') such that $W', s', \nu \models \phi(\vec{x})$ and $W, s \sim_\mu W', s'$.

Next, we analyze basic properties of forgetting. We say a formula ϕ *irrelevant* to a predicate P if ϕ is equivalent to a formula which does not use P. We say ϕ irrelevant to a ground atom $P(\vec{\tau})$ if ϕ is equivalent to a formula ψ where any appearance of $P(\vec{\tau}')$ must be in the form of $P(\vec{\tau}') \wedge \vec{\tau}' \neq \vec{\tau}$.

Proposition 4.5 *Let ϕ be a sentence and kforget$(\phi, \mu) \Leftrightarrow \psi$. Then $\phi \models \psi$; and for any η irrelevant to μ, $\phi \models \eta$ iff $\psi \models \eta$.*

Proposition 4.6 *1. If ϕ is an objective formula, then kforget$(\phi, \mu) \Leftrightarrow$ forget(ϕ, μ);*

2. *If kforget$(\phi_i, \mu) \Leftrightarrow \psi_i$, $i = 1, 2$, then kforget$(\phi_1 \vee \phi_2, \mu) \Leftrightarrow \psi_1 \vee \psi_2$;*
3. *If kforget$(\phi(\vec{x}, y), \mu) \Leftrightarrow \psi(\vec{x}, y)$, then kforget$(\exists y \phi(\vec{x}, y), \mu) \Leftrightarrow \exists y \psi(\vec{x}, y)$.*

In general, we do not have: kforget$(\phi_1 \wedge \phi_2, \mu) \Leftrightarrow \psi_1 \wedge \psi_2$, where kforget$(\phi_i, \mu) \Leftrightarrow \psi_i$, $i = 1, 2$. But we can identify a form of conjunctive formulas for which forgetting reduces to forgetting for component formulas. Clearly, we have $K\phi_1 \wedge K\phi_2 \Leftrightarrow K(\phi_1 \wedge \phi_2)$, but not $L\phi_1 \wedge L\phi_2 \Leftrightarrow L(\phi_1 \wedge \phi_2)$. An *extended term* is a formula of the form $\phi \wedge K\psi \wedge \bigwedge_i \forall \vec{x}_i L\eta_i$, where ϕ, ψ, and η_i's are all objective. The following proposition shows that extended terms are closed under forgetting.

Proposition 4.7 *Let ξ be an extended term $\phi \wedge K\psi \wedge \bigwedge_i \forall \vec{x}_i L\eta_i$. Let forget$(\phi \wedge \psi, \mu) \Leftrightarrow \phi'$, forget$(\psi, \mu) \Leftrightarrow \psi'$, forget$(\psi \wedge \eta_i, \mu) \Leftrightarrow \eta'_i$. Then kforget$(\xi, \mu) \Leftrightarrow \phi' \wedge K\psi' \wedge \bigwedge_i \forall \vec{x}_i L\eta'_i$.*

Proof sketch: Let ζ denote $\phi' \wedge K\psi' \wedge \bigwedge_i \forall \vec{x}_i L\eta'_i$. We show that for any S5 pair (W, s) and any ν, $W, s, \nu \models \zeta$ iff there is an S5 pair (W', s') s.t. $W', s', \nu \models \xi$ and $W, s \sim_\mu W', s'$. We only show the only-if direction here. So suppose $W, s, \nu \models \zeta$. Let $W = (S, R, D, \pi)$. Then $\pi(s), \nu \models \phi'$; for any t s.t. sRt, $\pi(t), \nu \models \psi'$; and for all i, for all $\vec{e}_i \in D$, there exists $\tau(\vec{e}_i)$ s.t. $sR\tau(\vec{e}_i)$ and $\pi(\tau(\vec{e}_i)), \nu(\vec{x}_i/\vec{e}_i) \models \eta'_i$. We define $W' = (S', R', D, \pi')$ and the bisimulation B as follows: there is a state s' in S', sBs', $\pi(s) \sim_\mu \pi'(s')$, and $\pi'(s'), \nu \models \phi \wedge \psi$; for any i and any $\vec{e}_i \in D$, there is a state $\tau'(\vec{e}_i)$ in S', $\tau(\vec{e}_i) B \tau'(\vec{e}_i)$, $\pi(\tau(\vec{e}_i)) \sim_\mu \pi'(\tau'(\vec{e}_i))$, and $\pi'(\tau'(\vec{e}_i)), \nu \models \psi \wedge \eta_i$; for any t s.t. sRt, if there has not been a state t' in S' s.t. tBt', add a state t' in S', let tBt', $\pi(t) \sim_\mu \pi'(t')$, and $\pi'(t'), \nu \models \psi$. Let $R' = S' \times S'$. Then $W, s \sim_\mu W', s'$ and $W', s', \nu \models \xi$. ∎

We now define a *normal form* for $\mathcal{L}_m$-formulas:

Definition 4.8 We say that an $\mathcal{L}_m$-formula ϕ is in $\exists$-disjunctive normal form ($\exists$-DNF) if it is of the form $\exists \vec{x}(\phi_1 \vee \ldots \vee \phi_n)$, where each ϕ_i is an extended term.

By Propositions 4.2, 4.6, and 4.7, we have:

Theorem 4.9 *Let ϕ be in $\exists$-DNF. Then the result of forgetting an atom in ϕ is definable in first-order modal logic, and the result of forgetting a predicate in ϕ is definable in second-order modal logic without second-order quantifying-in.*

By second-order quantifying-in, we mean occurrence of SO quantifiers outside the K operators. To shed light on the expressiveness of $\exists$-DNF, we note two results. First, due to the constant domain semantics, we have $\forall x K\phi(x) \Leftrightarrow K\forall x\phi(x)$ and $\exists x L\phi(x) \Leftrightarrow L\exists x\phi(x)$. Also, we have

Theorem 4.10 *Let ϕ be a formula without quantifying-in, i.e., no quantifiers occur outside the K operators. Then ϕ can be transformed in S5 to an equivalent formula without nesting of the K operators.*

The proof is the same as that for propositional S5, which can be found in, for example, [Hughes and Cresswell, 1996]. The resulting formula can be further transformed into a form of DNF whose atoms are objective formulas and knowledge atoms. Such DNF is a special case of $\exists$-DNF.

Thus the class of formulas, for which forgetting an atom is definable in FO modal logic and forgetting a predicate is definable in SO modal logic, includes the following, where ϕ is objective: formulas without quantifying-in; $\forall x K\phi$, $\exists x K\phi$, $\forall x L\phi$, $\exists x L\phi$; and $\exists x \forall y L\phi$, $\forall x \exists y L\phi$, $\exists x \forall y K\phi$.

Example 4.1 Consider a simple blocks world. There are two predicates: $clear(x)$: block x has no blocks on top of it, and $on(x, y)$: block x is on block y. Let ϕ be the conjunction of the following sentences:

1. $\phi_1 : \forall x(x \neq A \wedge x \neq B \wedge x \neq C \supset \neg clear(x))$,
2. $K\phi_2$, where ϕ_2 is $\forall x \forall y(on(x, y) \supset \neg clear(y))$,
3. $\phi_3 : \forall u \forall v L\, on(u, v) \wedge \forall u \forall v L\, \neg on(u, v)$, stating that the agent has no knowledge about the *on* predicate,
4. $K\neg clear(A) \vee K\neg clear(B)$,
5. $\exists z(z \neq A \wedge z \neq B \wedge K\neg clear(z))$, stating that the agent knows that some block other than A and B is not clear.

We will compute kforget(ϕ, on) and kforget($\phi, clear(C)$). To save space, in the sequel, we will use "$c(x)$" for "$clear(x)$".

We first convert ϕ into $\exists$-DNF $\exists z(\psi_1 \vee \psi_2)$, where ψ_1 is $\phi_1 \wedge z \neq A \wedge z \neq B \wedge K(\phi_2 \wedge \neg c(z) \wedge \neg c(A)) \wedge \phi_3$, and ψ_2 is $\phi_1 \wedge z \neq A \wedge z \neq B \wedge K(\phi_2 \wedge \neg c(z) \wedge \neg c(B)) \wedge \phi_3$.

Then kforget(ϕ, on) is $\exists z(\eta_1 \vee \eta_2)$, where η_1 is
$\exists R[\phi_1 \wedge z \neq A \wedge z \neq B \wedge \forall x \forall y(R(x,y) \supset \neg c(y)) \wedge$
$\quad \neg c(z) \wedge \neg c(A)] \wedge$
$K \exists R[\forall x \forall y(R(x,y) \supset \neg c(y)) \wedge \neg c(z) \wedge \neg c(A)] \wedge$
$\forall u \forall v L \exists R[R(u,v) \wedge \forall x \forall y(R(x,y) \supset \neg c(y)) \wedge$
$\quad \neg c(z) \wedge \neg c(A)] \wedge$
$\forall u \forall v L \exists R[\neg R(u,v) \wedge \forall x \forall y(R(x,y) \supset \neg c(y)) \wedge$
$\quad \neg c(z) \wedge \neg c(A)]$, and η_2 is the same as η_1 except that $\neg c(A)$ is replaced with $\neg c(B)$.

Finally, kforget($\phi, clear(C)$) is $\exists z(\xi_1 \vee \xi_2)$, where ξ_1 is
$\{[\phi_1 \wedge z \neq A \wedge z \neq B \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{true}_{c(C)} \vee$
$\quad [\phi_1 \wedge z \neq A \wedge z \neq B \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{false}_{c(C)}\} \wedge$
$K\{[\phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{true}_{c(C)} \vee [\phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{false}_{c(C)}\} \wedge$
$\forall u \forall v L\{[on(u,v) \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{true}_{c(C)} \vee$
$\quad [on(u,v) \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{false}_{c(C)}\}$
$\forall u \forall v L\{[\neg on(u,v) \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{true}_{c(C)} \vee$
$\quad [\neg on(u,v) \wedge \phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{false}_{c(C)}\}$, and ξ_2 is the same as ξ_1 except that $\neg c(A)$ is replaced with $\neg c(B)$.

For example, $[\phi_2 \wedge \neg c(z) \wedge \neg c(A)]^{false}_{c(C)}$ is
$\forall x \forall y(on(x,y) \supset y = C \vee \neg clear(y)) \wedge$
$(z = C \vee \neg c(z)) \wedge (A = C \vee \neg c(A))$.

5 Representing progression of knowledge

In this section, we prove that for physical actions, progression of knowledge reduces to forgetting predicates in first-order modal logic. Further, we show that for local-effect physical actions, when the initial KB is in $\exists$-DNF, progression of knowledge is definable in first-order modal logic. As for sensing actions, we show how to do progression when the initial KB does not contain negative knowledge.

5.1 Progression wrt physical actions

The SSA for any ordinary fluent F is in the form of $F(\vec{x}, do(a,s)) \equiv \Phi_F(\vec{x}, a)[s]$, where $\Phi_F(\vec{x}, a) \in \mathcal{L}$. Let α be a physical action. We let $\mathcal{D}_{ss}[\alpha, S_0]$ denote the instantiation of $\mathcal{D}_{ss}$ wrt α and S_0, *i.e.*, the set of sentences $F(\vec{x}, S_\alpha) \equiv \Phi_F(\vec{x}, \alpha)[S_0]$. We use $\mathcal{D}^*_{ss}[\alpha, S_0]$ to denote the set of sentences $F'(\vec{x}) \equiv \Phi_F(\vec{x}, \alpha)$.

The following theorem shows that for physical actions, progression of knowledge reduces to forgetting all the F predicates in an $\mathcal{L}^*_m$-formula.

Theorem 5.1 *Let $\mathcal{D}_{S_0}$ be $\phi[S_0]$ where $\phi \in \mathcal{L}_m$, and α a physical action. Let kforget($\phi \wedge K\mathcal{D}^*_{ss}[\alpha, S_0], \vec{F}$) $\Leftrightarrow \psi$, where $\psi \in \mathcal{L}'^2_m$. Then $\psi[S_\alpha]$ is a progression of $\mathcal{D}_{S_0}$ wrt α.*

Note about the K operator in front of $\mathcal{D}^*_{ss}[\alpha, S_0]$ in the theorem. As pointed out in [Reiter, 2001], a consequence of the way knowledge is modeled in the situation calculus is that an agent knows the successor state axioms of her actions.

To prove the theorem, we now introduce a method to induce a Kripke structure from a model of $\mathcal{D} - \mathcal{D}_{S_0}$. Note that the instantiation of the SSA for K wrt a physical action α is:

$$K(s', do(\alpha, s)) \equiv \exists s^*.s' = do(\alpha, s^*) \wedge K(s^*, s).$$

So the K relation on situations resulting from doing α in initial situations is a copy of the K relation on initial situations.

Definition 5.2 Let $M \models \mathcal{D} - \mathcal{D}_{S_0}$. We define a Kripke structure $W_\alpha = (S, R, D, \pi)$ as follows. S is the set of initial situations of M. R is the restriction of K^M to S. D is the union of M's domains for sorts action and object. Let $\epsilon \in S$. $\pi(\epsilon)$ interprets any situation-independent predicate and function symbol as M does. For each ordinary fluent $F(\vec{x}, s)$, $\pi(\epsilon)$ interprets $F(\vec{x})$ as M interprets $F(\vec{x}, s)$ at ϵ, and interprets $F'(\vec{x})$ as M interprets $F(\vec{x}, s)$ at $do^M(\alpha^M, \epsilon)$.

Proposition 5.3 *Let $M \models \mathcal{D} - \mathcal{D}_{S_0}$. Then*

1. *$W_\alpha, S_0^M \models K\mathcal{D}^*_{ss}[\alpha, S_0]$;*
2. *for any $\phi \in \mathcal{L}^2_m$, $M \models \phi[S_0]$ iff $W_\alpha, S_0^M \models \phi$;*
3. *for any $\phi \in \mathcal{L}'^2_m$, $M \models \phi[S_\alpha]$ iff $W_\alpha, S_0^M \models \phi$.*

We now prove the theorem:

Proof sketch: We first prove $\mathcal{D} \models \psi[S_\alpha]$. So let $M \models \mathcal{D}$. By Proposition 5.3, $W_\alpha, S_0^M \models \phi \wedge K\mathcal{D}^*_{ss}[\alpha, S_0]$, which, by Proposition 4.5, implies ψ. Thus $W_\alpha, S_0^M \models \psi$. So $M \models \psi[S_\alpha]$. Now we prove for any M, if $M \models (\mathcal{D} - \mathcal{D}_{S_0}) \cup \{\psi[S_\alpha]\}$, then there exists $M' \models \mathcal{D}$ s.t. $M \sim_{S_\alpha} M'$. So let $M \models (\mathcal{D} - \mathcal{D}_{S_0}) \cup \{\psi[S_\alpha]\}$. Then $W_\alpha, S_0^M \models \psi$. Thus there exists an S5 pair (W^*, s^*) s.t. $W^*, s^* \models \phi \wedge K\mathcal{D}^*_{ss}[\alpha, S_0]$ and $W_\alpha, S_0^M \sim_{\vec{F}} W^*, s^*$. We construct M' as follows: M and M' have the same domains for sorts action and object, and interpret any situation-independent predicate and function identically. The initial situations of M' are the states of M^* accessible from s^*, M' interprets S_0 as s^*, and the situations of M' form a forest rooted at the initial situations. All the initial situations of M' are K-related, and for each initial situation ϵ of M', M' interprets any ordinary fluent F at ϵ as M^* does at ϵ. Finally, $M' \models \mathcal{D}_{ss} \cup \mathcal{D}_{ap}$. ∎

By Theorems 4.9 and 5.1, we have:

Corollary 5.4 *Let $\mathcal{D}_{S_0}$ be in $\exists$-DNF, and α a physical action. Then progression of $\mathcal{D}_{S_0}$ wrt α is definable in second-order modal logic without second-order quantifying-in.*

5.2 Progression wrt local-effect physical actions

We first present an intuitive result concerning forgetting a predicate in first-order modal logic: if a sentence ϕ entails knowing that the truth values of two predicates P and Q are different at only a finite number of certain instances, then forgetting Q in ϕ can be obtained from forgetting the Q atoms of these instances in ϕ and then replacing Q by P in the result.

Let $\vec{x}$ be a vector of variables, and let $\Delta = \{\vec{\tau}_1, \ldots, \vec{\tau}_m\}$ be a set of vectors of ground terms, where all the vectors have the same length. We use $\vec{x} \in \Delta$ to represent the formula $\vec{x} = \vec{\tau}_1 \vee \ldots \vee \vec{x} = \vec{\tau}_m$. Let P and Q be two predicates. We let $Q(\Delta)$ denote the set $\{Q(\vec{\tau}) \mid \vec{\tau} \in \Delta\}$, and we let $P \approx_\Delta Q$ represent the sentence $\forall \vec{x}.\vec{x} \notin \Delta \supset P(\vec{x}) \equiv Q(\vec{x})$.

Proposition 5.5 *Let kforget($\phi, Q(\Delta)$) $\Leftrightarrow \psi$, and $M, s \models K(P \approx_\Delta Q)$. Then $M, s \models \psi$ iff $M, s \models \psi(Q/P)$.*

Theorem 5.6 *Let P and Q be two predicates, and Δ a finite set of vectors of ground terms. If kforget$(\phi, Q(\Delta)) \Leftrightarrow \psi$, then kforget$(\phi \wedge K(P \approx_\Delta Q), Q) \Leftrightarrow \psi(Q/P)$.*

Actions in many dynamic domains have only local effects in the sense that if an action $A(\vec{c})$ changes the truth value of an atom $F(\vec{a}, s)$, then $\vec{a}$ is contained in $\vec{c}$. This contrasts with actions having non-local effects such as moving a briefcase, which will also move all the objects inside the briefcase without having mentioned them. For a formal definition of local-effect action theory, see [Liu and Lakemeyer, 2009].

Now let $\mathcal{D}$ be a local-effect action theory. Let $\alpha = A(\vec{c})$ be a physical action. We use Ω to denote the set of $F(\vec{a})$ where F is an ordinary fluent, and $\vec{a}$ is contained in $\vec{c}$. Then α only changes the truth values of atoms from Ω. Recall that $\mathcal{D}^*_{ss}[\alpha, S_0]$ denotes the set of sentences $F'(\vec{x}) \equiv \Phi_F(\vec{x}, \alpha)$. We let $\mathcal{D}^*_{ss}[\Omega]$ denote the set of sentences $F'(\vec{a}) \equiv \Phi_F(\vec{a}, \alpha)$, where $F(\vec{a}) \in \Omega$. Then $\mathcal{D}^*_{ss}[\alpha, S_0]$ is logically equivalent to $\mathcal{D}^*_{ss}[\Omega]$ together with the set of sentences $F' \approx_\Delta F$, where $\Delta = \{\vec{a} \mid \vec{a} \text{ is contained in } \vec{c}\}$.

By Theorems 4.9, 5.1, and 5.6, we get:

Theorem 5.7 *Let $\mathcal{D}$ be local-effect, and α a physical action. Let $\mathcal{D}_{S_0}$ be $\phi[S_0]$ where ϕ is in $\exists$-DNF. Then there is $\psi \in \mathcal{L}^*_m$ such that kforget$(\phi \wedge K\mathcal{D}^*_{ss}[\Omega], \Omega) \Leftrightarrow \psi$, and $\psi(\vec{F}/\vec{F}')[S_\alpha]$ is a progression of $\mathcal{D}_{S_0}$ wrt α.*

So for local-effect actions, when $\mathcal{D}_{S_0}$ is in $\exists$-DNF, progression of knowledge is definable in first-order modal logic, and can be computed by forgetting in $\phi \wedge K\mathcal{D}^*_{ss}[\Omega]$ atoms from Ω and then replacing each F by F' in the result.

Example 5.1 Consider the litmus testing example. There are two actions: $test$, insert litmus paper into solution, and $sense_{red}$, sense whether the paper turns red.
$\mathcal{D}_{S_0} = \{acid(S_0), \textbf{Knows}(\neg red, S_0)\}$. $\mathcal{D}_{ss}$ includes:
$red(do(a, s)) \equiv a = test \wedge acid(s) \vee red(s)$,
$acid(do(a, s)) \equiv acid(s) \wedge a \neq dilute$.

Here $test$ is a local-effect action: it only changes the truth value of red. Let $S_1 = do(test, S_0)$. $\mathcal{D}_{ss}[red]$ is $red(S_1) \equiv acid(S_0) \vee red(S_0)$. By Theorem 5.7, to compute $\mathcal{D}_{S_1}$, the progression of $\mathcal{D}_{S_0}$ wrt $test$, we compute kforget(ϕ, red), where $\phi = acid \wedge K\neg red \wedge K(red' \equiv acid \vee red)$, which is equivalent to $acid \wedge K(\neg red \wedge (red' \equiv acid))$. Since forget$(acid \wedge \neg red \wedge (red' \equiv acid), red) \Leftrightarrow acid \wedge red'$, and forget$(\neg red \wedge (red' \equiv acid), red) \Leftrightarrow (red' \equiv acid)$, by Proposition 4.7, we have kforget$(\phi, red) \Leftrightarrow acid \wedge red' \wedge K(red' \equiv acid)$. Thus $\mathcal{D}_{S_1} = \{acid(S_1), red(S_1), \textbf{Knows}(red \equiv acid, S_1)\}$.

5.3 Progression wrt sensing actions

We say that a formula does not contain negative knowledge if no knowledge atom appears within the scope of an odd number of negation operators. The following theorem shows when the initial KB does not contain negative knowledge, progression wrt sensing actions can be achieved by simply adding knowing whether the sensed formula holds.

Theorem 5.8 *Assume that $\mathcal{D}_{S_0}$ does not contain negative knowledge. Let $\psi(\vec{x})$ be an objective formula. Let α be the sensing action $sense_\psi(\vec{c})$. Then the following is a progression of $\mathcal{D}_{S_0}$ wrt α: $\mathcal{D}_{S_0}(S_0/S_\alpha) \cup \{W\psi(\vec{c})[S_\alpha]\}$.*

Proof sketch: We first prove $\mathcal{D} \models \mathcal{D}_{S_0}(S_0/S_\alpha) \cup \{W\psi(\vec{c})[S_\alpha]\}$. So let $M \models \mathcal{D}$. Since $\mathcal{D}_{S_0}$ does not contain negative knowledge and $M \models \mathcal{D}_{S_0}$, we get $M \models \mathcal{D}_{S_0}(S_0/S_\alpha)$. We can prove that if $M \models \psi(\vec{c}, S_0)$, then $M \models \textbf{Knows}(\psi(\vec{c}), S_\alpha)$, otherwise $M \models \textbf{Knows}(\neg\psi(\vec{c}), S_\alpha)$. Now let $M \models (\mathcal{D} - \mathcal{D}_{S_0}) \cup \mathcal{D}_{S_0}(S_0/S_\alpha) \cup \{W\psi(\vec{c})[S_\alpha]\}$. We prove that there exists $M' \models \mathcal{D}$ s.t. $M \sim_{S_\alpha} M'$. We construct M' as follows: The initial situations of M' is the set of situations K-related to S^M_α, and M' interprets S_0 as S^M_α. For each initial situation ϵ of M', M' interprets any ordinary fluent F at ϵ as M does at ϵ. The rest of the construction is the same as that in the proof of Theorem 5.1. Note that we need $M \models W\psi(\vec{c})[S_\alpha]$ to prove $M \sim_{S_\alpha} M'$. ■

Example 5.1 continued. Let $S_2 = do(sense_{red}, S_1)$. Since $\mathcal{D}_{S_1}$ does not contain negative knowledge, by Theorem 5.8, the progression of $\mathcal{D}_{S_1}$ wrt $sense_{red}$ is $\{acid(S_2), red(S_2), \textbf{Knows}(red \equiv acid, S_2), \textbf{KWhether}(red, S_2)\}$, which is equivalent to $\{\textbf{Knows}(red, S_2), \textbf{Knows}(acid, S_2)\}$.

6 Extension to the multi-agent case

In this section, we extend our results on forgetting and progression to the multi-agent case.

The multi-agent first-order modal language is the same as the single-agent one except that there is a modal operator K_i for each agent i, and there is an extra modal operator C. If ϕ is a formula, so are $K_i\phi$ ("agent i knows ϕ") and $C\phi$ ("ϕ is common knowledge"). A Kripke structure for n agents is the same as before except that there is an accessibility relation R_i for each agent i. The semantics is defined as follows:

1. $M, s, \nu \models K_i\phi$ iff for all t such that sR_it, $M, t, \nu \models \phi$;
2. $M, s, \nu \models C\phi$ iff $M, t, \nu \models \phi$ for all t that are reachable from s by the the relation R, the union of all R_i's.

The definition of forgetting is the same as before except: when defining the similarity relation between Kripke structures, we require the forth and back conditions hold for each agent i; and when defining forgetting, we consider S5 structures, *i.e.*, structures where each R_i is an equivalence relation.

Clearly, we have $C\phi_1 \wedge C\phi_2 \Leftrightarrow C(\phi_1 \wedge \phi_2)$. We now define $\exists$-DNF in the multi-agent case. We say that a formula ϕ is *purely objective* if it does not contain any modal operator. We say that ϕ is *objective* wrt agent i if it does not use any K_i operator. We inductively define extended terms as follows:

1. A purely objective formula is an extended term;
2. A formula of form $\phi \wedge C\varphi \wedge \bigwedge_i(K_i\psi_i \wedge \bigwedge_j \forall\vec{x}_{ij}L_i\eta_{ij})$ is an extended term if ϕ, φ and all ψ_i's are purely objective, and all η_{ij}'s are extended terms objective wrt agent i.

The following proposition shows that extended terms are closed under forgetting.

Proposition 6.1
Let ξ be an extended term $\phi \wedge C\varphi \wedge \bigwedge_i(K_i\psi_i \wedge \bigwedge_j \forall\vec{x}_{ij}L_i\eta_{ij})$. Let forget$(\phi \wedge \varphi \wedge \bigwedge_i \psi_i, \mu) \Leftrightarrow \phi'$, forget$(\varphi, \mu) \Leftrightarrow \varphi'$, forget$(\varphi \wedge \psi_i, \mu) \Leftrightarrow \psi_i'$, kforget$(\psi_i \wedge C\varphi \wedge \eta_{ij}, \mu) \Leftrightarrow \eta_{ij}'$. Let ζ be $\phi' \wedge C\varphi' \wedge \bigwedge_i(K_i\psi_i' \wedge \bigwedge_j \forall\vec{x}_{ij}L_i\eta_{ij}')$. Then kforget$(\xi, \mu) \Leftrightarrow \zeta$.

Proof sketch: The proof is similar to that of Proposition 4.7. Let $W, s, \nu \models \zeta$. Then for all i and j, for all $\vec{e}_{ij} \in D$, there is $\tau(\vec{e}_{ij})$ s.t. $sR_i\tau(\vec{e}_{ij})$ and $W, \tau(\vec{e}_{ij}), \nu(\vec{x}_{ij}/\vec{e}_{ij}) \models \eta'_{ij}$. Thus there exists $W(\vec{e}_{ij}), \tau'(\vec{e}_{ij}) \sim_\mu W, \tau(\vec{e}_{ij})$ such that $W(\vec{e}_{ij}), \tau'(\vec{e}_{ij}), \nu(\vec{x}_{ij}/\vec{e}_{ij}) \models \psi_i \wedge C\varphi \wedge \eta_{ij}$. When constructing W', we include $W(\vec{e}_{ij})$ and add K_i edges between $\tau'(\vec{e}_{ij})$ and other states of W'. Since η_{ij} is objective wrt agent i, we get $W', \tau'(\vec{e}_{ij}), \nu(\vec{x}_{ij}/\vec{e}_{ij}) \models \eta_{ij}$. ∎

The definition of ∃-DNF is as before, and Theorem 4.9 also holds in the multi-agent case, *i.e.*, for an ∃-DNF formula, forgetting an atom is definable in FO modal logic, and forgetting a predicate is definable in SO modal logic without SO quantifying-in.

We now consider progression in the multi-agent case. Each agent i can perform a number of sensing actions: $sense_{i,\psi_{ij}}(\vec{x}_{ij})$, $j = 1, \ldots, m$. For each agent i, there is a special fluent $K_i(s', s)$. The SSA for K_i is as follows:

$$K_i(s', do(a, s)) \equiv \exists s^*.s' = do(a, s^*) \wedge K_i(s^*, s) \wedge$$
$$\bigwedge_j \forall \vec{x}_{ij}[a = sense_{i,\psi_{ij}}(\vec{x}_{ij}) \supset \psi_{ij}(\vec{x}_{ij}, s^*) \equiv \psi_{ij}(\vec{x}_{ij}, s).]$$

We assume that all agents can perform the same kinds of actions. Under this assumption, a consequence of the way knowledge is modeled in the situation calculus is that the agents commonly knows the successor state axioms of physical actions. We consider the case of public actions whose occurrence is common knowledge. For example, when two agents cooperate in the blocks world, the physical actions are public. When two agents play a card game, a public sensing action of an agent is the action of picking up and reading a card dealt to her. We have the following results on progression, which are the same as in the single-agent case except that the K operator is replaced with the C operator. The first result shows that after an agent publicly performs a physical action, its effect becomes common knowledge. The second result shows that after agent i publicly senses a formula ψ, it becomes common knowledge that agent i knows if ψ holds.

Theorem 6.2 *Let $\mathcal{D}_{S_0}$ be $\phi[S_0]$ where $\phi \in \mathcal{L}_m$, and α a public physical action. Let kforget$(\phi \wedge C\mathcal{D}^*_{ss}[\alpha, S_0], \vec{F}) \Leftrightarrow \psi$, where $\psi \in \mathcal{L}'^2_m$. Then $\psi[S_\alpha]$ is a progression of $\mathcal{D}_{S_0}$ wrt α.*

Theorem 6.3 *Assume that $\mathcal{D}_{S_0}$ does not contain negative knowledge. Let $\psi(\vec{x})$ be an objective formula, and α a public sensing action $sense_{i,\psi}(\vec{c})$. Then the following is a progression of $\mathcal{D}_{S_0}$ wrt α: $\mathcal{D}_{S_0}(S_0/S_\alpha) \cup \{CW_i\psi(\vec{c})[S_\alpha]\}$, where $W_i\psi$ abbreviates for $K_i\psi \vee K_i\neg\psi$.*

Example 6.1 There is a red card and a green card. A dealer gives a card each to Ann and Bob. We use a fluent $red(i, s)$, meaning that agent i has the red card in situation s. We use a sensing action $sense_{i,red}(x)$, meaning that agent i picks up and reads his/her card. Suppose that the dealer has dealt the red card to Ann, and green card to Bob. Then
$D_{S_0} = \{red(a, S_0), \neg red(b, S_0), C(red(a) \wedge \neg red(b) \vee \neg red(a) \wedge red(b))[S_0]\}$. Let $S_1 = do(sense_{a,red}(a), S_0)$.
Then $\mathcal{D}_{S_1} = \mathcal{D}_{S_0}(S_0/S_1) \cup \{CW_a red(a)[S_1]\}$,
which entails $K_a red(a)[S_1]$, $K_b W_a red(a)[S_1]$, and $K_b W_a red(b)[S_1]$, but does not entail $K_b red(a)[S_1]$.
Let $S_2 = do(sense_{b,red}(b), S_1)$. Then
$\mathcal{D}_{S_2} = \mathcal{D}_{S_1}(S_1/S_2) \cup \{CW_b red(b)[S_2]\}$, which is logically equivalent to $\{Cred(a)[S_2], C\neg red(b)[S_2]\}$.

7 Conclusions

In this paper, we studied progression of knowledge in the situation calculus. We showed that for physical actions, this reduces to forgetting predicates in first-order modal logic. We studied forgetting in first-order modal logic, and identified a class of first-order modal formulas for which forgetting an atom is definable in first-order modal logic, and forgetting a predicate is definable in second-order modal logic. This class of formulas goes beyond formulas without quantifying-in. We showed that for local-effect physical actions, when the initial KB is a formula in this class, progression of knowledge is definable in first-order modal logic. As for sensing actions, we showed that when the initial KB does not contain negative knowledge, progression can be achieved by simply adding knowing whether the sensed formula holds. Finally, we extended our results to the multi-agent case. For the future, we would like to do an in-depth study of forgetting in first-order modal logic. We would also like to investigate how to do progression wrt sensing actions in the general case. It would also be interesting to explore progression for only-knowing.

Acknowledgments

We thank the anonymous reviewers for helpful comments. This work was supported in part by the Natural Science Foundation of China under Grant No. 61073053.

References

[French, 2005] T. French. Bisimulation quantified logics: Undecidability. In *Proc. of Foundations of Software Technology and Theoretical Computer Science*, 2005.

[Ghilardi *et al.*, 2006] S. Ghilardi, C. Lutz, F. Wolter, and M. Zakharyaschev. Conservative extensions in modal logic. In *Proc. Advances in Modal Logic*, 2006.

[Hughes and Cresswell, 1996] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[Lin and Reiter, 1994] F. Lin and R. Reiter. Forget it! In *Working Notes of AAAI Fall Symposium on Relevance*, 1994.

[Lin and Reiter, 1997] F. Lin and R. Reiter. How to progress a database. *Artificial Intelligence*, 92(1–2):131–167, 1997.

[Liu and Lakemeyer, 2009] Y. Liu and G. Lakemeyer. On first-order definability and computability of progression for local-effect actions and beyond. In *Proc. IJCAI-09*, 2009.

[Moore, 1985] R. C. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*. 1985.

[Reiter, 2001] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. 2001.

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

[Vassos and Levesque, 2008] S. Vassos and H. J. Levesque. On the progression of situation calculus basic action theories: Resolving a 10-year-old conjecture. In *Proc. AAAI-08*, 2008.

[Zhang and Zhou, 2009] Y. Zhang and Y. Zhou. Knowledge forgetting: Properties and applications. *Artificial Intelligence*, 173(16–17):1525–1537, 2009.