Forgetting in Action

David Rajaratnam[1], Hector J. Levesque[2], Maurice Pagnucco[1], Michael Thielscher[1]

[1]University of New South Wales
Sydney, Australia
{daver,morri,mit}@cse.unsw.edu.au

[2]University of Toronto
Toronto, Canada
hector@cs.toronto.edu

Abstract

In this paper we develop a general framework that allows for both knowledge acquisition and forgetting in the Situation Calculus. Based on the Scherl and Levesque (Scherl and Levesque 1993) possible worlds approach to knowledge in the Situation Calculus, we allow for both sensing as well as explicit forgetting actions. This model of forgetting is then compared to existing frameworks. In particular we show that forgetting is well-behaved with respect to the contraction operator of the well-known AGM theory of belief revision (Alchourrón, Gärdenfors, and Makinson 1985) but that knowledge forgetting is distinct from the more commonly known notion of logical forgetting (Lin and Reiter 1994).

Introduction

Typical Situation Calculus models do not consider the need for agents to forget knowledge. This reflects the types of applications to which formal models of agent action and behaviour have typically been applied. However, it is possible to identify classes of applications for which the need to model forgetting in an agent becomes crucial. Consequently, in this paper we develop a general model for forgetting acquired knowledge in the Situation Calculus. We then highlight useful sub-classes of this model.

The focus of this paper is to develop the theoretical machinery of agent forgetting. However, in order to motivate the importance of forgetting, we highlight two broad classes of potential applications: the modelling of bounded agents and the formal analysis of security protocols.

The most immediate application of forgetting is to model agents with limited resources (e.g., robots), or agents that need to deal with vast knowledge bases (e.g., cloud computing), or more ambitiously, dealing with the problem of lifelong learning. In all such cases it is no longer reasonable to assume that all knowledge acquired over the operation of an agent can be retained indefinitely. Furthermore, recent research on bounded Situation Calculus theories (Giacomo, Lespérance, and Patrizi 2012; 2013) shows that ensuring bounds on such theories can guarantee decidability of progression. The theory of forgetting we introduce here can be used to guarantee bounded Situation Calculus theories.

A further motivation for the need to consider the concept of forgetting can be seen in the formal analysis of security and cryptographic protocols. Cryptographic protocols have previously been represented and analysed using the Situation Calculus equipped with a notion of the knowledge of agents (Delgrande, Hunter, and Grote 2010). However, such an encoding cannot always represent the class of *nonmonotonic cryptographic protocols* (Rubin and Honeyman 1994). In particular, the need to consider forgetting actions can be important in some nonmonotonic cryptographic protocols such as the analysis of credit card protocols where a critical requirement is for vendors to forget (i.e., not retain) customer credit card details.

In order to highlight the features of our approach to forgetting in the Situation Calculus we provide the following simple scenario as a running example. *A robot needs to enter a room that is protected by a closed door with a keypad lock. When the robot senses that the door is closed, it needs to download the key combination from an external data source (e.g., the cloud or a database). It can then use this key combination to open the door and enter the room. Furthermore, since the door will now be open, the robot will no longer need the key combination so will be free to forget this information.*

The rest of the paper proceeds as follows. First we introduce the Situation Calculus (McCarthy 1963; Reiter 2001) and its epistemic extension (Lesperance et al. 1995) that allows for knowledge acquisition but not forgetting. We then present our approach that handles both knowledge acquisition and forgetting and perform an extensive analysis of its properties and the conditions under which both knowledge acquisition and forgetting can occur. Having established the main properties of our approach, we then place our model of forgetting within the broader context of two of the main models that have been developed within the literature: *AGM belief revision* (Alchourrón, Gärdenfors, and Makinson 1985) and *logical forgetting* (Lin and Reiter 1994). In particular, we show that our approach is well-behaved with respect to the AGM belief contraction postulates, but is distinct from that of logical forgetting since knowledge forgetting can provide for more fine-grained control over what is forgotten. Finally, we provide some concluding remarks and discuss directions for future research.

Background

Situation Calculus

The Situation Calculus provides a formal language based on classical first-order logic to describe dynamic domains (McCarthy 1963; Reiter 2001). It distinguishes three types of terms: *situations* representing histories as the world evolves; *fluents* denoting domain properties that may change as a result of actions; and, *actions* that can be performed by the reasoner.

The function $do(a, s)$ represents the situation that results from performing action a at situation s, while S_0 denotes the initial situation where no actions have taken place. For each action, a *precondition axiom* $Poss(a, s)$ specifies the conditions under which action a is possible in situation s and *successor state axioms* specify how the value of fluents change as the result of actions. Assuming perfect knowledge and no sensing actions, the precondition and successor state axioms for our example scenario could be encoded as follows:

$$\begin{aligned}
Poss(enter, s) &\equiv Open(s) \wedge \neg InRoom(s)\\
Poss(usekey, s) &\equiv \exists c.Key(c, s) \wedge \neg Open(s)\\
InRoom(do(a, s)) &\equiv InRoom(s) \vee a = enter\\
Key(c, do(a, s)) &\equiv Key(c, s)\\
Open(do(a, s)) &\equiv Open(s) \vee a = usekey
\end{aligned}$$

The first states that it is possible to enter the room if the robot is not already there and the door is open. The second states that it is possible to use the key code to open the door if there is a key code and the door is not already open. Finally, three successor state axioms are defined to indicate that: the robot will be in the room if it just entered it or was already there; key codes for the door do not change; and, the door will be open if it was already opened or the key code was entered.

With this formalisation it is possible to follow how the world evolves as actions are performed and, in particular, that the robot will be in the room if it inputs the key code, thus opening the door, and then enters the room.

It is useful to be able to refer to the relationship between situations resulting from a sequence of actions. In order to do this a partial order relation $<$ is defined over situations in order to indicate that one situation is the result of performing a sequence of actions in another situation:

$$\neg s < S_0, \quad s < do(a, s') \equiv s \leq s'$$

where $s \leq s'$ is shorthand for $s < s' \vee s = s'$. For a more comprehensive formulation of what is required of a Situation Calculus basic action theory, we refer to (Reiter 2001).

Situation Calculus with Knowledge

In its basic form the Situation Calculus can be used to show how the world changes in the face of actions. However, it lacks the ability to model how an agent with imperfect knowledge can both gain knowledge and act on that knowledge. To deal with this broader problem the Situation Calculus has been extended using a possible worlds framework to capture a notion of knowledge with sensing actions (Scherl and Levesque 1993). As we shall be drawing repeated comparisons between this extension and our own approach we shall refer to this extension as the *SL framework*.

Within the SL framework, an agent can be said to *know* that some fact ϕ is true if and only if ϕ is true in all possible states of the world that the agent can be in. This can be formalised using a special epistemic predicate $K(s', s)$ to represent the fact that in situation s the agent considers the world could equally be in situation s'. Situation s' is said to be *accessible* from s. Notions of knowledge are defined in terms of this epistemic relation:[1]

$$\begin{aligned}
\mathbf{Know}(\phi, s) &\stackrel{\text{def}}{=} \forall s'.K(s', s) \supset \phi[s']\\
\mathbf{KnowIf}(\phi, s) &\stackrel{\text{def}}{=} \mathbf{Know}(\phi, s) \vee \mathbf{Know}(\neg\phi, s)
\end{aligned}$$

The first definition states that an agent in situation s *knows* that fluent ϕ holds when ϕ holds in all *K* accessible situations. The second states that to *know if* something holds means to either know that it holds or to know that it doesn't hold. The predicate *K* is required to be reflexive, transitive and Euclidean, ensuring that the agent has introspection about its knowledge; it knows whether it knows something.

To allow for introspective statements, that is, to represent that an agent knows that it knows something, **Know** can be treated as a term when used in conjunction with the pseudo-variable *now* (Lesperance et al. 1995). This usage is clear from the following expansion:

$$\mathbf{Know}(\mathbf{Know}(\phi, now), s) \equiv \forall s'.K(s', s) \supset \mathbf{Know}(\phi, s')$$

Now, reasoning about knowledge is not very interesting unless that knowledge can itself change over time and sensing actions provide a mechanism to achieve this. As with non-sensing actions, sensing actions require precondition axioms to determine when they are executable. For our example scenario we introduce two sensing actions *isopen* and *getkey* that are always executable. The first senses if the door is open while the second models the process of querying a data source for the correct key combination:

$$\begin{aligned}
Poss(getkey, s) &\equiv True\\
Poss(isopen, s) &\equiv True
\end{aligned}$$

Next, the process of knowledge acquisition is encoded through the successor state axiom of the *K* relation. An example of such a successor state axiom, containing sensing actions for our robot scenario would be:

$$\begin{aligned}
K(s^*, do(a, s)) \equiv &\exists s'.s^* = do(a, s') \wedge Poss(a, s') \wedge K(s', s)\\
&\wedge a = isopen \supset [Open(s') \equiv Open(s)]\\
&\wedge a = getkey \supset [\forall k.Key(k, s') \equiv Key(k, s)]
\end{aligned}$$

For non-sensing actions, the accessible situations are simply the successors of all the situations that were previously accessible. However, for sensing actions only those situations that agree with the actual situation on the sensed fluent remain accessible. So, in the example scenario, the *isopen* sensing action ensures that only situations where the state of the door is the same are accessible, while the *getkey* action ensures that only the situations where all the keys are the same are accessible. Effectively, after performing the *isopen*

[1] $\phi[s]$ is the commonly used notation to represent that formula ϕ holds in the situation s; we use the name **KnowIf** rather than **KnowWhether** introduced in (Scherl and Levesque 1993).

action, the agent will know if the door is open and after the *getkey* action the agent will know about any keys.

The final component of the representation is to specify the truth of fluents in the initial situation and their possible alternatives. For example, that the door is closed but the robot doesn't know it and the door's key combination is C_1 but the robot only knows that it must be one of C_1 or C_2. Figure 1(a) shows a visualisation of this scenario and the accessibility of the alternative possible situations as the robot executes the sequence of actions resulting in it entering the room. Note, for simplicity and presentation we have only specified two possible keys but, of course, this could be extended to many key combinations through the specification of more alternative initial situations.

There is an important observation that can be made from the inclusion of the *K* relation and sensing actions. Namely, that it is possible to construct cases of hybrid sensing actions that both sense and change the environment by allowing sensing actions to also be contained in the non-*K* (i.e., fluent) successor state axioms. In such cases knowledge acquisition cannot be guaranteed because the sensing action could itself change the value of the sensed fluent. To avoid such pathological cases it is typical to consider only *pure* sensing actions where the sensing action only occurs in the successor state axiom for *K*.

A General Account of Knowledge and Forgetting

In this section we present our extension to the SL framework by incorporating the ability to forget the results of past sensing actions. The possible worlds approach is maintained, however we provide a simpler construction in terms of a possible situations relation, $W(s, s')$, which is then used to derive more complex epistemic properties.

Axiom 1 *Let* W *be a relation that is reflexive, transitive and Euclidean. The* possible worlds successor state axiom *for* W *is:*

$$\mathrm{W}(s^*, \mathrm{do}(a, s)) \equiv \exists s'.(s^* = \mathrm{do}(a, s') \wedge \mathrm{W}(s', s) \wedge \mathit{Poss}(a, s'))$$

This successor state axiom works in a similar manner to the original axiom for *K* in the sense that it tracks the evolution of possible situations as a result of actions. However, unlike the original *K* relation, *W* makes no reference to how the agent's knowledge changes as a result of these actions.

To account for sensing actions we first introduce the notion of a *sensing equivalence* relation *SEQ* as a means of separating the generic aspects of the formalism from those that are specific to a particular scenario:[2]

Definition 1 *Consider a Situation Calculus basic action theory* Σ*, a set of sensing actions* $a_1, \ldots, a_n$ *and corresponding sensed fluents* $\phi_1, \ldots, \phi_n$*. Then* $\mathrm{SEQ}(a, s', s)$ *is a* sensing equivalence *relation iff there exists an axiom in* Σ *of the form:*

$$\mathrm{SEQ}(a, s', s) \equiv \bigwedge_{i=1}^{n} a = a_i \supset [\nabla_i.(\phi_i[s'] \equiv \phi_i[s])],$$

where each ∇_i *is a composition of first-order quantifiers over the non-situation free variables in* ϕ_i*.*

In the definition of sensing equivalence each of the sensing actions a_i senses whether a formula ϕ_i is true or whether it is false in a given situation. For brevity we simply say that a_i *senses* ϕ_i. The equivalence specified by an *SEQ* relation is problem specific and encodes the knowledge that is acquired by the sensing actions. For the example scenario we can provide the following sensing equivalence relation:

$$\begin{aligned}\mathit{SEQ}(a, s', s) \equiv\ & (a = \mathit{isopen} \supset [\mathit{Open}(s') \equiv \mathit{Open}(s)]) \wedge \\ & (a = \mathit{getkey} \supset [\forall k.\mathit{Key}(k, s') \equiv \mathit{Key}(k, s)])\end{aligned}$$

The *W* relation and successor state axiom in combination with the sensing equivalence relation can now be used to both capture the original Scherl and Levesque notion of knowledge as well as define a more general notion of knowledge that allows for forgetting actions.

Definition 2 *Consider a Situation Calculus basic action theory, extended with the successor state axiom for* W *and a sensing equivalence relation* SEQ*. Then* $\mathrm{K_M}$ *is a* knowledge acquisition relation *defined as:*

$$\begin{aligned}\mathrm{K_M}(s', s) \stackrel{\mathrm{def}}{=}\ & \mathrm{W}(s', s) \wedge \\ & [\neg\exists \hat{s}', \hat{s}, a.\mathrm{W}(\hat{s}', \hat{s}) \wedge \neg\mathrm{SEQ}(a, \hat{s}', \hat{s}) \wedge \\ & \quad \mathrm{do}(a, \hat{s}') \leq s' \wedge \mathrm{do}(a, \hat{s}) \leq s]\end{aligned}$$

Essentially, this definition states that every *W* relation is also a K_M relation unless it has been *blocked* by a sensing action for which the two situations disagree with respect to the sensed fluents. The definition of **Know** would then be defined in terms of K_M in the standard manner outlined earlier. For space reasons we do not prove this here but it is straightforward to observe that the original notion of agent knowledge is exactly captured by this new formulation.

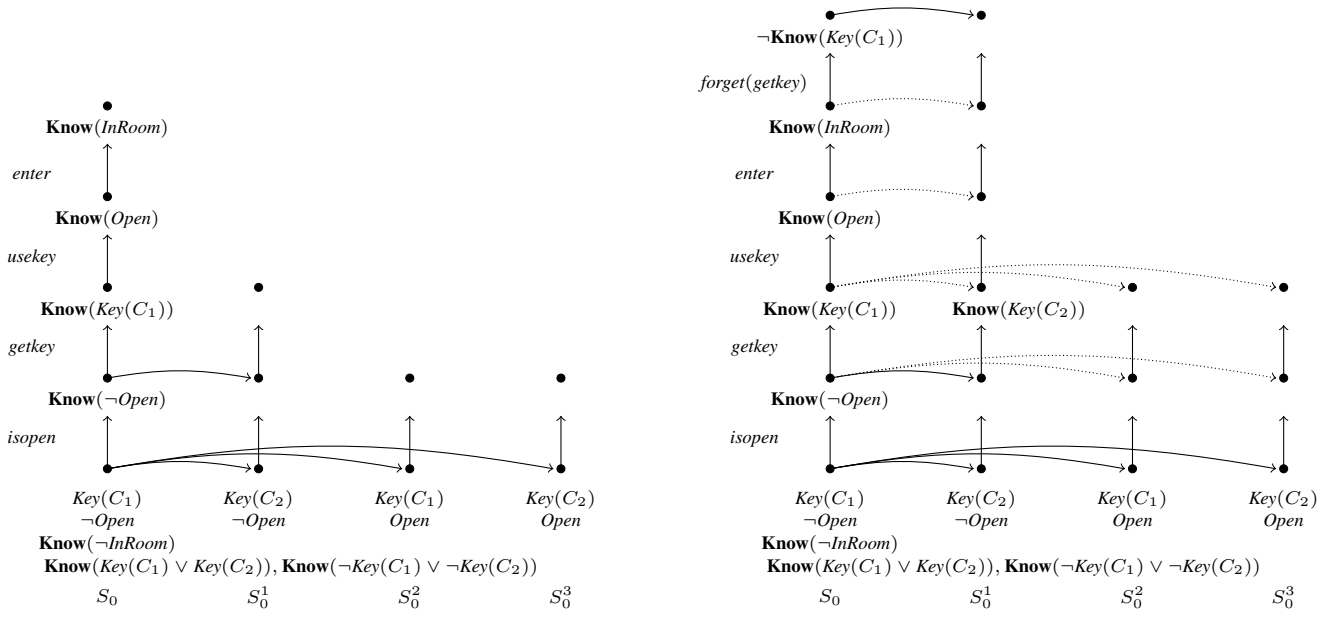
However, we are interested in defining a more general notion of knowledge. This is possible with the help of an explicit forgetting action *forget* that enables an agent to forget previously acquired knowledge.

Definition 3 *Consider a Situation Calculus basic action theory extended with the successor state axiom for* W *and a sensing equivalence relation* SEQ*. Then* K *is a* general knowledge relation *defined as:*

$$\begin{aligned}\mathrm{K}(s', s) \stackrel{\mathrm{def}}{=}\ & \mathrm{W}(s', s) \wedge \\ & [\neg\exists \hat{s}', \hat{s}, a.\mathrm{W}(\hat{s}', \hat{s}) \wedge \neg\mathrm{SEQ}(a, \hat{s}', \hat{s}) \wedge \\ & \quad \mathrm{do}(a, \hat{s}') \leq s' \wedge do(a, \hat{s}) \leq s \wedge \\ & \quad \{\neg\exists s^{*\prime}, s^*, b.b = \mathrm{forget}(a) \wedge \\ & \qquad \mathrm{do}(a, \hat{s}') < \mathrm{do}(b, s^{*\prime}) \leq s' \wedge \\ & \qquad \mathrm{do}(a, \hat{s}) < \mathrm{do}(b, s^*) \leq s\}]\end{aligned}$$

Again, here **Know** is defined in the usual manner. *K* works in a similar manner to K_M in that it allows sensing actions to act as a block on the accessible situations. However, it further allows this block to itself be blocked by a forgetting action, thus re-instating a *K* accessible situation that might otherwise be inaccessible. In effect the agent will have forgotten what it had previously learned by sensing.

[2] Later versions of the SL framework use similar approaches, introducing a sensing result function (Scherl and Levesque 2003) or sensing fluent (Levesque 1996).



(a) Knowledge in the SL Framework

(b) Knowledge with Forgetting

Figure 1: The progression of actions leading to the robot being in the room, and in (b) forgetting the key combination. Vertical lines indicate actions that are performed from the accessible situations. Solid curved horizontal lines represent *K* accessibility from the actual situation S_0 to the alternative possible situations, while in (b) dotted lines represent *W* only accessibility.

It is important to note that the single parameter of the forgetting action is itself an action, with the only useful case being that of a sensing action. Consequently, this construction formalises the notion of an agent that forgets the epistemic commitments of a previous sensing action rather than directly forgetting the value of a specified fluent.

Furthermore, the agent retains the knowledge that the sensing action itself did in fact take place by virtue of the action history encoded in the situation. However, it simply cannot remember the result of that action. We can understand this behaviour as the distinction between *semantic* and *episodic* memory (see, for example, (Schacter, Gilbert, and Wegner 2011)). Specifically, we are concerned with forgetting semantic memory. Forgetting of episodic memory corresponds to the robot forgetting some of its past actions which can be achieved in a Situation Calculus context through the use of *progression* (Reiter 2001).

In order to simplify the encoding of knowledge acquisition and forgetting as part of a Situation Calculus basic action theory we introduce some restrictions as follows:

Definition 4 A Situation Calculus Epistemic Basic Action Theory (EBAT) *is a Situation Calculus Basic Action Theory* Σ *extended with the successor state axiom for* W *such that:*

1. *A precondition axiom for forgetting actions:* $Poss(\text{forget}(a), s) \equiv True$*;*
2. *The successor state axioms of* Σ *do not contain a reference to the function symbol* forget*;*
3. *The sensing equivalence relation (*SEQ*) does not contain a reference to the function symbol* forget*;*
4. *All sensing actions are* pure*; and,*
5. *Along with the other standard basic action theory axioms for the Situation Calculus extended with knowledge (see (Scherl and Levesque 2003)).*

These restrictions result in three distinct and non-overlapping sets of actions: sensing actions, forgetting actions and actions that change the world. As they are concerned with the agent's knowledge, where convenient we refer to the sensing and forgetting actions as *epistemic actions*.

The requirement for forgetting actions to always be executable and not mentioned in the successor state axioms is not strictly necessary but simplifies the formalisation. In particular, not allowing forgetting actions to be contained in either successor state axioms or the sensing equivalence relation ensures that forgetting actions serve only one purpose, namely to forget, and cannot be misappropriated for multiple purposes. Of course, one can always simulate such multi-purpose actions if necessary by introducing individual actions that must be executed in a block. Clearly these restrictions are not onerous.

Framework Analysis and Properties

We now turn to analysing the properties of our framework. In particular, we establish intuitive knowledge theorems for the framework. This does however require certain restrictions on the formalisation of a given scenario, the reasons for which require some explanation.

Limits of the Possible Worlds Approach

Contingent facts Firstly, it is important to observe that our framework inherits the limitations of the SL framework in that the set of initial situations establishes the boundaries over what knowledge about the world is subject to change and what must remain either known or unknown throughout. We refer to such changeable knowledge as *contingent* and define this notion in relation to individual situations:

$$\textbf{Contingent}(\phi, s) \stackrel{\text{def}}{=} (\exists s'.W(s', s) \wedge \phi[s']) \wedge (\exists s'.W(s', s) \wedge \neg\phi[s'])$$

This establishes that a proposition is contingent in a situation if there is at least one related possible situation in which the proposition is true and another in which it is false. For completeness we say that a proposition is *necessarily* the case in a situation if it is not contingent in that situation.

The contingent facts are precisely those facts that are potentially discoverable about the world and therefore the target for sensing actions. However, it is important to observe that the set of contingent facts can vary as situations evolve. For example, if a door being open is contingent in some initial situation but an agent subsequently senses that the door is open and walks through it, then in the resulting situation the state of the door will no longer be contingent because there cannot be a possible world in which the agent walked through the closed door. Less obviously, a fact that was previously necessarily the case can become contingent after performing some action. For example, the action of replicating the truth value of a contingent fluent.

Knowledge acquisition without sensing While sensing is the primary means by which an agent can acquire knowledge, the theoretical construction of the framework masks a second method by which knowledge can be acquired without explicit sensing actions. This is a property that is implicit in the original SL framework as well as its various extensions, such as that of Shapiro *et al.* (Shapiro et al. 2011).

In essence the problem stems from how the precondition axioms are encoded and whether the representation of these axioms is intended to model the actions that an agent knows that it can execute, or the executability of actions irrespective of the agent's knowledge of whether those actions are executable. To clarify this problem, consider the scenario where the robot does not initially know if the door is open. Now, if the door happens to be open in the actual situation, then a precondition axiom defined without reference to the robot's knowledge allows it to walk through the door without knowing that it is open. However, in the alternative possible situation where the door was closed, this action was not possible, hence there will be no successor situation corresponding to the situation where the door is closed. Consequently, by definition the robot will "know" that the door is closed, despite never having sensed this about its environment.

At first blush this may not appear to be problematic. One might, for example, argue that if a person has their eyes closed and successfully walks through a door then they will know that the door must have been open in order for the action to have succeeded. However, such reasoning ignores the many sensing actions that would be taking place in such a scenario. Namely, that the person is able to know that the action was successful because they didn't feel the pain of hitting the door and finding themselves on the floor.

This problem can in practice be avoided with some added theoretical machinery. In particular, in order to formalise reasoning about agent *abilities*, the SL framework has been extended with the notion of an *action selection function* (Lesperance et al. 1995). This is a second-order construct that allows conditions to be defined such that an agent *can know whether or not it can get* to a situation where some property holds. Provided that one only considers sequences of actions where the agent knows that it can get to some state, then the problem of knowledge acquisition without sensing can be avoided.

Nevertheless the issue remains a troubling property of the framework. It is particularly problematic for our extension as it can affect the contingency of facts and consequently the conditions under which forgetting can take place.

Fortunately, the solution to this problem is relatively simple. Namely, to ensure that all action preconditions are specified so as to model the actions that the agent knows that it can execute. We shall refer to these as actions with *known preconditions*. With this in mind, the precondition axioms for the sample scenario can now be re-stated as follows:

$$\mathit{Poss}(\mathit{enter}, s) \equiv \textbf{Know}(\mathit{Open}, s) \wedge \textbf{Know}(\neg \mathit{InRoom}, s)$$
$$\mathit{Poss}(\mathit{usekey}, s) \equiv \exists k.\textbf{Know}(\mathit{Key}(k), s) \wedge \textbf{Know}(\neg \mathit{Open}, s)$$

Now it is possible for the robot to execute the *enter* action only when it knows that it is not currently in the room and knows that the door is open. Similarly, in order for the robot to use the key combination to open the door, it must know the combination and know that the door is closed. Note, actions that are always executable, such as the example sensing actions, implicitly have known preconditions (i.e., a tautology is true in every possible world and hence is known).

Epistemic Properties

We can now turn to establishing that the knowledge with forgetting framework genuinely satisfies the intuitive properties of an epistemic action theory. Namely, we establish that it satisfies the basic introspective properties of knowledge, that knowledge can indeed be acquired through sensing actions and, finally, that knowledge can also be forgotten through forgetting actions.

Introspection As is the case of the SL framework, our extension can be shown to also satisfy a strong notion of knowledge introspection. As a first step, it is useful to establish some key properties of the defined *K* relation.

Lemma 1 *Consider a Situation Calculus EBAT. The defined relation* K *is* reflexive*,* transitive *and* Euclidean*.*

Proof: Reflexivity is immediate since for every $W(s, s)$ relation $K(s, s)$ exists by definition. We now consider only the transitive case as the Euclidean case follows an identical argument. Consider if $K(s, s')$ and $K(s', s'')$ but $\neg K(s, s'')$. W is transitive so as $W(s, s')$ and $W(s', s'')$ therefore $W(s, s'')$. Hence $K(s, s'')$ is being blocked from holding by some sensing action a such that the fluent it senses, ϕ, is different

between s and s''. But then the value of ϕ in s' must be different to one of either s or s''. Hence either $K(s, s'')$ or $K(s', s'')$ must be false and a contradiction follows. □

These properties of the *K* relation can now be used to establish that knowledge is indeed introspective.

Theorem 1 (Introspection) *Let Σ be a Situation Calculus EBAT, then for any ϕ and situation s:*

$$\Sigma \models \mathbf{Know}(\phi, s) \textit{ iff } \Sigma \models \mathbf{Know}(\mathbf{Know}(\phi, now), s)$$

Proof: Assume that LHS holds but the RHS doesn't hold. Hence $\forall s'.K(s', s) \supset \phi[s']$ holds but $\forall s'.K(s', s) \supset (\forall s''.K(s'', s') \supset \phi[s''])$ does not. However, the only way for the second formula not to hold is if there is some s' and s'' such that $K(s', s)$ and $K(s'', s')$ and $\neg\phi[s'']$. But, from Lemma 1, *K* is transitive so $K(s'', s)$ must hold and contradicts with the LHS.

Now assume that the RHS holds but the LHS doesn't. So there exists some s^* such that $K(s^*, s)$ and $\neg\phi[s^*]$. Substituting s^* for an instance of s' in the RHS we have that $\forall s''.K(s^*, s) \supset K(s'', s^*) \supset \phi[s'']$ and since $K(s^*, s)$ holds so $\forall s''.K(s'', s^*) \supset \phi[s'']$. Now *K* is reflexive so $K(s^*, s^*)$ holds and therefore $\phi[s^*]$ must hold, which contradicts the assumption that the LHS doesn't hold. □

Knowledge Acquisition Naturally, it is important that an agent can actually gain knowledge through sensing actions, namely after sensing a contingent fluent the agent will know the truth value of that fluent.

Theorem 2 *Let Σ be a Situation Calculus EBAT, a_ϕ be a pure sensing action that senses fluent ϕ and is executable in situation s, then:*

1. $\Sigma \models \mathbf{Know}(\phi, \text{do}(a_\phi, s))$*, where ϕ is true in s, or*
2. $\Sigma \models \mathbf{Know}(\neg\phi, \text{do}(a_\phi, s))$*, otherwise.*

Proof: Case 1. Consider any s' such that $W(s', s)$ and $\neg\phi[s']$. If $Poss(a, s')$, then there will be a $W(do(a_\phi, s'), do(a_\phi, s))$ but a_ϕ detects the value of ϕ therefore $\neg SEQ(a_\phi, s', s)$ holds. Alternatively, $\neg Poss(a, s')$ so there will be no $W(do(a_\phi, s'), do(a_\phi, s))$. In either case $K(do(a_\phi, s'), do(a_\phi, s))$ will not hold. Hence the only situations in which $K(do(a_\phi, s'), do(a_\phi, s))$ holds will be those where the fluent ϕ will also hold, so $\mathbf{Know}(\phi, do(a_\phi, s))$. Case 2. follows in an identical manner. □

Together Theorems 1 and 2 establish that the new framework does indeed capture the properties of the SL framework.

Forgetting In a similar manner to the establishment that not all facts are *contingent*, forgetting allows only some facts to be forgotten. We now establish precise conditions under which this can take place.

The simplest case of forgetting is where a forgetting action is performed immediately after its corresponding sensing action. Interestingly, even in this case there are some conditions that are required. The critical property is to preserve the contingency of a fluent after the sensing action. Unfortunately, even for a pure sensing action, contingency is not necessarily preserved, and it is possible to provide a simple example to highlight this fact.

Consider a sensing action a_ϕ that senses ϕ and has the precondition that it can sense ϕ only when ϕ is itself known (i.e., $Poss(a_\phi, s) \equiv \mathbf{Know}(\phi, s)$). A set of possible situations can then be defined such that ϕ is both known and contingent. In such a case, after performing the sensing action, ϕ will still be known but will no longer be contingent.

The contrived nature of this example provides some hint that such sensing actions are not the typical types of sensing actions that we would want to consider and therefore we restrict our actions to remove such pathological cases.

Definition 5 *An action a is* contingency preserving *w.r.t. ϕ in s if ϕ is contingent in s iff ϕ is contingent in* $\text{do}(a_\phi, s)$.

In general it is reasonable to expect that an action that senses a fluent should be contingency preserving for that fluent, although not necessarily other fluents. Furthermore, it is straightforward to observe that, because forgetting actions are always possible, therefore they are contingency preserving for all fluents in all situations. We now establish the simplest condition under which forgetting can take place; cases where a fluent is forgotten immediately after sensing.

Lemma 2 *Let Σ be a Situation Calculus EBAT such that fluent ϕ is contingent in s and a_ϕ is a pure sensing action that preserves the contingency of ϕ in s, then:*

$$\Sigma \models \neg\mathbf{KnowIf}(\phi, \text{do}([a_\phi, \text{forget}(a_\phi)], s))$$

Proof: Both, $\Sigma \models \neg\mathbf{Know}(\phi, do([a_\phi, forget(a_\phi)], s))$ and $\Sigma \models \neg\mathbf{Know}(\neg\phi, do([a_\phi, forget(a_\phi)], s))$ must be shown, but the proofs are identical so consider only the first. Similarly, whether ϕ is *true* or *false* in s are mirrored so only consider the first case. ϕ is contingent in s and a_ϕ preserves contingency, therefore after executing a_ϕ, ϕ will be contingent in $do(a_\phi, s)$ although ϕ will also be known (Theorem 2). Hence there is some $\neg SEQ(a, s', do(a_\phi, s))$ that is blocking $K(s', do(a_\phi, s))$ from holding. Forgetting actions preserve contingency so, after performing the forgetting action $forget(a_\phi)$, then any such block will itself be blocked by the forgetting action and therefore there will exist a world s'' such that $K(s'', do([a_\phi, forget(a_\phi)], s))$ and $\neg\phi[s'']$ hence ϕ will not be known. □

Of course, sensing followed immediately by forgetting is not a particularly useful scenario. Therefore we now consider the case where some arbitrary number of actions are executed in between performing the sensing and forgetting actions. However, in order for this to be possible, it is necessary to establish that the actions that occur between a sensing action and its corresponding forgetting action are in some sense independent of the fact being forgotten.

The key point in establishing a notion of the independence of fluents and actions is to observe that the preconditions and successor state axioms of a basic action theory establish dependencies between actions in terms of how they relate to fluents. If the value of some fluent is related to an action, then it may not be possible to forget the value of that fluent without first forgetting the action itself. However, as already outlined, our framework only provides a mechanism to forget the value of fluents and does not provide a mechanism to forget the history of past actions.

To properly capture these notions requires the definition of a number of properties. Firstly, it is important to capture how non-epistemic actions interact with particular fluents.

Definition 6 (Consequential Independence) *For a Situation Calculus EBAT Σ, an action a is said to be* consequentially independent *from ϕ iff:*

$$\Sigma \models \forall s.\phi[s] \equiv \phi[\text{do}(a, s)]$$

This definition captures whether or not an action can change the value of a fluent under some condition. Importantly, it makes no reference to the executability of an action in a situation. Instead, it only examines the effects of actions irrespective of whether some precondition has been satisfied. Hence it is a strong requirement that genuinely establishes that an action can have no direct effect on some property of the environment. For example, that picking up a cup has no direct effect on the state of a light.

As well as consequential independence, it is also necessary to consider the *epistemic independence* of fluents from an agent's actions.

Definition 7 (Epistemic Independence (non-forgetting)) *For a Situation Calculus EBAT Σ, a non-forgetting action a is said to be* epistemically independent *of ϕ in s iff:*

$$(\textbf{Know}(\phi, s) \equiv \textbf{Know}(\phi, \text{do}(a, s))) \wedge$$
$$(\textbf{Contingent}(\phi, s) \equiv \textbf{Contingent}(\phi, \text{do}(a, s)))$$

This establishes two specific criteria for an action to be considered epistemically independent of a fluent. Most obviously it cannot change the agent's knowledge of that fluent. This applies to directly sensing the fluent but also any indirect epistemic ramifications of the action. For example, sensing that a light is on means knowing that the switch is also turned on. However, equally important is that the action must preserve the contingency of a fluent.

Definition 7 only captures the notion of epistemic independence for non-forgetting actions. However, forgetting actions also affect what an agent knows and it is therefore necessary to extend this notion to forgetting actions.

Definition 8 (Epistemic Independence (forgetting)) *For a Situation Calculus EBAT Σ, a forgetting action* forget(a_ϕ) *is said to be* epistemically independent *from ϕ in s iff a_ϕ is epistemically independent of ϕ in s.*

Simply, a forgetting action is epistemically independent of a fluent in exactly the cases where its corresponding sensing action is epistemically independent of that fluent.

We can now define a general notion of the independence of actions from fluents by grouping together the various specific types of independence.

Definition 9 (Independence) *For a Situation Calculus EBAT Σ, an action a is independent from fluent ϕ in situation s iff a is consequentially independent of ϕ in s and a is epistemically independent (forgetting and non-forgetting) of ϕ in s.*

Now that we are armed with a formal notion of the independence of actions from fluents, we can establish the conditions under which a fluent will be forgettable in a situation.

Definition 10 (Forgettable) *We say that a fluent ϕ is* forgettable *in situation s iff there is a situation s' such that $s = \text{do}([a_\phi, a_1, \ldots, a_n], s')$, where ϕ is contingent in s', a_ϕ is a pure sensing action that senses ϕ and preserves the contingency of ϕ in s', and:*

1. *action a_1 is independent of ϕ in $\text{do}(a_\phi, s')$; and,*
2. *for each $1 \leq i < n$, action a_{i+1} is independent of ϕ in $\text{do}([a_\phi, a_1, \ldots, a_i]), s')$.*

It can now be established that our defined notion of forgettability does indeed capture the right conditions to allow the results of a sensing action to be known in one situation but forgotten in a some later situation.

Theorem 3 *Let Σ be a Situation Calculus EBAT such that fluent ϕ is forgettable in situation s and action a_ϕ senses ϕ, then:*

$$\Sigma \models \neg\textbf{KnowIf}(\phi, \text{do}(\text{forget}(a_\phi), s))$$

Proof: Expanding, both $\Sigma \models \neg\textbf{Know}(\phi, do(forget(a_\phi), s))$ and $\Sigma \models \neg\textbf{Know}(\neg\phi, do(forget(a_\phi), s))$ must be shown, but the proofs are identical so consider only the first case. Similarly, whether ϕ is *true* or *false* in s are mirrored so only consider the first case. Applying the definition of ϕ being forgettable in s we have situation $s = do([a_\phi, a_1, \ldots, a_n], s')$, where ϕ is contingent in s'. Now provable by induction on the size of $a_1, \ldots, a_n$. Base case consider $n = 1$: Similar to Lemma 2. ϕ is known but contingent after action a_ϕ. Now, a_1 is independent of ϕ in $do(a_\phi, s')$ so ϕ is known and contingent in $do([a_\phi, a_1], s')$. Because ϕ is contingent in $do([a_\phi, a_1], s')$ so there is some $W(s'', do([a_\phi, a_1], s'))$ where $\neg\phi[s'']$ holds but is being blocked by some $\neg SEQ(a_\phi, s', s^*)$. Hence after executing $forget(a_\phi)$ this block will itself be blocked so there will be some $K(s''', do([a_\phi, a_1, forget(a_\phi)], s')$ where $\neg\phi[s''']$ so will not be known in $do([a_\phi, a_1, forget(a_\phi)], s')$. Now assume that this holds for $n = i$ and prove for $n = i + 1$. The proof for this is similar to the base case and stems from ϕ being contingent and known in $do([a_\phi, \ldots, a_i], s')$. Since ϕ is independent of a_{i+1} in $do([a_\phi, \ldots, a_i], s')$ so ϕ will also be known and contingent in $do([a_\phi, \ldots, a_{i+1}], s')$. ϕ being unknown in $do([a_\phi, \ldots, a_{i+1}, forget(a_\phi)], s')$ then follows in a straightforward manner. □

Theorem 3 establishes that knowledge can indeed be forgotten; after having been previously acquired through a sensing action.

It is worth observing that a sensing action immediately followed by its corresponding forgetting action results in an agent forgetting both the sensed value and any derived consequences. However, the presence of intermediate actions in between sensing and forgetting can prevent these consequences being lost. For example, if our example scenario (Figure 1(b)) had also included an initially accessible situation where there was no actually key, then after performing all the actions and forgetting the value of the key, the robot would nevertheless have retained the acquired knowledge that a key did in fact exist.

Introspection about Action Consequences A final property that is worth noting concerns the introspective nature of

knowledge and what the agent currently knows about what it will know in the future. It is possible to show that in the current situation the agent knows that after sensing a fluent it will know if that fluent is true or false, however it also knows that it will no longer know this fact after subsequently forgetting the sensing action. Due to space restrictions we only state the theorem without proofs.

Theorem 4 *Let Σ be a Situation Calculus EBAT such that ϕ is forgettable in situation s and action a_ϕ senses ϕ, then:*

$$\Sigma \models \begin{array}{l} \textbf{Know}(\textbf{KnowIf}(\phi, \text{do}(a_\phi, now), s) \ \wedge \\ \textbf{Know}(\neg\textbf{KnowIf}(\phi, \text{do}([a_\phi, \text{forget}(a_\phi)], now), s) \end{array}$$

Relationship to Existing Frameworks

In this section we explore the relationship of our framework to well-known formalisms in the literature that provide for similar behaviours.

AGM Belief Contraction

While we are concerned with notions of knowledge, there is nevertheless a close link to the research on belief change. Belief change is concerned with establishing how (rational) agents should change their beliefs in light of new information. The most well-known formal account of belief change is that of AGM belief revision (Alchourrón, Gärdenfors, and Makinson 1985) which defines rationality postulates that should be preserved by belief change operators.

Of particular interest to our account of forgetting is that the effects of a forgetting action are comparable to the AGM belief contraction operator. For reference, the traditional AGM belief contraction postulates are as follows.

(K–1) For any sentence ϕ and any belief set K, $K - \phi$ is a belief set. (**closure**)
(K–2) If $\phi \notin Cn(\emptyset)$, then $\phi \notin K - \phi$. (**success**)
(K–3) $K - \phi \subseteq K$. (**inclusion**)
(K–4) If $\phi \notin K$, then $K - \phi = K$. (**vacuity**)
(K–5) If $\phi \in K$, then $K \subseteq Cn((K - \phi) \cup \{\phi\})$. (**recovery**)
(K–6) if $\phi \leftrightarrow \psi \in Cn(\emptyset)$ then $K - \phi = K - \psi$. (**extensionality**)

In showing the link to belief contraction, it is first necessary to provide a bridge from the AGM framework to ours. Fortunately, we can adopt the approach already outlined in Shapiro *et al.* (Shapiro et al. 2011). In particular Shapiro *et al.* observe that, in the Situation Calculus framework, an agent is said to believe (in our case, know) a formula if the theory entails that the formula is believed (known). However, unless the theory is complete with respect to the beliefs (knowledge) of the agent, then we may not be able to determine whether some formulas are actually believed (known) or not. In contrast, within belief change frameworks, the beliefs of an agent are explicitly associated with a belief set, with the implicit closed-world assumption that those formulas not in the belief set are not believed. To align the two frameworks Shapiro *et al.* therefore assume a model M of the theory Σ which is used to fix the belief state of the agent. We adopt this approach here and assume a model of the theory in order to fix on the *knowledge state* of the agent.

Definition 11 *We denote the* knowledge state *of a situation s (in M) by $\mathbf{K}(s)$ as:*

$$\mathbf{K}(s) = \{\phi : M \models \textbf{Know}(\phi, s)\}$$

We can now provide a definition of what it means to expand a knowledge state by a formula.

Definition 12 *We denote the expansion of a knowledge state at situation s by ϕ (in M) by:*

$$s + \phi \stackrel{\text{def}}{=} \{\psi : M \models \textbf{Know}(\phi \supset \psi, s)\}$$

It is worth observing that this notion of knowledge state expansion should not be confused with sensing. Sensing actions sense particular fluents and have preconditions axioms that determine when that action is possible. On the other hand, knowledge state expansion simply defines the resulting state when given an initial state and an arbitrary formula that is taken to be known.

The AGM allows for the contraction of any (non-tautological) formula. However, our framework is more selective about what can be forgotten and therefore requires some qualifications.

Definition 13 *We denote the* forgettable *formulas at a situation s by $\mathbf{F}(s)$ as:*

$$\mathbf{F}(s) = \{\phi, \neg\phi : \phi \textit{ is forgettable in } s\}$$

An AGM belief contraction-like operator can now be defined within our extended Situation Calculus framework that specifies the conditions under which contraction is possible.

Definition 14 ($s - \phi$) *We denote the contraction of s by ϕ as $s - \phi$ and define it as follows:*

$$s - \phi \stackrel{\text{def}}{=} \begin{cases} \text{do}(\text{forget}(a_\phi), s), & \textit{if } a_\phi \textit{ senses } \phi \textit{ and} \\ & \phi \textit{ is forgettable in } s; \\ s, & \textit{otherwise.} \end{cases}$$

As a notational convenience we can indicate when two formulas are equivalent with respect to a Situation Calculus basic action theory Σ:

$$\phi \equiv_\Sigma \psi \stackrel{\text{def}}{=} \Sigma \models \forall s.\phi[s] \equiv \psi[s]$$

Finally, we can provide and prove rationality postulates for our contraction operator that mirror the original AGM belief contraction postulates.

Theorem 5 (Contraction) *For any situation s and fluents ϕ and ψ that are forgettable in s:*

(**K**–1) $\mathbf{K}(s - \phi)$ *is deductively closed.* (**closure**)
(**K**–2) $\phi \notin \mathbf{K}(s - \phi)$. (**success**)
(**K**–3) $\mathbf{K}(s - \phi) \subseteq \mathbf{K}(s)$. (**inclusion**)
(**K**–4) *If* $\phi \notin \mathbf{K}(s)$ *and* $\neg\phi \notin \mathbf{K}(s)$ *then* $\mathbf{K}(s - \phi) = s$. (**vacuity**)
(**K**–5) *If* $\phi \in \mathbf{K}(s)$, *then* $\mathbf{K}(s) \subseteq \mathbf{K}(s - \phi) + \phi$ *or* $\mathbf{K}(s) \subseteq \mathbf{K}(s - \phi) + \neg\phi$ (**recovery**)
(**K**–6) *if* $\phi, \psi \in \mathbf{F}(s)$ *and* $\phi \equiv_\Sigma \psi$ *then* $\mathbf{K}(s - \phi) = \mathbf{K}(s - \psi)$. (**extensionality**)

Proof: For (**K**–1) it is straightforward to see that *K* as defined is always deductively closed. (**K**–2) is a direct consequence of Theorem 3. (**K**–3) can be shown by examining the definition of *K* and confirming that what is known after a forgetting action will be a subset of what is known before the forgetting action. (**K**–4) is a direct result of the definition of contraction that only forgettable fluents (or their negations) can be contracted and that only fluents in the knowledge state can be forgotten. (**K**–5) is a consequence of deductive closure, expansion operator $s + [\neg]\phi$, and the defined contraction operator. (**K**–6) is a consequence of the deductive closure of $\mathbf{K}(s)$, $\mathbf{K}(s - \phi)$ and that the resulting knowledge states $\mathbf{K}(s - \phi)$ and $\mathbf{K}(s - \psi)$, ϕ is true in every situation in which ψ in true. □

Theorem 5 shows that forgetting in the Situation Calculus is well-behaved with respect to the AGM contraction postulates. It is worth clarifying the definition of the recovery postulate (**K**–5). While contraction is defined here with respect to a forgetting action, expansion is not defined in terms of sensing actions but rather in terms of a knowledge set. One reason for this is that an expansion operator defined in terms of sensing actions could easily be shown to fail recovery in cases where a sensing action's precondition is not satisfied after the forgetting action. For example, consider a robot that senses whether a light is on in the bedroom, then moves to the kitchen. If it then forgets that the light was on in the bedroom it cannot simply re-sense the fluent as it would first need to move back to the bedroom. However, an important aspect of the AGM recovery postulate is that it provides a notion of minimality; namely, that contraction does not throw away more than it needs to. Consequently, the chosen definition of expansion fits closely the intention of the AGM in defining a notion of recovery.

Literal Forgetting

In this paper we are interested in the notion of *knowledge forgetting*. However, the more commonly studied form of forgetting is that of *logical forgetting* (Lin and Reiter 1994). In a formal sense, logical forgetting involves the replacement of a formula with one that is logically weaker and comes in two forms: forgetting a grounded literal or fact (also known as *literal forgetting*) and forgetting a relation.

While logical and knowledge forgetting are superficially unrelated, nevertheless recent work has established a connection between literal forgetting and belief erasure (Nayak, Chen, and Lin 2007). Therefore, it is worth exploring whether there is any connection between logical and knowledge forgetting. However, we shall show that in general there is no link between these two and that logical forgetting is more drastic in what is forgotten than is strictly necessary for knowledge forgetting.

In the following discussion we only provide a brief outline of the formalism behind literal forgetting, however the interested reader is referred to (Lin and Reiter 1994). Literal forgetting is defined in terms of models of theories where conditions are provided such that a theory T' is the result of *forgetting about* ground atom p in theory T (written *forget*$(T; p)$). A key result is that, for finite theories, literal forgetting can be achieved through a number of purely syntactic manipulations:

1. For formula φ and ground atom $P(\vec{t})$, let $\varphi[P(\vec{t})]$ denote the replacement of occurrences of $P(\vec{t'})$ in ϕ by the disjunctive tautology: $[\vec{t} = \vec{t'} \wedge P(\vec{t})] \vee [\vec{t} \neq \vec{t'} \wedge P(\vec{t'})]$.
2. Denote $\varphi^+_{P(\vec{t})}$ as the result of replacing $P(\vec{t})$ by $true$ in $\varphi[P(\vec{t})]$, and $\varphi^-_{P(\vec{t})}$ as the result of replacing $P(\vec{t})$ by $false$ in $\varphi[P(\vec{t})]$.

Hence for $T = \{\varphi\}$ and ground atom p:

$$\mathit{forget}(T, p) = \{\varphi^+_p \vee \varphi^-_p\}$$

Example 1 (from (Lin and Reiter 1994)) *Let* $T = \{\varphi\}$, *where* $\varphi = \forall x.student(x)$ *and we want to forget that* $student(John)$. *Now,* $\varphi[student(John)]$ *is:*

$\forall x.[x = John \wedge student(John)] \vee [x \neq John \wedge student(x)]$.

Hence $\varphi^+_{student(John)}$ *is equivalent to:*

$$\forall x.x \neq John \supset student(x)$$

and $\varphi^-_{student(John)}$ *is equivalent to:*

$$\forall x.x \neq John \wedge student(x)$$

So forget$(T; student(John))$ *is equivalent to:*

$$\{\forall x.x \neq John \wedge student(x)\}$$

As is clear from Example 1, literal forgetting provides for a strong form of forgetting that removes the given literal entirely from the theory. Unfortunately, for our purposes, this is too strict a requirement for the form of knowledge forgetting that we require. This can be seen with reference to the robot scenario.

Firstly, in order to align the two frameworks, we can reuse the notion of a knowledge state defined in terms of a model (Definition 11). Now, consider the knowledge state of the robot in the example scenario from the point after it has entered the room but before it performs the forgetting action (Figure 1(b)). At this point the robot will certainly know that C_1 is the key combination. However it will also have the base knowledge that exactly one of C_1 and C_2 is the key combination. Hence the robot's knowledge state will contain the following key related clauses: $\mathit{Key}(C_1), \mathit{Key}(C_1) \vee \mathit{Key}(C_2), \neg\mathit{Key}(C_1) \vee \neg\mathit{Key}(C_2)$. Then, after forgetting, the robot will no longer know that C_1 is the key but critically it will still know that the key is exactly one of C_1 or C_2. Hence it will retain the key related facts: $\mathit{Key}(C_1) \vee \mathit{Key}(C_2), \neg\mathit{Key}(C_1) \vee \neg\mathit{Key}(C_2)$. The robot has forgotten the result of the sensing action and no more.

On the other hand, if we were to perform logical forgetting of $\mathit{Key}(C_1)$ on the robot's state, then applying the syntactic method outlined earlier, all references to the literal $\mathit{Key}(C_1)$ will be removed from the knowledge state and each of the three key related clauses will be replaced by $True$. Hence the robot will not only have forgotten that C_1 is the key but also that the key is one of C_1 or C_2.

This example highlights that knowledge forgetting is indeed distinct from logical forgetting. In particular, the knowledge forgetting approach can provide for a more fine-grained form of forgetting where appropriate.

Conclusion and Future Work

In this paper we have motivated the need for agents to be able to forget knowledge. Based on an existing approach to modelling agent knowledge acquisition in the Situation Calculus (Scherl and Levesque 1993), we developed a formal framework that allows for an agent to both acquire and forget knowledge. An extensive analysis of the framework was provided, highlighting important cases under which knowledge acquisition and forgetting can take place. We further evaluated our new framework against existing models that provide for similar behaviour. In particular, we showed that our model is well-behaved with respect to the well-known AGM contraction postulates (Alchourrón, Gärdenfors, and Makinson 1985), but is distinct from the notion of logical forgetting (Lin and Reiter 1994). In this latter case it was shown that knowledge forgetting can provide for more fine-grained control over what information can be forgotten.

This work presents a number of interesting avenues for future research. Importantly, the relationship to notions of forgetting in epistemic logics (Zhang and Zhou 2009; van Ditmarsch et al. 2009) should be explored. While not concerned with environmental change as a result of actions, nevertheless there are commonalities with Situation Calculus formalisms that deserve further exploration. For example, the formalisation of knowledge forgetting from Zhang and Zhou (Zhang and Zhou 2009) defines four postulates for a forgetting operator, which it should be possible to show are satisfied by our notion of forgetting.

It would also be useful to see if the techniques used to achieve forgetting in epistemic logics, for example the *introspective forgetting* approach (van Ditmarsch et al. 2009), can be applied to our framework. Depending on the requirements of the problem domain, this could potentially provide for alternative forgetting actions, perhaps with different properties and fewer restrictions over what knowledge can and cannot be forgotten.

Finally, it would be interesting to explore the applications that were motivated in the introduction. In particular, the application to the analysis of nonmonotonic cryptographic protocols would most probably necessitate an extension of this model to the multi-agent context. Using previous research that extends the original Scherl and Levesque knowledge framework to the multi-agent setting in the context of game playing (Schiffel and Thielscher 2011), it would then be necessary to see if similar techniques could be applied to our extended framework.

Acknowledgements

This research was supported under Australian Research Council's (ARC) *Discovery Projects* funding scheme (project number DP 120102144). The fourth author is the recipient of an ARC Future Fellowship (project number FT 0991348) and is also affiliated with the University of Western Sydney.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50(2):510–530.

Delgrande, J.; Hunter, A.; and Grote, T. 2010. On the representation and verification of cryptographic protocols in a theory of action. In *Proceeding of the 8th International Conference on Privacy, Security and Trust (PST)*, 39–45. IEEE.

Giacomo, G. D.; Lespérance, Y.; and Patrizi, F. 2012. Bounded situation calculus action theories and decidable verification. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 13th International Conference (KR'12)*. AAAI Press.

Giacomo, G. D.; Lespérance, Y.; and Patrizi, F. 2013. Bounded epistemic situation calculus theories. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. IJCAI/AAAI.

Lesperance, Y.; Levesque, H. J.; Lin, F.; and Scherl, R. B. 1995. Ability and knowing how in the situation calculus. *Studia Logica* 66:2000.

Levesque, H. J. 1996. What is planning in the presence of sensing? In *Proceedings of the 13th National Conference on Artificial Intelligence*, AAAI'96, 1139–1146. AAAI Press.

Lin, F., and Reiter, R. 1994. Forget it! In *In Proceedings of the AAAI Fall Symposium on Relevance*, 154–159.

McCarthy, J. 1963. Situations, actions and causal laws. Stanford University Artificial Intelligence Project Memo 2.

Nayak, A. C.; Chen, Y.; and Lin, F. 2007. Forgetting and update – an exploration. In *Formal Models of Belief Change in Rational Agents*, Dagstuhl Seminar Proceedings.

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

Rubin, A., and Honeyman, P. 1994. Nonmonotonic cryptographic protocols. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, 100–116.

Schacter, D. L.; Gilbert, D. T.; and Wegner, D. M. 2011. *Psychology*. Worth Publishers. chapter 6, 240–241.

Scherl, R., and Levesque, H. 1993. The frame problem and knowledge-producing actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 689–695.

Scherl, R. B., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(12):1 – 39.

Schiffel, S., and Thielscher, M. 2011. Reasoning about general games described in GDL-II. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 846–851.

Shapiro, S.; Pagnucco, M.; Lespérance, Y.; and Levesque, H. 2011. Iterated belief change in the situation calculus. *Artificial Intelligence* 175(1):165–192.

van Ditmarsch, H. P.; Herzig, A.; Lang, J.; and Marquis, P. 2009. Introspective forgetting. *Synthese* 169(2):405–423.

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