Introspective forgetting

Hans van Ditmarsch   Andreas Herzig   Jérôme Lang   Pierre Marquis

1. Introduction

[1]There are different ways of forgetting.

Completely forgetting In the movie 'Men in Black', Will Smith makes you forget knowledge of extraterrestials by flashing you with a light in the face. After that, you have forgotten the green ooze flowing out of mock-humans and such: you not remember that you previously had these experiences. In other words, even though for some specific forgotten fact p it is now the case that $\neg Kp$ and $\neg K\neg p$, the flash victims have no memory that they previously knew the value of p. Worse, they forgot that p is an atomic proposition at all. This sort of forgetting is dual to awareness—in a logical setting it is uncommon that parameters of the language, such as the set of atoms, shrink, although there are ways to simulate that. We will leave this matter aside for now.

Remembering prior knowledge A different sort of forgetting is when you forgot which of two keys fits your office door, because you have been away from town for a while. In this case you remember that you knew which key it was, and you currently don't know which key it is. This is about forgetting the value of a atomic proposition p. Previously, either Kp or $K\neg p$, but currently $\neg Kp$ and $\neg K\neg p$. This sort of forgetting will be very central to our concerns.

Forgetting values Did it ever happen to you that you met a person whose face you recognize but whose name you no longer remember? Surely! Or that you no longer know the pincode of your bankcard? Hopefully not. But such a thing is very conceivable. This sort of forgetting means that you forgot the value of a proposition, or the assignment of two values from different sets of objects to each other. In the case of a bankcard you the four-number code that you forgot is just one one 10,000 options, so previous it was true that $K0000 \vee \ldots \vee K9999$ whereas currently we have that $\neg(K0000 \vee \ldots \vee K9999)$. (Let 0000 stand for the proposition that your pin number is 0000, etc.) Similarly, somewhat simplifying matters, the finite number of all humans only have a finite, somewhat smaller, number of names. An atomic proposition about your office keys is also a feature namely with two values only, true and false. The multiple-valued features can also be modelled as a *number* of atomic propositions; this can be done in a very uneconomic fashion as above, but also in a minimal way. We conclude that this sort of forgetting is like the previous kind.

Defaulting on obligations But there are other kinds of forgetting too. For example, say I forgot to pick you up at the airport at 4:30 PM. Forgetting an action is very different from forgetting a proposition. Forgetting an action amounts to defaulting on an obligation and the *observation* of having forgotten it is not at all related to ignorance. It points backwards in time to the moment when you were not *aware* of the obligation. Obligations can be modelled with deontic logics. We will not be concerned with this kind of forgetting.

Multi-agent versions of forgetting In a multi-agent setting additional, interactive, ways of forgetting crop up as well. Some of the above have group versions. For example, Will Smith only had to flash a whole group once, not each of its members individually. And if you have been flashed, although *you* don't know that you knew about the green ooze, Will Smith knows that you knew. So in a multi-agent setting some aspects of 'completely forgetting' can be modelled. When assuming standard notions of knowledge, that is introspective, we now run straight into trouble of another kind.

A group version for 'remembering prior knowledge' is hard to justify, because its interpretation typically involves introspection: you forgot something if *you are aware of* (in the sense of 'you know') previous knowledge and present ignorance of it. A setting wherein a group is collectively aware of its prior (common) knowledge is somewhat harder to imagine. It makes more sense to have a version of 'remembering prior knowledge' for individuals in a group, be-

[1]Emails of the authors: hans@cs.otago.ac.nz, herzig@irit.fr, lang@irit.fr, marquis@cril.univ-artois.fr. Hans van Ditmarsch is corresponding author. This document concerns work in progress. Hans van Ditmarsch acknowledges the support of the Netherlands Institute of Advanced Study where he was Lorentz Fellow in 2008.

cause they can inform and are observed by others: here you standing in front of your office door again now in company of four freshmen students, “Ohmigod, I forgot again which is my office key!”

For yet other multi-agent examples: I can notice that you forgot to pick me up at the airport, or that you no longer appear to know the way around town. The last may even be without me being aware of my ignorance. I may have forgotten whether you knew about a specific review result for our jointly editored journal issue. In other words, previously $K_{me}K_{you}$*accept* or $K_{me}K_{you}\neg$*accept* but currently $\neg K_{me}K_{you}$*accept* and $\neg K_{me}K_{you}\neg$*accept*. Some meaningful propositions that can be forgotten in a multi-modal context are therefore modal.

1.1. Forgetting and progression

In theory change (belief revision) the operation of forgetting is a form of *belief contraction*. Given prior belief in p or its negation, we want to remove that from the set of believed formulas including all its dependencies. In artificial intelligence this has become a search for efficient ways to implement such a contraction. The following way to model / implement forgetting an atomic proposition has recently been proposed [5]. Given a set of formulas (‘theory’) Φ, we compute the effect of forgetting information about atom p by a binary operation

$$\mathit{Fg}(\Phi, p) := \{\varphi(\top/p) \vee \varphi(\bot/p) \mid \varphi \in \Phi\}$$

Here, $\varphi(\psi/p)$ is the replacement of all (possibly zero) occurrences of p in φ by ψ. This proposal can be called the (syntactic) *progression* of Φ by the function *Fg*, relative to the forgotten information about p. It is well-known that this way to model forgetting as progression does not work for *modal* formulas. For example, if the agent already does not know whether p, surely that should remain the case after forgetting the value of p. But we now have that (write $\mathit{Fg}(\varphi, p)$ for $\mathit{Fg}(\{\varphi\}, p)$):

$$\begin{array}{l} \mathit{Fg}(\neg Kp \wedge \neg K\neg p, p) \\ = \\ (\neg K\top \wedge \neg K\neg\top) \vee (\neg K\bot \wedge \neg K\neg\bot) \\ \Leftrightarrow \\ (\neg\top \wedge \neg\bot) \vee (\neg\bot \wedge \neg\top) \\ \Leftrightarrow \\ \bot \end{array}$$

For another example of an undesirable feature, it is also not the case that knowledge of p is transformed into ignorance about p by this procedure:

$$\mathit{Fg}(Kp, p) \leftrightarrow (K\top \vee K\bot) \leftrightarrow \top.$$

In other words, this approach does not lead anywhere for modal formulas. Surely, one would like that $\neg Kp \wedge \neg K\neg p$ is true after forgetting the value of p, even when this was not true initially. For any theory Φ and atom p, the result of forgetting p should entail ignorance about p:

$$\mathit{Fg}(\Phi, p) \models \neg Kp \wedge \neg K\neg p.$$

The difficulties in obtaining this result by theory revision motivated us to model forgetting as an event in a dynamic epistemic logic.

1.2. Forgetting or no-forgetting, that’s the question

We model the action of forgetting an atomic proposition p as an *event* $\mathit{Fg}(p)$ (in its sense of remembering prior knowledge about p). We do this in a propositional logic expanded with an epistemic modal operator K and a dynamic modal operator $[\mathit{Fg}(p)]$ (including multi-agent versions). As usual, $K\varphi$ describes that the agent knows φ. Formula $[\mathit{Fg}(p)]\varphi$ means that after the agent forgets his knowledge about p, φ is true.

The obvious *precondition* for event $\mathit{Fg}(p)$ is prior knowledge of the value of p: $Kp \vee K\neg p$. The obvious *postcondition* for event $\mathit{Fg}(p)$ is ignorance of the value of p: $\neg Kp \wedge \neg K\neg p$. In other words

$$(Kp \vee K\neg p) \rightarrow \langle \mathit{Fg}(p)\rangle(\neg Kp \wedge \neg K\neg p)$$

should be valid in the information state prior to the event of forgetting ($\langle \mathit{Fg}(p)\rangle$ is the diamond version of $[\mathit{Fg}(p)]$). Or, abstracting from that precondition, it should be valid that:

$$[\mathit{Fg}(p)](\neg Kp \wedge \neg K\neg p).$$

Wasn’t dynamic epistemic logic supposed to satisfy the principle of ‘no forgetting’? So how on earth can one model forgetting in this setting? We can, because we cheat. ‘No forgetting’ (a.k.a. ‘perfect recall’) means that if states s and t resulting from the execution of (possibly different) events are indistinguishable, then the states before the execution of these respective events are also indistinguishable. If after the event of forgetting I am ignorant about p I cannot distinguish a p-state from a $\neg p$-state. Therefore, because of the principle of ‘no forgetting’, I should already have been unable to distinguish these states before the execution of this supposed event $\mathit{Fg}(p)$... I should have been already ignorant about p before... We solve this dilemma by the standard everyday solution of forgetful people: blame others for your forgetfulness. In this case, we blame the world, i.e., the state of the world: we *simulate* forgetting by *nondeterministically changing the value of p in the actual or other states, in a way unobservable by the agent*. Thus resulting in his ignorance about p. Note that this solution is different from how belief revision is modelled in dynamic epistemic logic: prior belief in p that is revised with $\neg p$ and results in belief in $\neg p$ is standardly modelled by considering this a 'soft' or defeasible form of belief (i.e., not knowledge) and implemented by changing a preference relation between states [11, 3].

Once we have the above, the relation with theory progression becomes clear. Let (M, s) be an information state (pointed Kripke model with designated state) for the theory Φ. Suppose we want to know if $Fg(\Phi, p) \models \psi$. The artificial intelligence community is particularly interested in efficient ways to perform such computations. Well, if it is the case, we then should also have $M, s \models [Fg(p)]\psi$. We propose a $[Fg(p)]$-operator that can be reduced (eliminated): there is an epistemic formula χ such that $[Fg(p)]\psi \leftrightarrow \chi$. There are fast and efficient algorithms to determine the truth of an epistemic formula in a given Kripke model: $M, s \models \chi$ can be done quickly. This answers the question whether $Fg(\Phi, p) \models \psi$. There is one drawback: the reduction of $[Fg(p)]\psi$ to a formula without reference to an event that should be initially true is known as regression. But in AI we also want to do progression: compute some Φ' from $Fg(\Phi, p)$. This is harder, also in a dynamic epistemic logical context, and we have no answer to that, although a few suggestions.

Expanding our perspective This contribution focusses on a clean solution on how to model an event $Fg(p)$ satisfying $(Kp \vee K\neg p) \rightarrow \langle Fg(p)\rangle(\neg Kp \wedge \neg K\neg p)$. From there on, the modelling desiderata diverge. There are many interesting options.

Is our perspective that of a modelling observer, in which case we might require that forgetting is an information-changing event only, so that the value of p in the actual state should not change? Or is our perspective that of an agent in the system, so that we are only considering the value of local propositions, i.e., formulas of the form $K\varphi$? Whether we simulate the desiderata by factual change inducing informational change does not matter in that case.

Are we talking about one or about all agents forgetting? How about forgetting epistemic propositions?

Our solution presumes the interpretation of $K\varphi$ as 'the agent *knows* φ' and, correspondingly, even though our semantics is general all our examples are of $S5$-structures. There are obvious slightly weaker modellings of forgetting to model introspective belief (to be interpreted on $KD45$-structures.

From the perspective of the agent we also want to look backwards. Let $Fg(p)^-$ be the converse of $Fg(p)$ (e.g. in the sense of [1, 15, 8]). All the time we are saying that an agent that has forgotten about p remembers prior knowledge of p. This we can now express as the validity of

$$K(\neg Kp \wedge \neg K\neg p \wedge \langle Fg(p)^-\rangle(Kp \vee K\neg p))$$

in the information state after the event of forgetting. (Note the different perspective from before, where our perspective was the information state before the act of forgetting.) In other words: the agent is aware of its current ignorance and its previous knowledge. We will indicate some ways to address this in the further research section at the end of our contribution. There is one main drawback of this approach: there is no way to reduce expressions with converse events to purely epistemic formulas. So, the advantage of dynamic epistemic logic for regression questions in AI has not been reached there (yet).

2. Language and semantics

Language Our language is

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid Fg_B(p)$$

In the single-agent context write $K\varphi$ for $K_a\varphi$ and $Fg(p)$ for $Fg_a(p)$.

Later on, in a multi-agent context, we write $Fg(p)$ for $Fg_A(p)$, and we also distinguish the converse ('remember') operator $Fg_B(p)^-$. For the forgetting of (not necessarily atomic) formulas φ we write $Fg_B(\varphi)$.

Structures Our structures are pointed Kripke models $(S, R, V), s)$ (with $R : A \rightarrow \mathfrak{P}(S \times S)$ and $V : P \rightarrow \mathfrak{P}(S)$) and multiple-pointed event models ('action models'). Our typical example structures are $S5$ to model knowledge and knowledge change and for $KD45$ to model belief and belief change.

The dynamic structures are event models, i.e., action models including assignments of atoms (a.k.a. substitutions) [10, 13]. We follow notational conventions as in [13]: if in event s the precondition is φ and the postcondition is that the valuation of atom p becomes that of ψ, we write: in s: if φ then $p := \psi$.

We visualize $S5$ models by linking states that are indistinguishable for an agent, possibly labelling the link with the agent name (not in the single-agent situation). Transitivity is assumed. In pictures of event models: a formula next to an event is its precondition, an assignment next to it a postcondition.

Semantics Assume an epistemic model $M = (S, R, V)$.

$M, s \models p$	iff	$s \in V(p)$
$M, s \models \neg\varphi$	iff	$M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	iff	$M, s \models \varphi$ and $M, s \models \psi$
$M, s \models K_a\varphi$	iff	for all $t \in S : (s, t) \in R_a$ implies $M, t \models \varphi$
$M, s \models [Fg_B(p)]\psi$	iff	for all $M', s' : (M, s)[\![Fg_B(p)]\!](M', s')$ implies $M', s' \models \varphi$

where $[\![Fg_B(p)]\!]$ is a binary relation between pointed epistemic states, as usual for the interpretation of events in dynamic epistemic logic. Of course, we model the execution of an event $Fg_B(p)$ as a restricted modal product and this will be the relation induced by that. In the next section we will define the event model $Fg_B(p)$. The set of validities in our logic is called FG.

3. Forgetting

In a single-agent setting we model forgetting as the non-deterministic event where the (anonymous) agent is uncertain which of two assignments have taken place: p becomes true, or p becomes false. Formally, this is a non-deterministic event model consisting of two events 0 and 1 that are indistinguishable for the agent, and such that $\mathsf{pre}(0) = \mathsf{pre}(1) = Kp \vee K\neg p$, $\mathsf{post}(0)(p) = \bot$, and $\mathsf{post}(1)(p) = \top$. We can visualize this event model $Fg(p)$ as follows (postconditions above, preconditions below actions):

$$\begin{array}{ccc} p := \top & & p := \bot \\ 1 & \rule[0.5ex]{4cm}{0.4pt} & 0 \\ Kp \vee K\neg p & & Kp \vee K\neg p \end{array}$$

The event model $Fg(p)$ is non-deterministic choice between two deterministic events $(Fg(p), 1)$ and $(Fg(p), 0)$. For the interpretation of such a pointed event we use the standard semantics of 'action models', for events/points $i = 0, 1$ (and we recall the equivalence $[Fg(p)]\psi \leftrightarrow ([Fg(p), 0]\psi \wedge [Fg(p), 1]\psi)$:

$$\begin{array}{lll} M, s \models [Fg(p), i]\varphi & \text{iff} & M, s \models Kp \vee K\neg p \text{ implies} \\ & & M \otimes Fg(p), (s, i) \models \varphi \end{array}$$

In the language we'd like to avoid directly referring to the pointed versions (out of some possibly mistaken sense of minimalism), and therefore introduce the pointed versions of forgetting by abbreviation (and this amounts indeed to the same):

$$\begin{array}{lcl} \langle Fg(p), 0\rangle\varphi & = & \langle Fg(p)\rangle(\neg p \wedge \varphi) \\ \langle Fg(p), 1\rangle\varphi & = & \langle Fg(p)\rangle(p \wedge \varphi) \end{array}$$

To obtain a complete axiomatization for FG we can simply apply the reduction axioms for event models, as specified in [13]. This is the axiomatization **FG** in Table 1. Note that from the above abbreviation also follows that $[Fg(p)](p \rightarrow \varphi)$ is equivalent to $[Fg(p), 1]\varphi$, and that $[Fg(p)](\neg p \rightarrow \varphi)$ equals $[Fg(p), 0]\varphi$.

Proposition 3.1 *Axiomatization* **FG** *is sound and complete.*

$$\begin{array}{lcl} [Fg(p)]p & \leftrightarrow & \neg(Kp \vee K\neg p) \\ [Fg(p)]q & \leftrightarrow & (Kp \vee K\neg p) \rightarrow q \;\text{ for } q \neq p \\ [Fg(p)]\neg\varphi & \leftrightarrow & (Kp \vee K\neg p) \rightarrow \\ & & (\neg[Fg(p)](\neg p \rightarrow \varphi) \wedge \neg[Fg(p)](p \rightarrow \varphi)) \\ [Fg(p)](\varphi \wedge \psi) & \leftrightarrow & [Fg(p)]\varphi \wedge [Fg(p)]\psi \\ [Fg(p)]K\varphi & \leftrightarrow & (Kp \vee K\neg p) \rightarrow K[Fg(p)]\varphi \end{array}$$

Table 1. Axiomatization FG—only reduction rules involving Fg are presented

Proof. We show that the axiomatization resulted from application of the reduction axioms in action model logic by Baltag et al. [2], by applying, case by case, the standard reduction rules for event models. This kills two birds (soundness and completeness) at one throw.

Case p.
$[Fg(p)]p$
$\Leftrightarrow$
$[Fg(p), 0]p \wedge [Fg(p), 1]p$
$\Leftrightarrow$
$(\mathsf{pre}(0) \rightarrow \mathsf{post}(0)(p)) \wedge (\mathsf{pre}(1) \rightarrow \mathsf{post}(1)(p))$
$\Leftrightarrow$
$((Kp \vee K\neg p) \rightarrow \bot) \wedge ((Kp \vee K\neg p) \rightarrow \top)$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow \bot$
$\Leftrightarrow$
$\neg(Kp \vee K\neg p)$

In other words, it is not the case that p is true after every execution of $Fg(p)$.

Case q.
$[Fg(p)]q$
$\Leftrightarrow$
$[Fg(p), 0]q \wedge [Fg(p), 1]q$
$\Leftrightarrow$
$(\mathsf{pre}(0) \rightarrow \mathsf{post}(0)(q)) \wedge (\mathsf{pre}(1) \rightarrow \mathsf{post}(1)(q))$
$\Leftrightarrow$ $\mathsf{pre}(0) = \mathsf{pre}(1) = Kp \vee K\neg p, \mathsf{post}(0)(q) = \mathsf{post}(q)(1) = q$
$(Kp \vee K\neg p) \rightarrow q$

This axiom expresses that, if $Fg(p)$ is executable, the value of atoms q other than p remains the same.

Case $\neg\varphi$.
$[Fg(p)]\neg\varphi$
$\Leftrightarrow$
$[Fg(p), 0]\neg\varphi \wedge [Fg(p), 1]\neg\varphi$
$\Leftrightarrow$

$(\mathsf{pre}(0) \rightarrow \neg[Fg(p),0]\varphi) \wedge (\mathsf{pre}(1) \rightarrow \neg[Fg(p),1]\varphi)$
$\Leftrightarrow$ $\mathsf{pre}(0) = \mathsf{pre}(1) = Kp \vee K\neg p$
$(Kp \vee K\neg p) \rightarrow (\neg[Fg(p),0]\varphi \wedge \neg[Fg(p),1]\varphi)$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow (\neg[Fg(p)](\neg p \rightarrow \varphi) \wedge \neg[Fg(p)](p \rightarrow \varphi))$

Note that the expression on the right is *not* equivalent to $\neg[Fg(p)]\varphi$. It would be if the conjunction in the middle had been a disjunction.

Case $\varphi \wedge \psi$.
$[Fg(p)](\varphi \wedge \psi)$
$\Leftrightarrow$
$[Fg(p),0](\varphi \wedge \psi) \wedge [Fg(p),1](\varphi \wedge \psi)$
$\Leftrightarrow$
$[Fg(p),0]\varphi \wedge [Fg(p),0]\psi \wedge [Fg(p),1]\varphi \wedge [Fg(p),1]\psi$
$\Leftrightarrow$
$[Fg(p)]\varphi \wedge [Fg(p)]\psi$

Case $K\varphi$.
$[Fg(p)]K\varphi$
$\Leftrightarrow$
$[Fg(p),0]K\varphi \wedge [Fg(p),1]K\varphi$
$\Leftrightarrow$
$(\mathsf{pre}(0) \rightarrow K[Fg(p)]\varphi) \wedge (\mathsf{pre}(1) \rightarrow K[Fg(p)]\varphi)$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow K[Fg(p)]\varphi$
QED

Proposition 3.2 *The formula* $[Fg(p)](\neg Kp \wedge \neg K\neg p)$ *is valid and derivable.*

Proof. Validity is trivial. Thus we have derivability. It is instructive to see part of the derivation. We apply the reduction rules in the axiomatization **FG**.

$[Fg(p)](\neg Kp \wedge \neg K\neg p)$
$\Leftrightarrow$
$[Fg(p)]\neg Kp \wedge [Fg(p)]\neg K\neg p$

Left conjunct of previous line:
$[Fg(p)]\neg Kp$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow (\neg[Fg(p),0]Kp \wedge \neg[Fg(p),1]Kp)$
$\Leftrightarrow$
$((Kp \vee K\neg p) \rightarrow \neg[Fg(p),0]Kp) \wedge ((Kp \vee K\neg p) \rightarrow \neg[Fg(p),1]Kp)$

Again, left conjunct of previous line:
$(Kp \vee K\neg p) \rightarrow \neg[Fg(p),0]Kp$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow \neg((Kp \vee K\neg p) \rightarrow K[Fg(p)]p$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow \neg((Kp \vee K\neg p) \rightarrow K\neg(Kp \vee K\neg p))$
$\Leftrightarrow$
$(Kp \vee K\neg p) \rightarrow \neg\bot$
$\Leftrightarrow$
$\top$

All together we have four cases (conjuncts), of which have now done one. The four cases are similar. QED

Proposition 3.3 $[Fg(p)][Fg(p)]\varphi$ *is valid.*

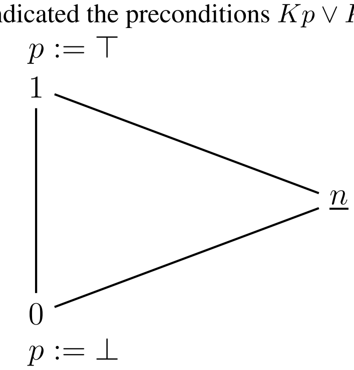
Proof. Assume the first $Fg(p)$ can be executed. Then the precondition $Kp \vee K\neg p$ was satisfied in the initial model. After the execution of that $Fg(p)$, we have $\neg(Kp \vee K\neg p)$. Therefore the second $Fg(p)$ cannot be executed. (So, trivially, any postcondition φ of that is then true.) QED

Unlike in real life, you cannot forget something *twice*. After you have forgotten it the first time, you have to be informed again about p and only then you can forget it again. Maybe that's quite a bit like real life after all.

Progress towards seeing this modelling of forgetting as *progression* in the AI sense would be made if we were to prove that $\psi \rightarrow [Fg(p)]\psi$ is valid for all ψ that do not contain occurrences of p. We think this is valid, and it may even be trivial, but we haven't given it sufficient attention yet.

4. Forgetting without changing the real world

An unfortunate side effect of our modelling of forgetting is that the actual value of p gets lost in the process. This is 'somewhat strange' if we only want to model that the agents forget the value of p, but that 'otherwise' nothing changes: the real value of p should then be unchanged. We can overcome that deficiency in the alternative modelling (Fg, n). It is very much like $Fg(p)$ except that there is one more alternative event in the model, indistinguishable from the other two, that represents the event 'nothing happens' except that the truth of p should be known (its precondition is $Kp \vee K\neg p$ and there are no postconditions). Also, unlike $Fg(p)$, the alternative (Fg, n) is pointed: this event model is deterministic, the real event is event n. This ensures that the real value of p does not change. In the figure we have not indicated the preconditions $Kp \vee K\neg p$.



The reduction rules for (Fg, n) are the same as for Fg except for the atomic case p and for negation, where:

$$\begin{array}{lcl} [\mathsf{Fg}(p), n]p & \leftrightarrow & (Kp \vee K\neg p) \rightarrow p \\ [\mathsf{Fg}(p), n]\neg\varphi & \leftrightarrow & (Kp \vee K\neg p) \rightarrow (\neg[\mathsf{Fg}(p), 0]\varphi \wedge \\ & & \neg[\mathsf{Fg}(p), 1]\varphi \wedge \neg[\mathsf{Fg}(p), n]\varphi) \end{array}$$

We can introduce $(\mathsf{Fg}(p), 0)$ and $(\mathsf{Fg}(p), 1)$ by abbreviation, somewhat different from before. We now have the interesting results that

Proposition 4.1 *Valid are (proof omitted):*

$$\begin{array}{ll} \psi \rightarrow [\mathsf{Fg}, n]\psi & \textit{for boolean } \psi \\ [\mathsf{Fg}, n]K\psi \leftrightarrow [Fg]K\psi & \textit{for any } \psi \end{array}$$

In other words: from the perspective of the agent, the different modellings of forgetting are indistinguishable. That makes the simpler modelling $Fg(p)$ preferable over the slightly more complex $(\mathsf{Fg}(p), n)$.

5. Further research and variations

In this section we present some less developed lines of research.

5.1. Forgetting by bisimulation quantification

A not strictly modal way to model forgetting is to see forgetting p as universal bisimulation quantification over p (as in [Hollenberg] [Visser]), i.e.:

$$[Fg^{\forall}(p)]\varphi := \forall p\varphi$$

where

$$\begin{array}{l} M, s \models \forall p\varphi \\ \Leftrightarrow \\ \text{for all } (M', s') \text{ such that } (M', s') \leftrightarroweq_{P-p} (M, s) : (M', s') \models \varphi \end{array}$$

The notation $(M', s') \leftrightarroweq_{P-p} (M, s)$ means that epistemic state (M', s') is bisimilar to epistemic state (M, s) with respect to the set P of all atoms *except* p. In other words the valuation of p may vary 'at random'. This includes the models constructed by $Fg(p)$ and by $(\mathsf{Fg}(p), n)$ from that given M. That is, for any epistemic model M we have that

$$\begin{array}{lll} M & \leftrightarroweq_{P-p} & M \otimes Fg(p) \\ M & \leftrightarroweq_{P-p} & M \otimes \mathsf{Fg}(p) \end{array}$$

from which follow the validities

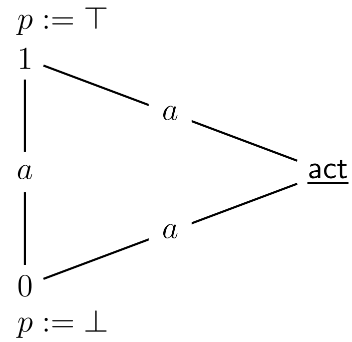
$$\begin{array}{l} [Fg^{\forall}(p)]\psi \rightarrow [Fg(p)]\psi \\ [Fg^{\forall}(p)]\psi \rightarrow [\mathsf{Fg}(p), n]\psi \end{array}$$

The axiomatizations of such bisimulation quantified logics are often complex; for $S5$ models they behave somewhat better [4]. We treat a bisimulation quantification operation non-standardly as 'some sort of dynamic modal operator' here. We justify this because it is a model changing operation. This perspective is also explored in [12].

Although theoretically an interesting alternative, the much simpler $Fg(p)$ and $(\mathsf{Fg}(p), n)$ seem to be preferable for computational results. However, the bisimulation version may have other advantages we are currently unaware of.

5.2. Single agent forgetting in a multi-agent context

Suppose a single agent says 'I forgot p' in the presence of others. This can be modelled by the multi-agent event model $Fg_a(p)$, where, again, all preconditions are $K_a p \vee K_a \neg p$.



The visualization means that all agents *except* a can distinguish between the three alternatives. (So for all agents in $A - a$ the accessibility relation is the identity on the domain.) In this case, a similar two-event model (as in the single-agent approach) would not suffice (said to be irrelevant from agent a's point of view), as we also have to take the other agents into consideration: surely, we don't want them do doubt the value of p all of a sudden.

Again, there is an obvious complete axiomatization applying the reduction rules for event models, and we have the validity

$$[Fg_a(p)]C_A(\neg K_a p \wedge \neg K_a \neg p)$$

where C_A stands for 'common knowledge among group A.'

5.3. Remembering that you have forgotten

To remember in the object language that you have forgotten something requires a language allowing

$$K(\neg Kp \wedge \neg K\neg p \wedge \langle Fg(p)^{-}\rangle(Kp \vee K\neg p))$$

By instead of pointed Kripke models taking what is known as the 'forest' produced by the initial model and all possible sequences of all $Fg(p)$ events (for all atoms) (see [9] and various other publications, this relates strongly to the history-based approaches by Parikh & Ramanujam [7], and later Pacuit [6], and others [14]), we get a model that allows us to refer to past events (à la [15] and [8] — Sack's approach is also properly based on histories). We can now combine this recent strand of research, with another strand of adding assignments to the language, as we already did, and additionally to that we can add theories for event models using converse actions, as done in [1] and also outlined in, e.g. [9]. This not so grand but nevertheless not yet realized scheme leads somewhere, namely to a complete axiomatization, but very likely not to the desirable result that expressions containing event operators (converse or not) can be reduced to epistemic formulas. So from an AI point of view, this is probably not a productive point of view. From a cognitive modelling point of view, it is of course interesting as we can refer to previous knowledge. (Also note that, unlike the typical counterexamples in [15], in this case the agent *knows* that prior to the current situation he/she had knowledge of p—absence of that created the problems with finding reduction axioms. So within this restricted setting of specific events, maybe more useful can be done... To be continued.

5.4. Other matters

Modeling the forgetting of features with multiple values can be done by a simple adjustment of the above proposals. This is easy. How to model the forgetting modal formulas is a different piece of cake altogether; in this case we have made no progress yet.

References

[1] G. Aucher. *Perspectives on belief and change*. PhD thesis, University of Otago & Institut de Recherche en Informatique de Toulouse, New Zealand & France, 2008.

[2] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, pages 43–56, 1998.

[3] A. Baltag and S. Smets. Dynamic belief revision over multi-agent plausibility models. Proceedings of LOFT 2006 (7th Conference on Logic and the Foundations of Game and Decision Theory), 2006.

[4] T. French. *Bisimulation quantifiers for modal logic*. PhD thesis, University of Western Australia, 2006.

[5] J. Lang, P. Liberatore, and P. Marquis. Propositional independence: Formula-variable independence and forgetting. *J. Artif. Intell. Res. (JAIR)*, 18:391–443, 2003.

[6] E. Pacuit. Some comments on history-based structures. To appear in Journal of Applied Logic, 2007.

[7] R. Parikh and R. Ramanujam. Distributed processing and the logic of knowledge. In *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 256–268. Springer, 1985. A newer version appeared in *Journal of Logic, Language and Information*, vol. 12, 2003, pp. 453–467.

[8] Y. Sack. *Adding Temporal Logic to Dynamic Epistemic Logic*. PhD thesis, Indiana University, Bloomington, USA, 2007.

[9] J.F.A.K. van Benthem, J.D. Gerbrandy, and E. Pacuit. Merging frameworks for interaction: DEL and ETL. In D. Samet, editor, *Proceedings of TARK 2007*, pages 72–81, 2007.

[10] J.F.A.K. van Benthem, J. van Eijck, and B.P. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[11] H.P. van Ditmarsch. Prolegomena to dynamic logic for belief revision. *Synthese (Knowledge, Rationality & Action)*, 147:229–275, 2005.

[12] H.P. van Ditmarsch and T. French. Simulation and information. (Electronic) Proceedings of LOFT 2008, Amsterdam, 2008.

[13] H.P. van Ditmarsch and B.P. Kooi. Semantic results for ontic and epistemic change. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Postproceedings of LOFT 2006*. Amsterdam University Press, 2008. To appear in the series Texts in Logic and Games.

[14] H.P. van Ditmarsch, J. Ruan, and W. van der Hoek. Model checking dynamic epistemics in branching time. In *(Informal) Proceedings of FAMAS 2007, Durham UK*, 2007.

[15] A. Yap. Product update and looking backward. Technical report, University of Amsterdam, 2006. ILLC Research Report PP-2006-39.