Introspective forgetting

Hans van Ditmarsch[12⋆], Andreas Herzig[2], Jérôme Lang[2], and Pierre Marquis[3]

[1] Computer Science, University of Otago, New Zealand, hans@cs.otago.ac.nz
[2] IRIT, Université Paul Sabatier, France, {hans,herzig,lang}@irit.fr
[3] CRIL, Université d'Artois, France, marquis@cril.univ-artois.fr

Abstract. We model the forgetting of propositional variables in a modal logical context where agents become ignorant and are aware of each others' or their own resulting ignorance. The resulting logic is sound and complete. It can be compared to variable-forgetting as abstraction from information, wherein agents become unaware of certain variables: by employing elementary results for bisimulation, it follows that beliefs not involving the forgotten atom(s) remain true.

Keywords: modal logic, forgetting, abstraction, action logic, belief change

1 There are different ways of forgetting

Becoming unaware In the movie 'Men in Black', Will Smith makes you forget knowledge of extraterrestrials 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 do 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 this means that parameters of the language, such as the set of atoms, shrink.

Becoming ignorant 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. Is it the bigger or the smaller key? This is about forgetting the value of an atomic proposition p—such as "the bigger key fits the door." You are embarrassingly aware of your current ignorance: introspection is involved. We have $K(\neg Kp \land \neg K \neg p)$. This sort of forgetting is central to our concerns.

Remembering prior knowledge You also remember that you *knew* which key it was. You just forgot. Previously Kp or $K\neg p$, and only now $\neg Kp$ and $\neg K \neg p$.

⋆ Hans van Ditmarsch is corresponding author.

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. An atomic proposition about your office keys is a feature with two values only, true and false. The (finitely) multiple-valued feature can be modelled as a *number* of atomic propositions. Forgetting of such multiple boolean variables is in our approach similar to forgetting a single boolean variable.

Multi-agent versions of forgetting Will Smith only had to flash a whole group once, not each of its members individually. So, in a multi-agent setting some aspects of collectively 'becoming unaware' can be modelled. A different, familiar phenomenon is that of an individual becoming unaware in a group: "You forgot my birthday, *again*!"

A group version for 'remembering prior knowledge' would involve common awareness, and prior common knowledge. This collective introspective character is not always easy to justify. On the other hand, a version of 'remembering prior knowledge' for *individuals* in a group is more intuitive, because they can inform and are observed by others: here you are standing in front of your office door again, now in company of four freshmen students, "Ohmigod, I forgot again which is my office key!"

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 themselves modal.

Forgetting events Say I forgot to pick you up at the airport at 4:30 PM. Forgetting an *action* (event) is different from forgetting a *proposition*. 'Forgetting of events' amounts to introducing *temporal uncertainty* in the model, apart from epistemic uncertainty. The *observation* of having forgotten it, is about the recovery that takes place after forgetting the event.

2 Motivation

A short history of forgetting in AI In 'Forget it!' [1] Lin and Reiter proposed a way to abstract from ground atoms (that can be equated to propositional variables) in a set of first-order beliefs, employing the notion of *similarity of models for a theory except for such a ground atom*. They leave it open whether such forgetting is the result of an agent consciously updating a knowledge base after having learnt about *factual* change, or whether this is simple erosion of her working memory, purely *epistemic* change. Their work was built upon by Lang, Liberatore and Marquis with their in-depth study on the computational costs of transforming theories by variable forgetting [2], or rather the costs of determining the independence of parts of a theory from specific variables. In [3] Baral and Zhang took part of this battleground to involve more explicit operators for knowledge and belief, where the result of an agent forgetting a variable results in her (explicit) ignorance of that variable's value, and in [4], in progress, Zhang and Zhou make an original and interesting backtrack to the ideas of [1] by suggesting *bisimulation invariance except for the forgotten variable*, in order to model forgetting. Forgetting has been generalized to logic programs in [5–7] and to description logics in [8]. Forgetting of (abstracting from) actions in planning has been investigated in [9].

Progression of belief sets Generalizing the results in [2] to forgetting in *positive* epistemic formulas (subformulas expressing ignorance are not allowed) is easy [10], but beyond that it is hard. Consider the binary operation

$$Fg(\Phi, p) \equiv_{\mathsf{def}} \{\varphi(\top/p) \vee \varphi(\bot/p) \mid \varphi \in \Phi\}$$

wherein $\varphi(\psi/p)$ is the replacement of all (possibly zero) occurrences of p in φ by ψ. This defines the (syntactic) *progression* of Φ when forgetting about p. When applying this recipe to formulas expressing ignorance, we get undesirable consequences, e.g. that $Fg(\neg Kp \wedge \neg K\neg p, p)$ is equivalent to the contradiction $\bot$. This is shown as follows:

$Fg(\neg Kp \wedge \neg K\neg p, p)$
is by definition
$(\neg K\top \wedge \neg K\neg\top) \vee (\neg K\bot \wedge \neg K\neg\bot)$
iff
$(\neg\top \wedge \neg\bot) \vee (\neg\bot \wedge \neg\top)$
iff
$\bot$

Such problems motivated us to model forgetting as an event in a dynamic epistemic logic.

Forgetting as a dynamic modal operator We model the action of forgetting an atomic proposition p as an *event* $Fg(p)$. We do this in a propositional logic expanded with an epistemic modal operator K and a dynamic modal operator $[Fg(p)]$, with obvious multiple-value and multi-agent versions. Formula $[Fg(p)]\varphi$ means that after the agent forgets his knowledge about p, φ is true. We call $[Fg(p)]$ a *dynamic* modal operator because it is interpreted by a state transformation, more particularly: by changing an information state that is represented by a pointed Kripke model (M, s) into another information state (M', s'). The relation to the theory transforming operation $Fg(\Phi, p)$ is as follows: for all models (M, s) of Φ, $[Fg(p)]\varphi$ should be true in (M, s) if and only if $\varphi \in Fg(\Phi, p)$.

A *precondition* for event $Fg(p)$ seems prior knowledge of the value of p: $Kp \vee K\neg p$. How can you forget something unless you know it in the first place? To make our approach comparable to variable forgetting in the 'abstracting-from-information'-sense, we do not require prior knowledge as a precondition for forgetting. The obvious *postcondition* for event $Fg(p)$ is ignorance of the value of p: $\neg Kp \wedge \neg K\neg p$. It should therefore be valid that

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

Forgetting or no-forgetting? On ontic and epistemic change Wasn't dynamic epistemic logic supposed to satisfy the principle of 'no forgetting' (a.k.a. 'perfect recall')? This entails that positive knowledge such as factual knowledge Kp and $K\neg p$, is preserved after any event. Or, dually: if you are ignorant about p now, then you must have been ignorant about p before. So how on earth can one model forgetting in this setting? We can, because we cheat. We solve this dilemma by the standard everyday solution of forgetful people: blame others. In this case: blame the world; we *simulate* forgetting by *changing the value of* p *in the actual or other states, in a way known to be unobservable by the agent*. Thus resulting in her ignorance about p.[4]

Having cheated in that way, our logic is equivalent to one without actual change of facts in the one and only way that counts: it makes no difference for believed formulas, i.e., for expressions of the form $K\varphi$.

Remembering prior knowledge To express that an agent recalls prior knowledge we have to be able to refer to past events. Let $Fg(p)^-$ be the converse of $Fg(p)$ (e.g. in the sense of [13–15]). We can now express prior knowledge of now forgotten variables as

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

This stands for: "the agent knows that (she does not know p and she does not know $\neg p$ and before forgetting about p she either knew p or knew $\neg p$). We will outline our progress towards modelling this in the concluding section.

3 A logic of propositional variable forgetting

We present a single agent and single variable version of the logic *only*. All results trivially generalize to multiple agents and multiple values (see page 10).

Language, structures and semantics Given is a set P of propositional variables.

Definition 1 (Language and structures). *Our language* $\mathcal{L}$ *is*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid [Fg(p)]\varphi$$

where $p \in P$*, and our structures are* pointed Kripke models $((S, R, V), s)$*, with* $R \subseteq (S \times S)$*,* $V : P \rightarrow \mathcal{P}(S)$*, and* $s \in S$*.*

[4] This is different from how belief revision is modelled in dynamic epistemic (doxastic) 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, 12].

If $P' \subseteq P$, then $\mathcal{L}(P')$ is the language restricted to P'. The diamond versions of our modal operators are defined as $\hat{K}\varphi \equiv_{\mathsf{def}} \neg K \neg\varphi$ and $\langle Fg(p)\rangle\varphi \equiv_{\mathsf{def}} \neg[Fg(p)]\neg\varphi$. The structures are typically $S5$ to model knowledge and $KD45$ to model belief—but this is not a requirement.

The dynamic operator $[Fg(p)]$ is relative to the state transformer $Fg(p)$ that is an *event model*. The pointed Kripke models are static structures, encoding knowledge and belief, and the event models are dynamic structures, encoding *change of* knowledge and belief. Formally, *multiple-pointed event models* (a.k.a. action models) are structures $(\mathsf{M}, \mathsf{S}') = ((\mathsf{S}, \mathsf{R}, \mathsf{pre}, \mathsf{post}), \mathsf{S}')$, where $\mathsf{S}' \subseteq \mathsf{S}$, where $\mathsf{pre} : \mathsf{S} \to \mathcal{L}$ assigns to each event $\mathsf{s} \in \mathsf{S}$ a *precondition* and where $\mathsf{post} : \mathsf{S} \to (P \to \mathcal{L})$ assigns to each event a *postcondition* (a.k.a. assignment) for each atom (of a *finite* subset of all atoms—the remaining atoms do not change value). For such event models see [16, 17]—we follow notational conventions as in [17]. If $\mathsf{post}(\mathsf{s})(p) = \psi$, then we also write that $p := \psi$ (the valuation of atom p becomes that of formula ψ) in the event of s. Dynamic operators expressing event model execution (*semantics*) can be seen as part of the logical language (*syntax*), similar to how this is done for automata-PDL [18].

Forgetting $Fg(p)$ is the event model that expresses that the agent cannot distinguish between two assignments having taken place: p becomes true, or p becomes false. It consists of two events, that are both points (this expresses non-determinism). Both events are always executable: their precondition is $\top$.

Definition 2 (Forgetting). *$Fg(p)$ is the event model $((\mathsf{S}, \mathsf{R}, \mathsf{pre}, \mathsf{post}), \mathsf{S}')$ where $\mathsf{S} = \{0, 1\}$, $\mathsf{R} = \mathsf{S} \times \mathsf{S}$, $\mathsf{pre}(0) = \top$ and $\mathsf{pre}(1) = \top$, $\mathsf{post}(0)(p) = \bot$ and $\mathsf{post}(1)(p) = \top$ (and $\mathsf{post}(i)(q) = q$ for all $q \neq p$, $i = 0, 1$), and $\mathsf{S}' = \mathsf{S}$.*

Definition 3 (Semantics). *Assume an epistemic model $M = (S, R, V)$.*

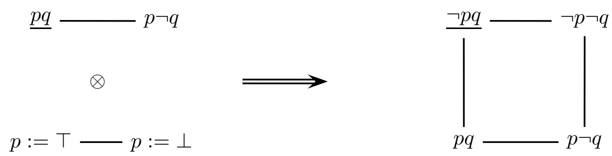
$M, s \models p$	*iff*	$s \in V(p)$
$M, s \models \neg\varphi$	*iff*	$M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	*iff*	$M, s \models \varphi$ *and* $M, s \models \psi$
$M, s \models K\varphi$	*iff*	*for all* $t \in S : (s, t) \in R$ *implies* $M, t \models \varphi$
$M, s \models [Fg(p)]\varphi$	*iff*	$M \otimes Fg(p), (s, 0) \models \varphi$ *and* $M \otimes Fg(p), (s, 1) \models \varphi$

where $M \otimes Fg(p) = (S', R', V')$ such that $S' = S \times \{0, 1\}$, $((s, i), (t, j)) \in R'$ iff $(s, t) \in R$ and $i, j \in \{0, 1\}$, $V'(p) = \{(s, 1) \mid s \in S\}$ and $V'(q) = V(q) \times S$ for $q \neq p$. The set of validities is called FG.

In fact, $M \otimes Fg(p)$ is the restricted modal product of M and event model $Fg(p)$ according to [19, 17], which in this case amounts to taking two copies of the model M, making p true everywhere in the first, making p false everywhere in the second, and making corresponding states indistinguishable for the agent.

Example We visualize $S5$ models by linking states that are indistinguishable for an agent. Reflexivity and transitivity are assumed. In these visualizations we abuse the language by writing valuations instead of states and postconditions instead of events, and we write $\top$ for an event with empty postcondition. The actual state is underlined.

Suppose the agent knows p but does not know q (and where in fact q is true), and where the agent forgets that p. The execution of event model $Fg(p)$ (and where in fact p becomes false) is pictured as follows. In the resulting Kripke model, the agent no longer knows p, and remains uncertain about q.



Deterministic forgetting The pointed (deterministic) versions of the forgetting event can be defined as notational abbreviations of the not-pointed primitives $[Fg(p), 1]\varphi \equiv_{\mathsf{def}} [Fg(p)](p \rightarrow \varphi)$ and $[Fg(p), 0]\varphi \equiv_{\mathsf{def}} [Fg(p)](\neg p \rightarrow \varphi)$. From this follow the validities $\langle Fg(p), 0\rangle\varphi \leftrightarrow \langle Fg(p)\rangle(\neg p \wedge \varphi)$ and $\langle Fg(p), 1\rangle\varphi \leftrightarrow \langle Fg(p)\rangle(p \wedge \varphi)$, and also the axiom for non-determinism

$$[Fg(p)]\varphi \leftrightarrow [Fg(p), 0]\varphi \wedge [Fg(p), 1]\varphi.$$

Axiomatization To obtain a complete axiomatization **FG** for the logic FG we apply the reduction axioms for event models, as specified in [19] and [17]. The case $[Fg(p)]p$ for the forgotten atom expresses that you cannot guarantee that p is true after forgetting it *by way of varying its value*; see Section 4 for a modelling where actual facts do not change value after forgetting. In the case for negation, note that $[Fg(p)]\neg\varphi$ is *not* equivalent to $\neg[Fg(p)]\varphi$, and note the correspondence with deterministic forgetting by abbreviation. The epistemic operator commutes with the forgetting operator. Thus the consequences of forgetting are known before it takes place ('no miracles'). We emphasize that the negated epistemic operator (for 'possible that') does *not* commute with forgetting ($[Fg(p)]\neg K\varphi$ is *not* equivalent to $\neg K[Fg(p)]\varphi$); therefore, K cannot be eliminated. Further details are omitted.[5] It follows that the axiomatization **FG** is sound and complete.

Definition 4 (Axiomatization FG). *Only axioms involving Fg are shown.*

$[Fg(p)]p$	$\leftrightarrow$	$\bot$	
$[Fg(p)]q$	$\leftrightarrow$	q	*for* $q \neq p$
$[Fg(p)]\neg\varphi$	$\leftrightarrow$	$\neg[Fg(p)](\neg p \rightarrow \varphi) \wedge \neg[Fg(p)](p \rightarrow \varphi)$	
$[Fg(p)](\varphi \wedge \psi)$	$\leftrightarrow$	$[Fg(p)]\varphi \wedge [Fg(p)]\psi$	
$[Fg(p)]K\varphi$	$\leftrightarrow$	$K[Fg(p)]\varphi$	

Theorem 1. *Axiomatization* **FG** *is sound and complete.*

Proof. The axiomatization resulted from applying the reduction axioms in [19, 17]. This kills two birds (soundness and completeness) in one throw. We show the basic case for the forgotten atom (the relevant axiom is $[\mathsf{M}, \mathsf{s}]p \leftrightarrow (\mathsf{pre}(\mathsf{s}) \rightarrow \mathsf{post}(\mathsf{s})(p))$ and non-determinism).

[5] The axiomatization **FG** can be made into a *reduction* system by having pointed event models as primitives instead of abbreviations. We then employ the reduction axiom for non-determinism (above) and $[Fg(p), 0]\neg\psi \leftrightarrow (\neg p \rightarrow [Fg(p), 0]\psi)$, etc.

$[Fg(p)]p$
iff
$[Fg(p), 0]p \wedge [Fg(p), 1]p$
iff
$(\mathsf{pre}(0) \rightarrow \mathsf{post}(0)(p)) \wedge (\mathsf{pre}(1) \rightarrow \mathsf{post}(1)(p))$
iff
$(\top \rightarrow \bot) \wedge (\top \rightarrow \top)$
iff
$\bot$

4 Results, and other forgetting operators

Using this simple logic we can now harvest an interesting number of theoretical results. A number of different perspectives on forgetting propositional variables (such as release, elimination, bisimulation quantification, symmetric contraction, and value swapping or switching) all amount to the same: although resulting in different structures, these cannot be distinguished from each other in the language, i.e., they all represent the same set of believed formulas. An important Theorem 2 states that becoming unaware (the original [1]-sense of forgetting as data abstraction) is the same as becoming ignorant. We also show that our results generalize to more agents or variables. (As long as R is serial, so that $K\bot \leftrightarrow \bot$ and $K\top \leftrightarrow \top$,) Ignorance is indeed obtained (and, trivially also awareness of it—$[Fg(p)]K(\neg Kp \wedge \neg K\neg p)$):

Proposition 1. *$[Fg(p)](\neg Kp \wedge \neg K\neg p)$ is valid.*

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 of forgetting, such as in the example on page 5. This is undesirable if we *only* want to model that the agents forget the value of p but that otherwise nothing changes: in particular, the actual value of p should not change. We can overcome that deficiency in the event model for *epistemic forgetting.*

Definition 5 (Epistemic forgetting). *Epistemic forgetting is the pointed event model $(\mathsf{Fg}(p), n)$ where $\mathsf{Fg}(p)$ is like $Fg(p)$ except that there is one more event n in the model, indistinguishable from the other two, with empty postcondition (and with precondition $\top$).*

The point n represents the event that 'nothing happens'. As it is the point of the event model, it ensures that the actual value of p does not change. We can visualize this event as

$p := \top$ ——— $\underline{\top}$ ——— $p := \bot$

Definition 6 (Axioms for epistemic forgetting). *The axioms for* $(\mathsf{Fg}(p), n)$ *are as for* $Fg(p)$ *except that*

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

Dynamic modal operators for pointed events $(\mathsf{Fg}(p), 0)$ and $(\mathsf{Fg}(p), 1)$ are again introduced in the language by abbreviation, now from $[\mathsf{Fg}(p), n]$, somewhat different from before. We have the results that (proof omitted)

Proposition 2 (Preservation of factual information).
Schema $\psi \rightarrow [\mathsf{Fg}(p), n]\psi$ *is valid for boolean* ψ*.*

Proposition 3 (Epistemic propositions are preserved).
Schema $[\mathsf{Fg}(p), n]K\psi \leftrightarrow [Fg(p)]K\psi$ *is valid (for all* ψ *in the language).*

In other words, from the perspective of the agent, the different modellings of forgetting are indistinguishable. The different occurrences of ψ in Proposition 3 are actually in different languages (both may contain forgetting operators!), a trivial translation can make this more precise.

That makes the simpler modelling $Fg(p)$ preferable over the slightly more complex $(\mathsf{Fg}(p), n)$. We are vague in Proposition 3 about 'the language', as the language with $[\mathsf{Fg}(p), n]\varphi$ as inductive construct is different from the language with $[Fg(p)]\varphi$ as inductive construct. More strictly the result is that $[\mathsf{Fg}(p), n]K\psi \leftrightarrow [Fg(p)]K\mathsf{trs}(\psi)$ is valid, subject to the translation with inductive clause $\mathsf{trs}([\mathsf{Fg}(p), n]\varphi) = [Fg(p)]\mathsf{trs}(\varphi)$.

Swapping values Yet another way to model forgetting is by making every state in the model indistinguishable from one wherein the value of p has been swapped / switched: if true, it became false, and if false it became true.

Definition 7 (Forgetting by swapping values). *Forgetting by swapping is the pointed event model that is like* $Fg(p)$ *except that in one event, the actual event, nothing happens, whereas in the other event the assignment* $p := \neg p$ *is executed.*

$p := \neg p$ ——— $\underline{\top}$

Again we can adjust the axiomatization, we obtain the results that actual facts do not change value, and that propositions under the scope of the epistemic operator are preserved.

Scrambling the valuation of the forgotten atom Instead of making p randomly (but indistinguishably!) true or false in every state of the Kripke model, the more proper way of 'releasing the value of p' in a modal logical context is to make p randomly true in a subset of the domain of the model. One can then make all those results indistinguishable from one another for the agent. Unlike the former, where two copies of the model M suffice, we now need $2^{|M|}$ copies.

Consider again the structure pq——$p\neg q$ encoding that the agent knows p but is ignorant about q. In proper Lin and Reiter [1] fashion, the models agreeing with pq——$p\neg q$ on anything except maybe p are the following four:

$$pq\text{——}p\neg q \quad pq\text{——}\neg p\neg q \quad \neg pq\text{——}p\neg q \quad \neg pq\text{——}\neg p\neg q$$

These four still have ignorance about q in common, but only two of them satisfy ignorance about p. We have encoded *unawareness* of p, but not *ignorance* about p. If we make corresponding points in all these models indistinguishable for the agent, it again follows that the agent is ignorant about p, and the result is bisimilar to that achieved by $Fg(p)$. (Bisimilarity is a notion of structural similarity that guarantees logical equivalence, i.e., of sets of beliefs, see [20].) Doing so, we can after all reclaim *unawareness* of p, using the more abstract perspective of *bisimulation quantification*: apart from the four above, pq——$p\neg q$ is also similar except for the value of p to other structures, e.g. to pq——$p\neg q$——$\neg pq$ (three indistinguishable states), which satisfies that $K(p \vee q)$ is true. Because if we abstract from the value of p, q——$\neg q$ is bisimilar to q——$\neg q$——q. This prepares the ground for the next paragraph.

Bisimulation quantification Becoming unaware of an atom p can be modelled as universal bisimulation quantification over p [21–23] namely as

$$[Fg^{\forall}(p)]\varphi \equiv_{\mathsf{def}} \forall p\varphi$$

where $M, s \models \forall p\varphi$ iff for all (M', s') s.t. $(M', s') \leftrightarrows_{P-p} (M, s)$: $(M', s') \models \varphi$. The notation $(M', s') \leftrightarrows_{P-p} (M, s)$ means that epistemic state (M', s') is bisimilar to epistemic state (M, s) with respect to the set of all atoms *except* p.[6] In other words the valuation of p may vary 'at random'. This includes the model constructed by $Fg(p)$ (and that by $(\mathsf{Fg}(p), n)$) from a given M so that

$$M \leftrightarrows_{P-p} M \otimes Fg(p)$$

which immediately delivers that:

Theorem 2. *If $\psi \in \mathcal{L}(P - p)$ then $\psi \to [Fg(p)]\psi$ is valid.*

This fixes *progression* in the AI sense (which formulas initially believed that do not involve p are still believed in the new state where p is forgotten), and therefore creates a strong link with [2].

Of course we do not have that $[Fg^{\forall}(p)](\neg Kp \wedge \neg K\neg p)$ is valid. To adjust this 'becoming unaware of p' towards 'becoming ignorant of p', we have to let all $P - p$-bisimilar states be indistinguishable by the agent.[7] For this alternative 'becoming ignorant by bisimulation quantification' operation $\mathsf{Fg}^{\forall}(p)$

[6] Applied to forgetting, this is the original proposal in [4]. Then, to achieve ignorance as in [3] they constrain this set of models to those satisfying $\neg Kp \wedge \neg K\neg p$. That is different from what we do.

[7] Given (M, s), let $\mathfrak{M} = \{(M', s') \mid (M, s) \leftrightarrows_{P-p} (M', s')\}$. Given $\mathfrak{R} : (M, s) \leftrightarrows_{P-p} (M', s')$ and $\mathfrak{R}' : (M, s) \leftrightarrows_{P-p} (M'', s'')$, add pairs (s', s'') to the relation R on $\mathfrak{M}$ whenever there is a $s \in S$ such that $(s, s') \in \mathfrak{R}$ and $(s, s'') \in \mathfrak{R}'$. Then $\mathfrak{M} \models \neg Kp \wedge \neg K\neg p$.

we again have the desired $[\mathsf{Fg}^{\forall}(p)](\neg Kp \wedge \neg K\neg p)$ and we then also have that $[\mathsf{Fg}^{\forall}(p)]K\varphi \leftrightarrow [Fg(p)]K\varphi$. The much simpler $Fg(p)$ is preferable for computational reasons.

Multiple variables The forgetting of multiple variables can be modelled by a simple adjustment. For n propositional variables, we get an event model $Fg(p_1, \dots, p_n)$ with a domain consisting of 2^n events, one for each combination of assignments of different variables to true and false. All prior results still follow (including bisimulation quantification for n variables).

Combining learning and forgetting One might wish to combine forgetting with other dynamic operations such as learning (by public announcements). We simply add an inductive construct $[\varphi]\psi$ to the language, which stands for 'after announcement of φ, ψ holds' (see [24]). The resulting logic is again equally expressive as epistemic logic: just add the rewrite rules involving announcements.

Multiple agents Given a parameter set of agents A, we adjust the language to $\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid K_a\varphi \mid [Fg_B(p)]\varphi$, where $a \in A$ and $B \subseteq A$, and we adjust the accessibility relation $R \subseteq (S \times S)$ to an accessibility function $R : A \to \mathcal{P}(S \times S)$—accessibility relation R_a (for $R(a)$) interprets operator K_a (knowledge of agent a). (Further details omitted.) The case $Fg_A(p)$ where $B = A$ models forgetting as group ignorance, and the case $Fg_a(p)$ where $B = \{a\}$ models the forgetting of an individual a in the presence of A (see page 2). Both are most succinctly modelled by a version of 'swapping values' forgetting (Definition 7) namely as event models visualized as

$$\underline{p := \neg p} \overset{A}{\text{———}} \underline{\top} \qquad\qquad \underline{p := \neg p} \overset{a}{\text{———}} \underline{\top}$$

The visualization on the right means that all agents *except* a can distinguish between the two alternatives: access for a is the universal relation on the domain, and access for all agents in $A - a$ is the identity. Again, all former results generalize, both versions are axiomatizable very similarly to the previous, etc. The more obvious multi-agent version of $Fg(p)$ (with assignments to true and to false only) does *not* model individual forgetting in a group: this would express that the other agents learn that p is true or learn that p is false, clearly undesirable.

5 Further research

Remembering prior knowledge For the agent to recall prior knowledge we have to be able to refer to past events. Let $Fg(p)^-$ be the converse of $Fg(p)$ (e.g. in the sense of [13–15]). Awareness of present ignorance and prior knowledge about p can now be formalized as

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

We now need a structure allowing us to interpret such converse events. This is not possible in pointed Kripke models, but it can be elegantly done employing what is known as the 'forest' produced by the initial Kripke model and all possible sequences of all $Fg(p)$ events (for all atoms), see [25–28, 14, 15]. We now add assignments to the language, as in the underlying proposal, and additionally add theories for event models using converse actions [13, 26]. Thus we get a complete axiomatization, though not a reduction result to epistemic formulas (converse events cannot be eliminated from the logical language by equivalences).

Regression and progression By applying equivalences we can reduce a formula of the form $[Fg(p)]\psi$ to an equivalent expression χ without dynamic operators—ψ is a final condition and χ is an initial condition that is derived from it by way of these equivalences. This process is known as *regression*. Dynamic epistemic logic is very suitable for this kind of regression, and there are efficient model checkers for epistemic formulas. Progression is harder in this setting. (See page 2.)

Forgetting modal formulas How to model the forgetting modal formulas is a different piece of cake altogether; in this case we have made no progress yet.

Forgetting of events This amounts to introducing *temporal uncertainty* in the model, apart from epistemic uncertainty. This can be done by introducing histories of events to structures, or moving to a temporal epistemic perspective using 'forests', as above, see [26]. It is clear how this has to be done, and the results should prove interesting.

Acknowledgements The authors would like to thank the four anonymous AI08 referees, and also the participants of the ESSLLI 2008 Hamburg workshop Logic and Intelligent Interaction. Hans van Ditmarsch acknowledges support of the Netherlands Institute of Advanced Study where he was Lorentz Fellow in 2008.

References

1. Lin, F., Reiter, R.: Forget it! Presented at the AAAI Fall Symposium on Relevance, New Orleans (1994)
2. Lang, J., Liberatore, P., Marquis, P.: Propositional independence: Formula-variable independence and forgetting. J. Artif. Intell. Res. (JAIR) **18** (2003) 391–443
3. Baral, C., Zhang, Y.: Knowledge updates: semantics and complexity issues. Artificial Intelligence **164**(1-2) (2005) 209–243
4. Zhang, Y., Zhou, Y.: Knowledge forgetting: Properties and applications. Work in progress under submission (2008)
5. Wang, K., Sattar, A., Su, K.: A theory of forgetting in logic programming. In: AAAI. (2005) 682–688
6. Zhang, Y., Foo, N., Wang, K.: Solving logic program conflict through strong and weak forgettings. In: Proceedings of IJCAI. (2005) 627–634
7. Eiter, T., Wang, K.: Forgetting and conflict resolving in disjunctive logic programming. In: Proceedings of AAAI. (2006)
8. Zhao, Y., Wang, K., Topor, R., Pan, J., Giunchiglia, F.: Semantic cooperation and knowledge reuse by using autonomous ontologies. In: ISWC/ASWC. (2007) 666–679

9. Erdem, E., Ferraris, P.: Forgetting actions in domain descriptions. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, AAAI Press (2007) 409–414
10. Herzig, A., Lang, J., Marquis, P.: Action representation and partially observable planning using epistemic logic. In: Proceedings of IJCAI. (2003)
11. van Ditmarsch, H.: Prolegomena to dynamic logic for belief revision. Synthese (Knowledge, Rationality & Action) **147** (2005) 229–275
12. Baltag, A., Smets, S.: 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)
13. Aucher, G.: Perspectives on belief and change. PhD thesis, University of Otago & Institut de Recherche en Informatique de Toulouse, New Zealand & France (2008)
14. Yap, A.: Product update and looking backward. Technical report, University of Amsterdam (2006) ILLC Research Report PP-2006-39.
15. Sack, Y.: Adding Temporal Logic to Dynamic Epistemic Logic. PhD thesis, Indiana University, Bloomington, USA (2007)
16. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Information and Computation **204(11)** (2006) 1620–1662
17. van Ditmarsch, H., Kooi, B.: Semantic results for ontic and epistemic change. In Bonanno, G., van der Hoek, W., Wooldridge, M., eds.: Post-proceedings of LOFT 2006. Amsterdam University Press (2008) To appear in the series Texts in Logic and Games.
18. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge MA (2000) Foundations of Computing Series.
19. Baltag, A., Moss, L., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In Gilboa, I., ed.: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98). (1998) 43–56
20. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001) Cambridge Tracts in Theoretical Computer Science 53.
21. Visser, A.: Bisimulations, model descriptions and propositional quantifiers (1996) Logic Group Preprint Series 161, Department of Philosophy, Utrecht University.
22. French, T.: Bisimulation quantifiers for modal logic. PhD thesis, University of Western Australia (2006)
23. van Ditmarsch, H., French, T.: Simulation and information. (Electronic) Proceedings of LOFT 2008, Amsterdam (2008)
24. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Volume 337 of Synthese Library. Springer (2007)
25. Parikh, R., Ramanujam, R.: Distributed processing and the logic of knowledge. In: Logic of Programs. Volume 193 of Lecture Notes in Computer Science., Springer (1985) 256–268 A newer version appeared in *Journal of Logic, Language and Information*, vol. 12, 2003, pp. 453–467.
26. van Benthem, J., Gerbrandy, J., Pacuit, E.: Merging frameworks for interaction: DEL and ETL. In Samet, D., ed.: Proceedings of TARK 2007. (2007) 72–81
27. Pacuit, E.: Some comments on history-based structures. To appear in Journal of Applied Logic (2007)
28. van Ditmarsch, H., Ruan, J., van der Hoek, W.: Model checking dynamic epistemics in branching time. In: (Informal) Proceedings of FAMAS 2007, Durham UK. (2007)