Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Action Models with Postconditions

Mario Benevides, Isaque Maçalam Saab Lima

Federal University of Rio de Janeiro,
Programa de Engenharia de Sistemas e Computação e Instituto de Matemática,
Brazil

mario@cos.ufrj.br, isaque@ufrj.br

Abstract. This work proposes a extension of dynamic epistemic logic to work with assignment. The difference between this work and others works, such as [7], is the use of actions models, from DEL, to make Boolean assignments to the propositions, instead of creating new mechanisms to make the assignments. We extend the definition of action model by creating the postcondition property of each state of the model, making it possible to assign Boolean values to the propositions.

Keywords. Epistemic logic, action models, postconditions.

1 Introduction

Although the formalization of epistemic logic dates from 1962 ([3]) and the concept of attribution is a primitive operation of programming languages, the idea of joining these two concepts, to change values of propositions in the epistemic logic, is more recent, being found in [7], [4], [1] and [6].

In this work, we aim to reinforce the connection between these concepts, presenting a new approach to perform assignment operations in propositions of Dynamic Epistemic Logic. This approach differs from those found in [7] and [4], since it does not change the dynamics of the updates to perform the assignments, because we use the action model from Dynamic Epistemic Logic to deal with assignment, what does not occur in the works mentioned above. Our work can be seen as a simplification of [6], where assignments are treated as substitutions. In our proposal, the assignments are part of the action model and are performed as postconditions being much more intuitive and easy to define.

We extend the concept of an action model to perform assignment operations by adding a postcondition property in the structure of the action model. Thus, we can use this new action model in Epistemic Logic with Assignment. We will restrict ourselves to operations of Boolean assignments, that is, a proposition can only receive true or false values.

2 Action Model with Assignment

Action model has a structure that resembles the Kripke model [5], where each action has a precondition that must be satisfied for the action to be performed. Our proposal is that in addition to a precondition, each action must also have a postcondition, which would be a list of Boolean assignments.

2.1 Example: Action Model with Assignment

Suppose a card game with three cards **x**, **y** and **w**, and three players **a**, **b** and **c**. Initially, each player receives a card and does not know the card of the other players. We use a positional notation where x, y, w means that player a has the card x, b has the card y and c has the card w. Edges labeled with the agent name means that the agent does not distinguish the bound states from the edge. Initially, we have this epistemic model:

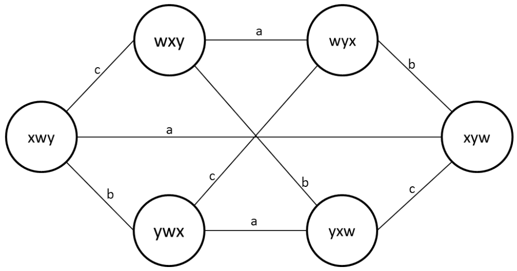

Fig. 1. Card game with three cards and three players

Suppose that players a and b show their cards to each other. Since there was no card exchange, we can model this action model as an action model without assignment.

In this case we have the action model of the Fig.2. This model has six possible actions ($s_1, ..., s_6$), one for each combination of three cards arranged two-by-two .

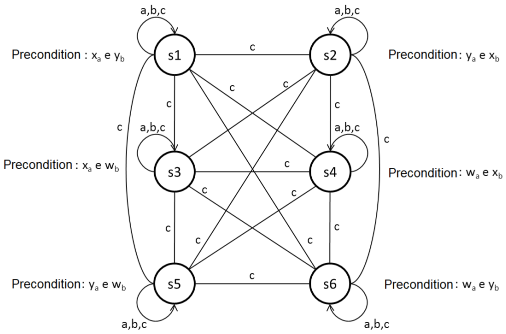
Precondition : x_a e y_b
Precondition : y_a e x_b
Precondition : x_a e w_b
Precondition : w_a e x_b
Precondition : y_a e w_b
Precondition : w_a e y_b
s1
s2
s3
s4
s5
s6
a,b,c
c

Fig. 2. Action Model without Assignment

From the action model above we can notice that players a and b have no doubt among actions, but player c does not know which action it was performed.

Now suppose that initially player a has the card x, player b has the card y and that players a and b has exchanged their cards. The precondition for the player a to exchange the card x for the card y with the player b is that the player a must have the card x and the player b must have the card y.

The postcondition of action of player a swapping cards with player b is $x_a = false$, $x_b = true$, $y_a = true$ and $y_b = false$, where x_a , x_b, y_a y_b means that a certain card (x—y) is in the possession of a certain agent(a—b). Extending this concept for all possible states we have the following action model with assignment:

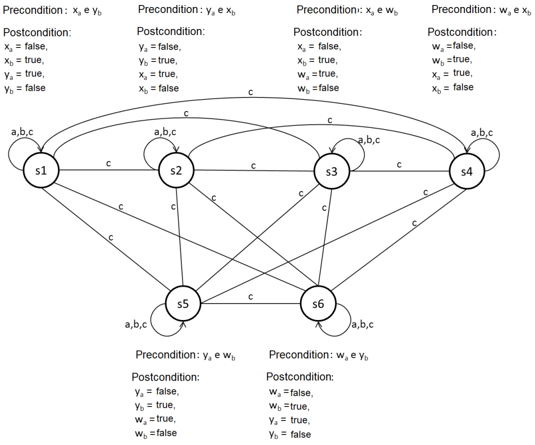

Fig. 3. Action Model with Assignment

This model is similar to the model present before, the difference is that in this model we add the postconditions.

We will make the Cartesian product of the initial epistemic model with the action model with assignment. For a better visualization, we will omit the nodes generated by the Cartesian product that do not satisfy the preconditions, since they will be eliminated in any way.

We will show this product step by step, first before applying the postcondition (Fig.4) and after applying it (Fig.5).

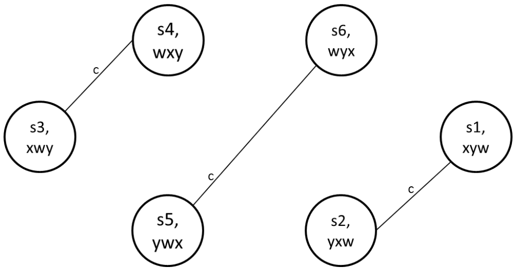

Fig. 4. Epistemic Model before postconditions

Notice that the edges ’a’ and ’b’ were removed by the action model. This is because a and b have no further questions about what the actual state of the system is. The player c knows that players a and b know the real state of the system, but he does not know what that state is.

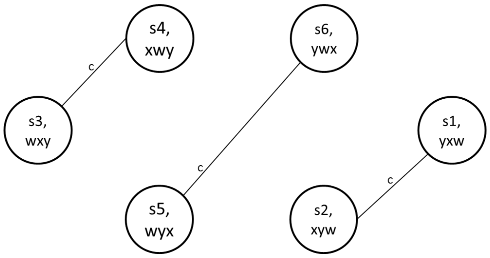

Fig. 5. Epistemic Model after postconditions

At that point, players a and b exchanged the cards and player c knows that an exchange took place but does not know which one. We can see that the whole process occurs normally, as if it were a simple action model, only at the end the assignments are made.

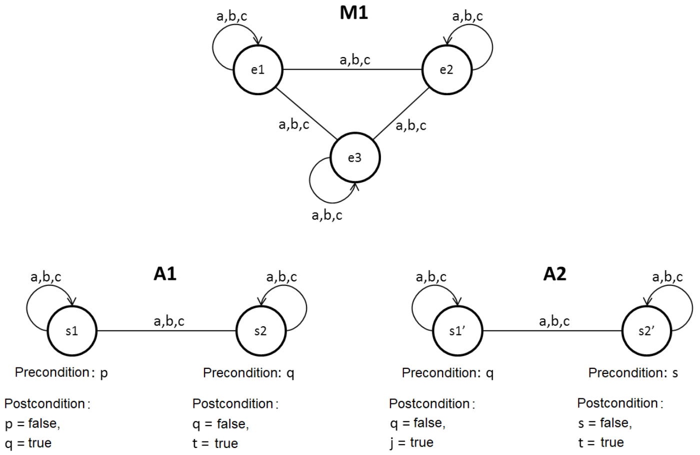

Fig. 6. Composition of action models

2.2 Syntax and Semantics

Definition 1 *The language of the action model with assignments consists of a finite set Φ $(p_1, p_2, ..., p_n)$ of propositional symbols, a finite set $\mathcal{A}$ of agents, the boolean connectives $\neg$ and $\wedge$, the operator K_a for each agent a, and the operator $[\mathsf{M}, \mathsf{j}]$. The formulas are defined as follows:*

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_a\varphi \mid [\alpha]\varphi$$

$$\alpha ::= (\mathsf{M}, \mathsf{j}) \mid (\alpha \cup \alpha) \mid (\alpha; \alpha).$$

Definition 2 *A action model with assignments* M *consist of a structure $\langle \mathsf{S}, \sim, \mathsf{pre}, \mathsf{pos}\rangle$, where:*

- *S is a finite domain of action points or events,*
- *$\sim_a$ is the equivalence relation in S for the agent a,*
- *$\mathsf{pre} : \mathsf{S} \mapsto \mathcal{L}$ is the precondition function that assigns a precondition for each $\mathsf{j} \in \mathsf{S}$,*
- *$\mathsf{pos}(\mathsf{j}) = \{(p, x) | \forall p \in \Phi \; and \; x = V \; or \; F\}$.*

The language of the action model with assignments is the same language of the action model without assignment. The premise that if an agent can differentiate the two actions consequently he can differentiate the states resulting from these actions, still hold.

Definition 3 *Given an epistemic state $(\mathcal{M}, s)$ with $\mathcal{M} = \langle S, \sim_a, V\rangle$ and a rooted action model (M, j) with $\mathsf{M} = \langle \mathsf{S}, \sim, \mathsf{pre}, \mathsf{pos}\rangle$, the result of a execution of (M, j) in $(\mathcal{M}, s)$ is $(\mathcal{M} \otimes \mathsf{M}, (s, \mathsf{j}))$ where $\mathcal{M} \otimes \mathsf{M} = \langle S', \sim', V'\rangle$ such that:*

1. *$S' = \{(s, \mathsf{j}) | s \in S, \mathsf{j} \in \mathsf{S},$ and $\mathcal{M}, s \models \mathsf{pre}(\mathsf{j})\}$,*
2. *$(s, \mathsf{j}) \sim'_a (t, \mathsf{k})$ iff $(s \sim_a t$ and $\mathsf{j} \sim_a \mathsf{k})$,*
3. *$V'(p) = \{(s, \mathsf{j}) \mid (p, V) \in \mathsf{pos}(\mathsf{j})\}$.*

Definition 4 *Given a rooted action model (M, j) with $\mathsf{M} = \langle \mathsf{S}, \sim, \mathsf{pre}, \mathsf{pos}\rangle$, the definition of fpos(j) is:*

1. *$L(\mathsf{j}) = \{p \mid (p, V) \in \mathsf{pos}(\mathsf{j})\}$, set of true propositions in* j.
2. *$p_1, ..., p_h \in L(\mathsf{j})$.*
3. *$q_1, ..., q_m \notin L(\mathsf{j})$.*
4. *fpos$(\mathsf{j}) = p_1 \wedge \cdots \wedge p_h \wedge \neg q_1 \wedge \cdots \wedge \neg q_m$.*

Definition 5 *Given an epistemic state $(\mathcal{M}, s)$ with $\mathcal{M} = \langle S, \sim_a, V\rangle$ and a rooted action model (M, j) with $\mathsf{M} = \langle \mathsf{S}, \sim, \mathsf{pre}, \mathsf{pos}\rangle$, the notion of satisfaction $\mathcal{M}, s \models \varphi$ is defined below:*

$$\begin{array}{lll}
\mathcal{M}, s \models p & iff & s \in V(p) \\
\mathcal{M}, s \models \neg\phi & iff & \mathcal{M}, s \not\models \phi \\
\mathcal{M}, s \models \phi \wedge \psi & iff & \mathcal{M}, s \models \phi \text{ and } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models K_a\phi & iff & \text{for all } s' \in S : s \sim_a s' \text{ implies} \\
 & & \mathcal{M}, s' \models \phi \\
\mathcal{M}, s \models [\mathsf{M}, \mathsf{j}]\phi & iff & \mathcal{M}, s \models \mathsf{pre}(\mathsf{j}) \text{ implies} \\
 & & \mathcal{M}, s \otimes \mathsf{M}, (s, \mathsf{j}) \models (\mathit{fpos}(\mathsf{j}) \rightarrow \phi).
\end{array}$$

2.2.1 Composition of action models with assignments

In the action models without assignments it is possible to make the composition of models. For this to work in the action models with assignments, we have to make some adjustments. Initially, we will think informally about this problem, then formalize the definition of composition of action models with assignments.

To make it easy to understand, we will use a simple epistemic model (M1), with only 3 states, and 2 action models (A1 and A2) with only 2 actions, as shown below:

In action models without assignments the order in which the composition is made doesn't matter, since the end result will be the same. In action models with assignment the values of propositions are changed and therefore not necessarily this rule remains true.

Suppose that we apply the action A1 in the model M1, generating the model M2, and then we apply the action A2 in the model M2, generating the model M3. This sequence can be seen in the figure below:

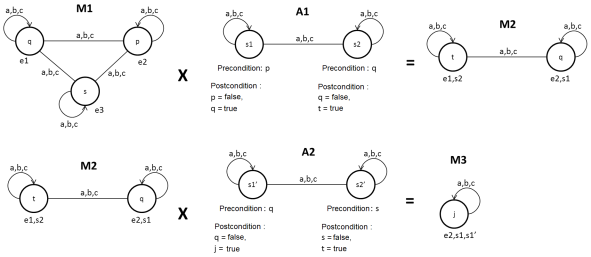

Fig. 7. Update of M1 with A1 and A2

However, if we apply the action A2 in the model M1, generating the model M2' and then apply the action A1 in this model, we have the following result:

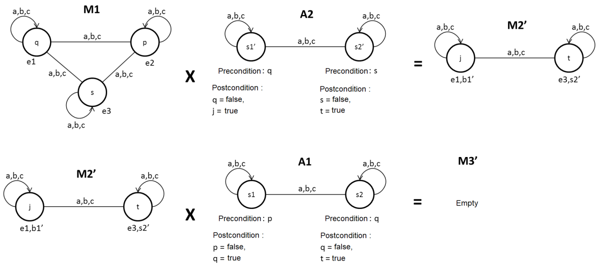

Fig. 8. Update of M1 with A2 and A1

We can conclude that, in the case of the composition of action models with assignments, the order matters because an action can alter a proposition and thus render the other action, which was incompatible in the initial model, compatible or otherwise.

Initially, we will take a naive approach to compose two action models (A1 and A2, in that order), simply by making the Cartesian product of them and ignoring the pre and postconditions. With this, we have the action model of the Fig.9.

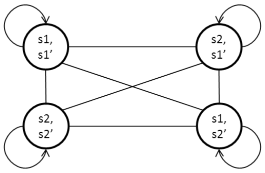

Fig. 9. Composition of action models A1 and A2 before pre and postconditions

Now, in each resulting state (action), we verify whether the postconditions of the action of model A1 are compatible with the preconditions of the action of model A2. If they are not compatible, we will delete the state. With this, the state (s2, s1') has been eliminated, resulting in the model of the Fig.10.

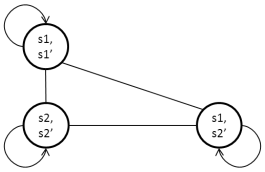

Fig. 10. Composition of action models A1 and A2 after pre and postconditions

The precondition of the action model resulting from the composition of the two action models has to contemplate the needs of the two action models, so a naive idea would be to merge the preconditions.As the postcondition changes the values of the propositions, we have to take them in consideration. For example, if q = true is postcondition of s1 and q is precondition of s1', q does not have to be in the preconditions of state (s1, s1'), because it will always be true. Thus, the precondition of the new state is formed by the merge of the precondition of the model A1 with the precondition of the model A2, excluding the true propositions in the postcondition A1.

In the case of postconditions it is simpler, as we always have assignments for all propositions of the model in postconditions, the postcondition of the resulting state (action) will be the postcondition of the second action.

Definition 6 *Given the rooted action models* (M,j) *with* $\mathsf{M} = \langle \mathsf{S}, \sim, \mathsf{pre}, \mathsf{pos}\rangle$ *and* $(\mathsf{M}',\mathsf{j}')$ *with* $\mathsf{M}' = \langle \mathsf{S}', \sim', \mathsf{pre}', \mathsf{pos}'\rangle$*, their composition is the model of action* $(\mathsf{M};\mathsf{M}',(\mathsf{j},\mathsf{j}'))$ *with* $\mathsf{M};\mathsf{M}' = \langle \mathsf{S}'', \sim'', \mathsf{pre}'', \mathsf{pos}''\rangle$*:*

- $\mathsf{S}'' = \{(\mathsf{j},\mathsf{j}') \mid \mathsf{j} \in \mathsf{S}, \mathsf{j}' \in \mathsf{S}'\}$,
- $(\mathsf{j},\mathsf{j}') \sim''_a (\mathsf{k},\mathsf{k}')$ *iff* $(\mathsf{j} \sim_a \mathsf{k}$ *and* $\mathsf{j}' \sim_a \mathsf{k}')$,
- $\mathsf{pre}''(\mathsf{j},\mathsf{j}') = \langle(\mathsf{M},\mathsf{j})\rangle \mathsf{pre}'(\mathsf{j}')$,
- $\mathsf{pos}''(\mathsf{j},\mathsf{j}') = \mathsf{pos}'(\mathsf{j}')$.

Incompatible states are eliminated by the precondition, that is, states where the precondition is $\perp$

Using this definition, the composition of the action models A1 and A2 (in that order), generates the following action model of the Fig.11.

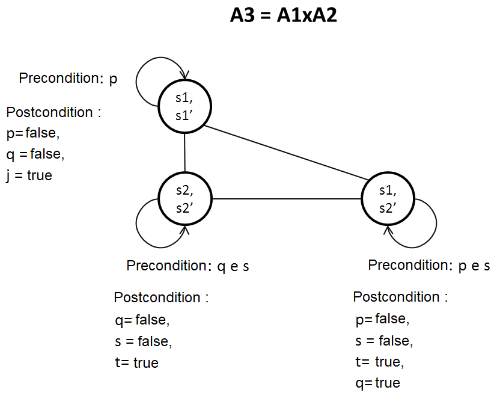

Fig. 11. Composition of A1 and A2

Applying A3 in M1, we have the epistemic model from the Fig.12.

We can note that the result is the same as when we applied the actions separately.

2.2.2 Axiomatic Systems

The following is an axiomatization for the Dynamic Epistemic Logic with Attributions **DELWA**:

fpos(j) = $p_1 \wedge ... \wedge p_n \wedge \neg p_{n+1} \wedge ... \wedge \neg p_m$.

Axioms

Axioms of Epistemic Logic [2]:

1. $[\mathsf{M},\mathsf{j}]p \leftrightarrow (\mathsf{pre}(\mathsf{j}) \to (\mathit{fpos}(\mathsf{j}) \to p))$,
2. $[\mathsf{M},\mathsf{j}]\neg\phi \leftrightarrow (\mathsf{pre}(\mathsf{j}) \to \neg[\mathsf{M},\mathsf{j}]\phi)$,
3. $[\mathsf{M},\mathsf{j}](\phi \wedge \psi) \leftrightarrow ([\mathsf{M},\mathsf{j}]\phi \wedge [\mathsf{M},\mathsf{j}]\psi)$,
4. $[\mathsf{M},\mathsf{j}]K_a\phi \leftrightarrow (\mathsf{pre}(\mathsf{j}) \to \bigwedge_{\mathsf{j}\sim_a\mathsf{k}} K_a[\mathsf{M},\mathsf{k}]\phi)$,
5. $[[\mathsf{M},\mathsf{j}] \cup [\mathsf{M}',\mathsf{j}']]\phi \leftrightarrow [\mathsf{M},\mathsf{j}]\phi \wedge [\mathsf{M}',\mathsf{j}']\phi$,

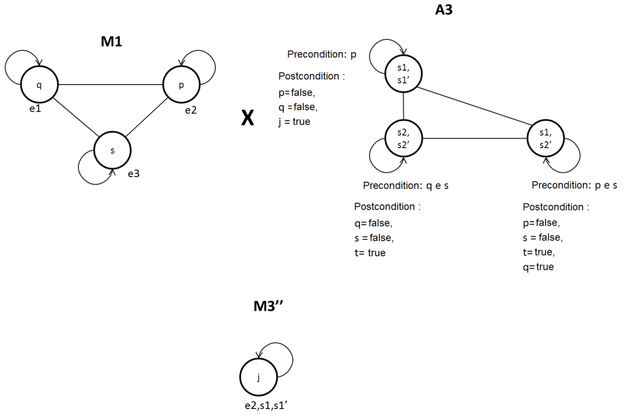

Fig. 12. M1 with A3

6. $[\mathsf{M},\mathsf{j}][\mathsf{M}',\mathsf{j}']\phi \leftrightarrow [(\mathsf{M},\mathsf{j});(\mathsf{M}',\mathsf{j}')]\phi$ (Composition of action models).

Inference Rules

M.P. $\varphi, \varphi \to \psi/\psi$ U.G. $\varphi/K_a\varphi$ $\varphi/[\alpha]\varphi$ UB. $\varphi/\sigma\varphi$,

where σ is a uniform substitution of formulas by propositional variables.

2.2.3 Soundness

To prove the our logic is correct, we need to show that our axioms are valid. The addition of the postcondition condition changes only axioms 1 and 5. We need to prove that they continue to be valid with the postcondition.

Lemma 1 $[\mathsf{M},\mathsf{j}]p \leftrightarrow (\mathsf{pre}(\mathsf{j}) \to (\mathit{fpos}(\mathsf{j}) \to p))$ *is valid.*

Lemma 2 $[\mathsf{M},\mathsf{j}][\mathsf{M}',\mathsf{j}']\phi \leftrightarrow [(\mathsf{M},\mathsf{j});(\mathsf{M}',\mathsf{j}')]\phi$ *is valid.*

2.2.4 Completeness

Completeness follows directly from the axioms, defining a translation/reduction of the action model with assignments for S5. This proof was based on the proof of section 7.6 of [8].

We define a translation of formulas that contain modalities of action for formulas that do not contain these modalities. This translation follows the axioms of the proof system.

Definition 7 *The translation t: $\mathcal{L}_{K\otimes} \rightarrow \mathcal{L}_K$ is defined as follows:*

$$\begin{array}{lcl}
t(p) & = & p, \\
t(\neg\varphi) & = & \neg t(\varphi), \\
t(\varphi \wedge \psi) & = & t(\varphi) \wedge t(\psi), \\
t(K_a\varphi) & = & K_a t(\varphi), \\
t([\mathsf{M},\mathsf{j}]p) & = & t(\mathsf{pre}(\mathsf{j}) \rightarrow (\mathit{pos}(\mathsf{j}) \rightarrow p)), \\
t([\mathsf{M},\mathsf{j}]\neg\varphi) & = & t(\mathsf{pre}(\mathsf{j}) \rightarrow \neg[\mathsf{M},\mathsf{j}]\varphi), \\
t([\mathsf{M},\mathsf{j}](\varphi \wedge \psi)) & = & t([\mathsf{M},\mathsf{j}]\varphi \wedge [\mathsf{M},\mathsf{j}]\psi), \\
t([\mathsf{M},\mathsf{j}]K_a\varphi) & = & t(\mathsf{pre}(\mathsf{j}) \rightarrow K_a[\mathsf{M},\mathsf{j}]\varphi), \\
t([\mathsf{M},\mathsf{j}][\mathsf{M}',\mathsf{j}']\varphi) & = & t([\mathsf{M},\mathsf{j};\mathsf{M}',\mathsf{j}']\mathcal{X}), \\
t([\alpha \cup \alpha']\varphi) & = & t([\alpha]\varphi) \wedge t([\alpha']\varphi).
\end{array}$$

One can prove that every formula is equivalent to its translation.

Lemma 3 *For all formulas $\varphi \in \mathcal{L}_{K\otimes}$, this holds:*

$$\vdash \varphi \leftrightarrow t(\varphi).$$

Completeness follows automatically.

Theorem 1 *Completeness For all $\varphi \in \mathcal{L}_{K\otimes}$*

$$\models \varphi \text{ implies } \vdash \varphi.$$

Proof 1 *Suppose $\models \varphi$. By Lemma 3 we have $\vdash \varphi \leftrightarrow t(\varphi)$, by soundness we can conclude that $\models t(\varphi)$. Since $t(\varphi)$ does not have any action it is a* **S5** *formula and since* **S5** *is complete we have $\vdash_{S5} t(\varphi)$,and since* **S5** *is contained in* **DELWA***,we have $\vdash_{DELWA} t(\varphi)$.*

3 Conclusion

In this paper we propose an extension of the Dynamic Epistemic Logic with postconditions. The main motivation is to represent situations where the actions do not only change the epistemic state of the agents, but also change the Boolean value of propositional state variables. In Dynamic Logics the actions remove doubts but every proposition that was true/false before execution remains true/false after the execution. In order to contemplate these changes we define a new Dynamic Logic **DELWA** that is able to represent Boolean assignments.

Our approach differs from others found in the literature because it uses the framework of the action model (which is widely used in dynamic epistemic logic) to accomplish the assignments. This framework is used by DEMO [9] (epistemic model checker).

We show that all definitions, with their respective adaptations, of the action model of dynamic epistemic logic also apply to the new action model with assignments, however, some of the definitions contains constraints. For example, in the composition of action models with assignments to the order in which the composition is made matters, since each action can modifies the values of the propositions, which did not happen in the action models without assignments, because the values never change.

A constraint imposed in this work, for the action model with assignments, is that the assignments would be only Boolean. This constraint was placed to make easier the formalization of the composition of action models. We can indicate as direction of the future work the removal of this constraint so that any type of attribution can be made.

References

1. **Balbiani, P., Herzig, A., & Troquard, N. (2013).** Dynamic logic of propositional assignments: A well-behaved variant of PDL. *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pp. 143–152.
2. **Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. Y. (1995).** *Reasoning About Knowledge*. The MIT Press.
3. **Hintikka, J. (1962).** *Knowledge and Belief*. Cornell University Press, Ithaca, N.Y.
4. **Kooi, B. (2007).** Expressivity and completeness for public update logic via reduction axioms. *Journal of Applied Non-Classical Logics*, Vol. 17, No. 2, pp. 231–253.
5. **Kripke, S. A. (1959).** A completeness theorem in modal logic. *Journal of Symbolic Logic*, Vol. 24, No. 1, pp. 1–14.
6. **van Benthen, J., van Eijck, J., & Kooi, B. (2006).** Logics of communication and change. *Information and Computation*, Vol. 204, No. 11, pp. 1620–1662.
7. **van Ditmarsch, H., van der Hoek, W., & Kooi, B. (2005).** Dynamic epistemic logic with assignment. *AAMAS*, Vol. 4, No. 1, pp. 141–148.
8. **van Ditmarsch, H., van der Hoek, W., & Kooi, B. (2008).** *Dynamic Epistemic Logic*. Springer.
9. **van Eijck, J. (2006).** *DEMO - A Demo of Epistemic Modelling*.

Article received on 02/09/2017; accepted on 19/09/2017.
Corresponding author is Mario Benevides.