A New Semantics for Action Language m$\mathcal{A}^*$

Loc Pham(✉), Yusuf Izmirlioglu(✉), Tran Cao Son, and Enrico Pontelli

New Mexico State University, Las Cruces, NM, USA
{locpham,yizmir,epontell}@nmsu.edu, tson@cs.nmsu.edu

Abstract. The action language m$\mathcal{A}^*$ employs the notion of update models in defining transitions between states. Given an action occurrence and a state, the update model of the action occurrence is automatically constructed from the given state and the observability of agents. A main criticism of this approach is that it cannot deal with situations when agents' have incorrect beliefs about the observability of other agents. The present paper addresses this shortcoming by defining a new semantics for m$\mathcal{A}^*$. The new semantics addresses the aforementioned problem of m$\mathcal{A}^*$ while maintaining the simplicity of its semantics; the new definitions continue to employ simple update models, with at most three events for all types of actions, which can be constructed given the action specification and independently from the state in which the action occurs.

Keywords: Epistemic reasoning · Update models · Action language

1 Introduction

In multi-agent environments, agents not only need to reason about properties of the world, but also about agents' knowledge and beliefs. Among the various formalisms for reasoning about actions in Multi-Agent Systems (MAS), a commonly used one is the *action model*, introduced in [1,2] and later extended to the *update model* [5,9]. Update models have been employed in the study of epistemic planning problems in MAS [3,6,11]. The action language m$\mathcal{A}^*$, proposed in [4], and its earlier versions are among the first action languages that utilize update models in defining a transition function based semantics for multi-agent domains. Update models have also been adopted in [13]. Given an action occurrence, a corresponding update model is automatically derived from the action description and the pointed Kripke model encoding the current state of the world and the state of beliefs/knowledge of agents; such update model is used to compute the resulting state from the action occurrence. This simple construction only uses update models with at most three events. However, as discussed in [4,8], the simplicity of m$\mathcal{A}^*$ presents some challenges for its application.

Example 1. Three agents, A, B and C, are in a room with a box containing a coin. It is common knowledge that: (1) no agent knows whether the coin lies

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
R. Aydoğan et al. (Eds.): PRIMA 2022, LNAI 13753, pp. 553–562, 2023.
https://doi.org/10.1007/978-3-031-21203-1_33

heads or tails up; (2) the box is locked and only A can open it; (3) an agent needs to peek into the open box to learn the position of the coin; and (4) if one agent is looking at the box and a second agent peeks into the box, then the first agent will learn that the second agent knows the status of the coin; nevertheless, the first agent's knowledge about which face of the coin is up will not change.

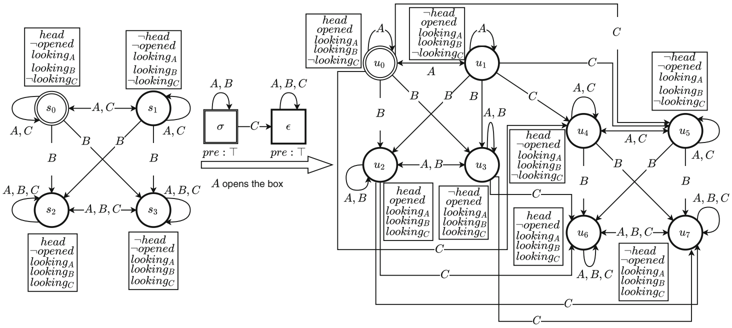

Fig. 1. A opens the box

Suppose that only A and B are looking at the box. However, B believes that all three agents have their eyes on the box. The situation is illustrated by the pointed Kripke structure on left of Fig. 1. Assume that A opens the box and anyone who is looking at the box will observe this action. Intuitively, after A opens the box, B should believe that C is also aware of the box being open. The design of m$\mathcal{A}^*$ produces the pointed Kripke structure on the right of Fig. 1, leading to the conclusion that B thinks that C considers the box still closed. This is because the current update models in m$\mathcal{A}^*$ assume that all full observers know about the observability of all other agents. While this assumption is reasonable in many situations, it implies that the use of m$\mathcal{A}^*$ requires very careful considerations by the domain designers, as pointed out in [4].

In this paper, we propose an extension of the language m$\mathcal{A}^*$ that can handle situations where agents' have incorrect beliefs about the observability of other agents, using *edge-conditioned event update models*, introduced by [7]. We begin with a short review of m$\mathcal{A}^*$, then show how to apply the edge-conditioned event update models to help m$\mathcal{A}^*$ solve the problems mentioned in the previous example. We prove relevant properties and provide final considerations.

2 Background

Belief Formulae. A *multi-agent* domain $\langle \mathcal{AG}, \mathcal{F} \rangle$ includes a finite and non-empty set of agents $\mathcal{AG}$ and a set of fluents (atomic propositions) $\mathcal{F}$ encoding properties of the world. *Belief formulae* over $\langle \mathcal{AG}, \mathcal{F} \rangle$ are defined by the BNF: "$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \mathbf{B}_i \varphi$" where $p \in \mathcal{F}$ is a fluent and $i \in \mathcal{AG}$. We refer to a belief formula which does not contain any occurrence of $\mathbf{B}_i$ as *a fluent formula*. In addition, for a formula φ and a non-empty set $\alpha \subseteq \mathcal{AG}$, $\mathbf{B}_\alpha\varphi$ and $\mathbf{C}_\alpha\varphi$ denote $\bigwedge_{i\in\alpha} \mathbf{B}_i\varphi$ and $\bigwedge_{k=1}^{\infty} \mathbf{B}_\alpha^k\varphi$, where $\mathbf{B}_\alpha^1\varphi$=$\mathbf{B}_\alpha\varphi$ and $\mathbf{B}_\alpha^k\varphi$=$\mathbf{B}_\alpha^{k-1}\mathbf{B}_\alpha\varphi$ for $k > 1$, respectively. $\mathcal{L}_{\mathcal{AG}}$ denotes the set of belief formulae over $\langle \mathcal{AG}, \mathcal{F}\rangle$.

Satisfaction of belief formulae is defined over *pointed Kripke structures* [10]. A Kripke structure M is a tuple $\langle S, \pi, \{\mathcal{B}_i\}_{i\in\mathcal{AG}}\rangle$, where S is a set of worlds (denoted by $M[S]$), $\pi : S \mapsto 2^{\mathcal{F}}$ is a function that associates an interpretation of $\mathcal{F}$ to each element of S (denoted by $M[\pi]$), and for $i \in \mathcal{AG}$, $\mathcal{B}_i \subseteq S \times S$ is a binary relation over S (denoted by $M[i]$). For convenience, we will often draw a Kripke structure M as a directed labeled graph, whose set of labeled nodes represents S and whose set of labeled edges contains $s \xrightarrow{i} t$ iff $(s, t) \in \mathcal{B}_i$; the label of each node is the name of the world and its interpretation is displayed as a text box next to it (see, e.g., Fig 1). For $u \in S$ and a fluent formula φ, $M[\pi](u)$ and $M[\pi](u)(\varphi)$ denote the interpretation associated to u via π and the truth value of φ with respect to $M[\pi](u)$. For a world $s \in M[S]$, (M, s) is a *pointed Kripke structure*, hereafter called a *state*.

The satisfaction relation $\models$ between belief formulae and a state (M, s) is defined as follows: (1) $(M, s) \models p$ if p is a fluent and $M[\pi](s)(p)$ is true; (2) $(M, s) \models \neg\varphi$ if $(M, s) \not\models \varphi$; (3) $(M, s) \models \varphi_1 \wedge \varphi_2$ if $(M, s) \models \varphi_1$ and $(M, s) \models \varphi_2$; (4) $(M, s) \models \varphi_1 \vee \varphi_2$ if $(M, s) \models \varphi_1$ or $(M, s) \models \varphi_2$; and (5) $(M, s) \models \mathbf{B}_i\varphi$ if $\forall t.[(s, t) \in M[i] \Rightarrow (M, t) \models \varphi]$.

Edge-Conditioned Update Models. The formalism of *update models* has been used to describe transformations of states according to a predetermined transformation pattern (see, e.g., [1,5]). This formalism makes use of the notion of $\mathcal{L}_{\mathcal{AG}}$-substitution, which is a set $\{p_1 \to \varphi_1, \ldots, p_k \to \varphi_k\}$, where each p_i is a distinct fluent in $\mathcal{F}$ and each $\varphi_i \in \mathcal{L}_{\mathcal{AG}}$. $SUB_{\mathcal{L}_{\mathcal{AG}}}$ denotes the set of all $\mathcal{L}_{\mathcal{AG}}$-substitutions. To handle the nested belief about agents' observability problem, in this extension of m$\mathcal{A}^*$, we will utilize the *edge-conditioned event update models* as proposed by [7]. An edge-conditioned event update model $\mathbf{\Sigma}$ is a tuple $\langle \Sigma, \{R_i\}_{i\in\mathcal{AG}}, pre, sub\rangle$ where Σ is a set of events, $R_i \subseteq \Sigma \times \mathcal{L}_{\mathcal{AG}} \times \Sigma$ is the accessibility relation of agent i between events, $pre : \Sigma \to \mathcal{L}_{\mathcal{AG}}$ is a function mapping each event $e \in \Sigma$ to a formula in $\mathcal{L}_{\mathcal{AG}}$, $sub : \Sigma \to SUB_{\mathcal{L}_{\mathcal{AG}}}$ is a function mapping each event $e \in \Sigma$ to a substitution in $SUB_{\mathcal{L}_{\mathcal{AG}}}$. Elements of R_i are of the form (e_1, γ, e_2) where γ is a belief formula. In the graph representation, such an accessibility relation is shown by a directed edge from e_1 to e_2 with the label $i : \gamma$. We will omit γ and write simply i as label of the edge when $\gamma = \top$.

Given an edge-conditioned update model $\mathbf{\Sigma}$, an *update instance* ω is a pair $(\mathbf{\Sigma}, e)$ where e is an event in Σ, referred to as a *designated event* (or *true event*). For simplicity of presentation, we often draw an update instance as a graph whose events are rectangles, whose links represent the accessibility relations between events, and a double square represents the *designated event* (see, e.g., Fig. 2).

Given a Kripke structure M and an edge-conditioned update model $\mathbf{\Sigma} = \langle \Sigma, \{R_i\}_{i\in\mathcal{AG}}, pre, sub\rangle$, the *update* of M induced by $\mathbf{\Sigma}$ results in a new Kripke structure M', denoted by $M' = M \otimes \mathbf{\Sigma}$, defined by: **(i)** $M'[S] = \{(s, \tau) \mid \tau \in \Sigma, s \in M[S], (M, s) \models pre(\tau)\}$; **(ii)** $((s, \tau), (s', \tau')) \in M'[i]$ iff (s, τ),

$(s', \tau') \in M'[S]$, $(s, s') \in M[i]$, $(\tau, \gamma, \tau') \in R_i$ and $(M, s) \models \gamma$; and **(iii)** for all $(s, \tau) \in M'[S]$ and $f \in \mathcal{F}$, $M'[\pi]((s, \tau)) \models f$ if $f \rightarrow \varphi \in sub(\tau)$ and $(M, s) \models \varphi$.

An *update template* is a pair $(\mathbf{\Sigma}, \Gamma)$, where $\mathbf{\Sigma}$ is an update model with the set of events Σ and $\Gamma \subseteq \Sigma$. The update of a state (M, s) given an update template $(\mathbf{\Sigma}, \Gamma)$ is a set of states, denoted by $(M, s) \otimes (\mathbf{\Sigma}, \Gamma)$, where $(M, s) \otimes (\mathbf{\Sigma}, \Gamma) = \{(M \otimes \mathbf{\Sigma}, (s, \tau)) \mid \tau \in \Gamma, (M, s) \models pre(\tau)\}$.

Syntax of m$\mathcal{A}^*$. An action theory in the language m$\mathcal{A}^*$ over $\langle \mathcal{AG}, \mathcal{F} \rangle$ consists of a set of action instances $\mathcal{AI}$ of the form $a\langle\alpha\rangle$, representing that a set of agents α performs action a, and a collection of statements of the following forms:

a	**executable_if**	ψ	(1)	a	**announces**	φ	(4)
a	**causes** ℓ **if**	φ	(2)	z	**observes** a **if**	δ_z	(5)
a	**determines**	φ	(3)	z	**aware_of** a **if**	θ_z	(6)

where ℓ is a fluent literal (a fluent $f \in \mathcal{F}$ or its negation $\neg f$), ψ is a belief formula, φ, δ_z and θ_z are fluent formulae, $\mathsf{a} \in \mathcal{AI}$, and $z \in \mathcal{AG}$. (1) encodes the executability condition of a. (2) describes the effect of the ontic (i.e., world-changing) action a. (3) enables the agents who execute a to learn the value of the formula φ. (4) encodes an *announcement* action, whose owner announces that φ is true. (5) indicates that agent z is a full observer of a if δ_z holds. (6) states that agent z is a partial observer of a if θ_z holds. It is assumed that the sets of ontic actions, sensing actions, and announcement actions are pairwise disjoint. Furthermore, for every pair of a and z, if z and a occur in a statement of the form (5) then they do not occur in any statement of the form (6) and vice versa. An action domain is a collection of statements (1)–(6). An action theory is a pair of an action domain and a set of statements of the form "**initially** ψ", indicate that ψ is true in the initial state. By this definition, action domains are deterministic in that each ontic action, when executed in a world, results in a unique world.

3 Edge-Conditioned Event Update Models

In this section, we will show how to define the transition function m$\mathcal{A}^*$ using edge-conditioned event update models. We will use Example 1 as a running example to illustrate the application of edge-conditioned event update models. We follow the same notation and rules as in Sect. 2.

Let us denote the multi-agent domain described in Example 1 by D_{coin}. For this domain, we have that $\mathcal{AG} = \{A, B, C\}$. The set of fluents $\mathcal{F}$ for this domain consists of *head* (the coin is heads up), $looking_x$ (agent x is looking at the box where $x \in \{A, B, C\}$), and *opened* (the box is open). D_{coin} has two actions: *open* and *peek*, that can be represented by the following m$\mathcal{A}^*$ statements:

$$open\langle x\rangle \textbf{ causes } opened \quad (7)$$
$$peek\langle x\rangle \textbf{ executable\_if } opened \quad (8)$$
$$peek\langle x\rangle \textbf{ determines } head \quad (9)$$
$$x \textbf{ observes } open\langle x\rangle \quad (10)$$
$$y \textbf{ observes } open\langle x\rangle \textbf{ if } looking_y \quad (11)$$
$$x \textbf{ observes } peek\langle x\rangle \quad (12)$$
$$y \textbf{ aware\_of } peek\langle x\rangle \textbf{ if } looking_y \quad (13)$$

where $x, y \in \{A, B, C\}$ and $x \neq y$.

Initially, the coin is heads up, the box is closed and A, B are looking at it; however, B thinks that all three agents are looking at the box. The initial state (M_0, s_0) of D_{coin} is on the left of Fig. 1. Suppose that agent A would like to know whether the coin lies heads or tails up. She would also like to let agent B know that she knows this fact. However, she would like to make B also thinks that agent C is aware of this fact. Intuitively, because B has already believed that C is looking at the box, agent A could achieve her goals by: (1) opening the box; and then (2) peeking into the box.

Observe that under the current semantics of m$\mathcal{A}^*$ [4], A could not achieve her goal by executing the above sequence of actions. This is because B believes that C does not know that the box is open (as showed in Fig. 1), therefor B would conclude that C is not observing the execution of the action of A opening the box. When A peeks into the box, B reasons that C is still thinking that A knows nothing because, according to B, C still believes that the box is closed. Therefore, B will think that C's belief about A's belief about the state of the coin does not change. This is not intuitive. A more intuitive outcome with respect to B's beliefs after the execution of the plan $[open\langle A\rangle; peek\langle A\rangle]$ is as follows: B should believe that C knows that the box is open and that A knows the value of the coin after the execution of the plan.

The main reason for the above inadequacy of m$\mathcal{A}^*$ lies in the fact that the construction of the update models in m$\mathcal{A}^*$ assumes that full observers have the correct observability of *all agents,* which is not the case for B, who is a full observer, and C: B believes that C is a full observer while C is not. One possible way to address the above issue is to create different update models, whose set of events depends on the effects of actions, as done in [13], or to define transition functions by directly manipulating the accessibility relations and the worlds in the resulting Kripke structure as in [8]. In this paper, we introduce a different approach to this problem, through the use of edge-conditioned update models.

3.1 Ontic Actions

We assume that an action domain D is given. As in m$\mathcal{A}^*$, we assume that an agent can either observe or not observe the execution of an ontic action, i.e., for an ontic action instance a, there exists no statement of the form (6) whose action is a.

Definition 1 (Ontic Actions). *Let* a *be an ontic action instance with the precondition* ψ. *The update model for* a, *denoted by* $\omega(\mathsf{a})$, *is defined by* $\langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub\rangle$ *where: (1)* $\Sigma = \{\sigma, \epsilon\}$*; (2)* $R_i = \{(\sigma, \delta_i, \sigma), (\sigma, \neg\delta_i, \epsilon), (\epsilon, \top, \epsilon)\}$ *where*

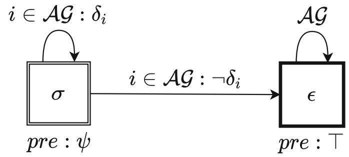

Fig. 2. Edge-conditioned update model for an ontic action

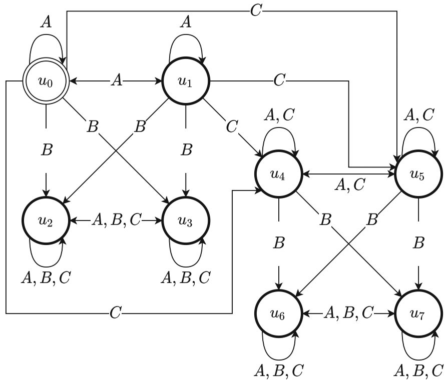

Fig. 3. (M_1, v_0) after A opened the box using edge-conditioned update model

"i **observes** a **if** δ_i*" belongs to D; (3) pre*$(\sigma)=\psi$ *and pre*$(\epsilon)=\top$*; and (4) sub*$(\epsilon)=\emptyset$ *and sub*$(\sigma) = \{p \rightarrow \Psi^+(p, \mathsf{a}) \vee (p \wedge \neg\Psi^-(p, \mathsf{a})) \mid p \in \mathcal{F}\}$*, where* $\Psi^+(p, \mathsf{a}) = \bigvee\{\varphi \mid [\mathsf{a}\ \textbf{causes}\ p\ \textbf{if}\ \varphi] \in D\}$ *and* $\Psi^-(p, \mathsf{a}) = \bigvee\{\varphi \mid [\mathsf{a}\ \textbf{causes}\ \neg p\ \textbf{if}\ \varphi] \in D\}$.

When an ontic action occurs, an agent may or may not observe its occurrence. As such, $\omega(\mathsf{a})$ has two events. σ is the designated event representing the true occurrence of the action whereas ϵ denotes the null event representing that the action does not occur. σ is the event full observers believe occurring and ϵ is the event seen by oblivious agents. Figure 2 shows the edge-conditioned update model of an ontic action a. In the figure, we use $i \in X : \delta_i$ as a shorthand for the set of links with labels $\{i : \delta_i \mid i \in X\}$.

Observe that the presence of the condition attached to the link and the definition of the cross product between a Kripke structure and the update model enable a flexible update of the accessibility relations, allowing us to eliminate the problem of the definition in [4]. For example, given a state (M, s), the link $(\sigma, i : \delta_i, \sigma)$ in $\omega(\mathsf{a})$ indicates that $((\sigma, s), i, (\sigma, s))$ is an element in the accessibility relation of i in the state resulting from the execution of a in (M, s) iff $(M, s) \models \delta_i$.

The update induced by the edge-conditioned update model for *open*$\langle A\rangle$ on the pointed Kripke structure at the left of Fig. 1 is shown in Fig. 3. In this figure, the worlds and their interpretations are the same as in the pointed Kripke structure at the right of Fig. 1. The differences lie in the removal of the links labeled C from u_2/u_3 to u_6/u_7 and the addition of the loops labeled C at u_2/u_3. The loops labeled C at u_2 and u_3, denoting the worlds (s_2, σ) and (s_3, σ), respectively, are added because $(M_0, s_2) \models looking_C$ and $(M_0, s_3) \models looking_C$ hold. This is also the reason for the removal of the links labeled C from u_2 and u_3 to u_6 and u_7.

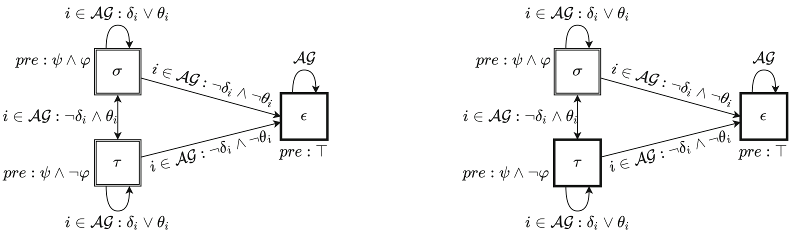

Fig. 4. Edge-conditioned update models for sensing action (left) and truthful announcement Action (right)

3.2 Sensing Actions and Announcement Actions

An agent can either observe, partially observe, or not observe the occurrence of a sensing or announcement action occurrence. Therefore, the update models for sensing or announcement actions are different from that of update models for ontic actions. They are defined as follows.

Definition 2 (Sensing and Announcement Actions). *Let* a *be a sensing action instance that senses* φ *or an announcement action instance that announces* φ *with the precondition* ψ. *The update model for* a, *denoted by* $\omega(\mathsf{a})$, *is defined by* $\langle \Sigma, \{R_i\}_{i \in \mathcal{AG}}, pre, sub \rangle$ *where: (1)* $\Sigma = \{\sigma, \tau, \epsilon\}$; *(2)* $R_i = \{(\sigma, \delta_i \vee \theta_i, \sigma), (\tau, \delta_i \vee \theta_i, \tau), (\sigma, \neg\delta_i \wedge \theta_i, \tau), (\tau, \neg\delta_i \wedge \theta_i, \sigma), (\sigma, \neg\delta_i \wedge \neg\theta_i, \epsilon), (\tau, \neg\delta_i \wedge \neg\theta_i, \epsilon), (\epsilon, \top, \epsilon)\}$ *where "i* **observes** a **if** δ_i*" and "i* **aware_of** a **if** θ_i*" belong to D; (3)* $pre(\sigma) = \psi \wedge \varphi$, $pre(\tau) = \psi \wedge \neg\varphi$ *and* $pre(\epsilon) = \top$; *and (4)* $sub(x) = \emptyset$ *for each* $x \in \Sigma$.

Observe that an update model of a sensing or announcement action instance has three events. However in sensing actions, the true event can be σ or τ whereas in announcement actions, the true event is σ. As for ontic actions, ϵ is the "null event" representing that the action does not occur. Sensing and announcement actions do not alter the state of the world and thus *sub* is empty for every event. Figure 4 illustrates the edge-conditioned update model for an announcement a that truthfully announces φ (right) and a sensing action a that determines φ (left).

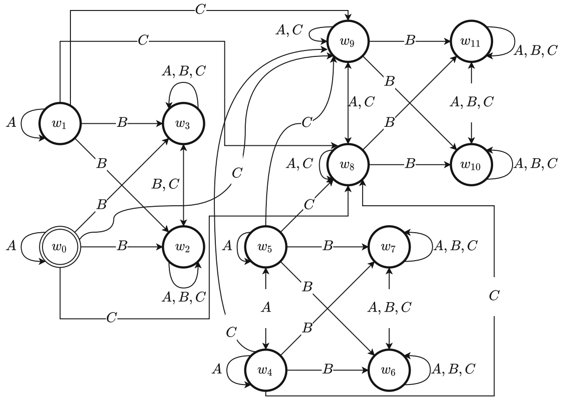

Fig. 5. (M_2, w_0) after A peeked into the box

The application of the edge-conditioned update model for $peek\langle A\rangle$ in the state (M_1, u_0) from Fig. 3 is given in Fig. 5. In this figure, w_0-w_3 have the same interpretation as u_0-u_3; and w_4-w_{11} have the same interpretation as u_0-u_7. Observe that A now achieves her goals: not only does A realize that the coin lies head up ($(M_2, w_0) \models \mathbf{B}_A head$) but B also believes that C knows the fact that A knows the value of the coin now ($(M_2, w_0) \models \mathbf{B}_B\mathbf{B}_C(\mathbf{B}_A head \vee \mathbf{B}_A \neg head)$). This example shows that the use of edge-conditioned update models enables m$\mathcal{A}^*$ to avoid the side problem discussed in [4].

Having defined the update models for actions in a multi-agent domain D, we can define the transition function Φ_D in D in the similar fashion as in [4]. We omit the details for brevity.

3.3 Properties of Edge-Conditioned Update Models

The use of edge-conditioned update models enables the modification of the semantics of m$\mathcal{A}^*$ that takes into consideration the observability of the agents at the local level. A consequence of this treatment is that the belief of an agent i about the belief of another agent j with respect to the action occurrence will change in accordance to the belief of i about j before. The following proposition indicates these properties of edge-conditioned update models.

Proposition 1. *Let* (M, s) *be a state and* a *be an ontic action instance that is executable in* (M, s) *and* $\omega(\mathsf{a})$ *be given in Definition 1. It holds that:*

1. *For every agent* $x \in \mathcal{AG}$, $[x$ **observes** a **if** $\delta_x]$ *and* $[\mathsf{a}$ **causes** ℓ **if** $\varphi]$ *belong to* D, *if* $(M, s) \models \delta_x$, $(M, s) \models \mathbf{B}_x\varphi$ *and* $(M', s') = (M, s) \otimes (\omega(\mathsf{a}), \sigma)$ *then* $(M', s') \models \mathbf{B}_x\ell$.
2. *For every pair of agents* $x, y \in \mathcal{AG}$, $[\mathsf{a}$ **causes** ℓ **if** $\varphi]$, $[x$ **observes** a **if** $\delta_x]$ *and* $[y$ **observes** a **if** $\delta_y]$ *belong to* D, *if* $(M, s) \models \delta_x$, $(M, s) \models \mathbf{B}_x\delta_y$, $(M, s) \models \mathbf{B}_x\mathbf{B}_y\varphi$ *and* $(M', s') = (M, s) \otimes (\omega(\mathsf{a}), \sigma)$ *then* $(M', s') \models \mathbf{B}_x\mathbf{B}_y\ell$.

3. *For every pair of agents* $x, y \in \mathcal{AG}$*, a belief formula* η*,* [x **observes** a **if** δ_x] *and* [y **observes** a **if** δ_y] *belong to* D*, if* $(M, s) \models \delta_x$*,* $(M, s) \models \mathbf{B}_x \neg \delta_y$*,* $(M, s) \models \mathbf{B}_x \mathbf{B}_y \eta$ *and* $(M', s') = (M, s) \otimes (\omega(\mathsf{a}), \sigma)$ *then* $(M', s') \models \mathbf{B}_x \mathbf{B}_y \eta$*.*

Proof. All proofs are omitted for lack of space and detailed in [12].

The second and third item of Proposition 1 show that a full observer will update her beliefs about another agent's beliefs, if she thinks that the agent is also a full observer, or her own beliefs about the other agent will not change if she believes that such agent is unaware of the action occurrence. These two items do not hold w.r.t. the old semantics of m$\mathcal{A}^*$. Similar propositions can be established for sensing/announcement actions and can be found in [12].

4 Discussion and Related Work

Update models have been used in formalizing actions in multi-agent domains by several authors [1,2,5,6,9,11]. However, the automatic generation of update models from an action description has only been discussed with the introduction of actions languages for multi-agent domains in [3] and subsequent versions of languages like m$\mathcal{A}^*$. To the best of our knowledge, the present work is among the first that attempts to use edge-conditioned update models, introduced by [7], in an action language. As shown in Proposition 1, the use of edge-conditioned update models eliminates the problem encountered by earlier semantics of m$\mathcal{A}^*$. We note that [7] discussed only edge-conditioned update models for world-altering actions (ontic actions), while we use it in modeling other types of actions as well.

The use of event models for reasoning about effects of actions in multi-agent domains is also studied in [13] within the language DER. In this language, the observability of agents is encoded by an observations set $\mathcal{O}$ and no distinction between ontic, sensing, and announcement actions is made. Comparing with the update models used in [13], we can see that updated models used in the present paper have a fixed number of events, given the type of the action: two events for ontic actions and three events for sensing/announcement actions. On the other hand, the number of events in DER can vary given the number of statements specifying its effects and observations. We believe that this feature might bring some advantages if update models are used for planning, where efficient construction of update models is critical (in the new m$\mathcal{A}^*$, the model need to be constructed only once).

5 Conclusion

We define a new semantics for the high-level action language m$\mathcal{A}^*$ using edge-conditioned update models. In this method, edge-conditioned update models are constructed directly from the domain specification and are independent from the states in which the action occurs. We prove that the new semantics satisfies a desirable property that second order beliefs of agents about other agents' beliefs change consistently with its first-order beliefs about observability of action occurrence, i.e., it overcomes a problem of the earlier semantics of m$\mathcal{A}^*$. This overcomes a limitation of the earlier version of m$\mathcal{A}^*$, which requires a careful domain design for dealing with certain types of questions as described in [4].

Acknowledgments. The authors have been partially supported by NSF grants 2151254, 1914635 and 1757207. Tran Cao Son was also partially supported by NSF grant 1812628.

References

1. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese **139**, 165–224 (2004)
2. Baltag, A., Moss, L., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: 7th TARK, pp. 43–56 (1998)
3. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: Reasoning about the beliefs of agents in multi-agent domains in the presence of state constraints: the action language mAL. In: Leite, J., Son, T.C., Torroni, P., van der Torre, L., Woltran, S. (eds.) CLIMA 2013. LNCS (LNAI), vol. 8143, pp. 290–306. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40624-9_18
4. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for multi-agent domains. Artif. Intell. **302**, 103601 (2022)
5. van Benthem, J., van Eijck, J., Kooi, B.P.: Logics of communication and change. Inf. Comput. **204**(11), 1620–1662 (2006)
6. Bolander, T., Andersen, M.: Epistemic planning for single and multi-agent systems. J. Appl. Non-Classical Logics **21**(1), 9–34 (2011)
7. Bolander, T.: Seeing is believing: formalising false-belief tasks in dynamic epistemic logic. In: van Ditmarsch, H., Sandu, G. (eds.) Jaakko Hintikka on Knowledge and Game-Theoretical Semantics. OCL, vol. 12, pp. 207–236. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62864-6_8
8. Buckingham, D., Kasenberg, D., Scheutz, M.: Simultaneous representation of knowledge and belief for epistemic planning with belief revision, pp. 172–181 (2020)
9. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, 1st edn. Springer, Heidelberg (2007). https://doi.org/10.1007/978-1-4020-5839-4
10. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT press, Cambridge (1995)
11. Löwe, B., Pacuit, E., Witzel, A.: DEL planning and some tractable cases. In: van Ditmarsch, H., Lang, J., Ju, S. (eds.) LORI 2011. LNCS (LNAI), vol. 6953, pp. 179–192. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24130-7_13
12. Pham, L., Izmirlioglu, Y., Son, T.C., Pontelli, E.: A new semantics for the action language ma*. Technical report, NMSU (2022). https://github.com/phhuuloc/New-semantic-mAstar
13. Rajaratnam, D., Thielscher, M.: Representing and reasoning with event models for epistemic planning. In: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, pp. 519–528 (11 2021)