An Action Language for Multi-Agent Domains

Chitta Baral[†], Gregory Gelfond[‡], Enrico Pontelli[⋄], Tran Cao Son[⋄⋆]

[†]*Department of Computer Science and Engineering, Arizona State University*
[‡]*Department of Computer Science, University of Nebraska Omaha*
[⋄]*Department of Computer Science, New Mexico State University*
[⋆]*Corresponding author*

Abstract

The goal of this paper is to investigate an action language, called m$\mathcal{A}^*$, for representing and reasoning about actions and change in multi-agent domains. The language, as designed, can also serve as a specification language for epistemic planning, thereby addressing an important issue in the development of multi-agent epistemic planning systems. The m$\mathcal{A}^*$ action language is a generalization of the single-agent action languages, extensively studied in the literature, to the case of *multi-agent domains.* The language allows the representation of different types of actions that an agent can perform in a domain where many other agents might be present—such as world-altering actions, sensing actions, and communication actions. The action language also allows the specification of agents' *dynamic awareness* of action occurrences—which has implications on what agents' know about the world and other agents' knowledge about the world. These features are embedded in a language that is simple, yet powerful enough to address a large variety of knowledge manipulation scenarios in multi-agent domains.

The semantics of m$\mathcal{A}^*$ relies on the notion of *state,* which is described by a *pointed Kripke model* and is used to encode the agents' knowledge[1] and the real state of the world. The semantics is defined by a transition function that maps pairs of actions and states into sets of states. The paper presents a number of properties of the action theories and relates m$\mathcal{A}^*$ to other relevant formalisms in the area of reasoning about actions in multi-agent domains.

Keywords: Action Languages, Epistemic Planning, Reasoning about Knowledge

1. Introduction

1.1. Motivations

Reasoning about Actions and Change (RAC) has been a research focus since the early days of artificial intelligence (McCarthy, 1959), and languages for representing actions and their effects have been proposed very early in the AI literature (Fikes and Nilson, 1971). While the early papers on this topic by Fikes and Nilson (1971) did not include formal semantics, papers with formal semantics

Email address: chitta@asu.edu, ggelfond@unomaha.edu, epontell@cs.nmsu.edu, tson@cs.nmsu.edu (Chitta Baral[†], Gregory Gelfond[‡], Enrico Pontelli[⋄], Tran Cao Son[⋄⋆])

[1]We will use the term "knowledge" to mean both "knowledge" and "beliefs" when clear from the context.

came some years after, with leading efforts by Lifschitz (1987). The approach adopted in this paper is predominantly influenced by the methodology for representing and reasoning about actions and change proposed by Gelfond and Lifschitz (1993). In this approach, actions of agents are described in a high-level language, with an English-like syntax and a transition function-based semantics. Action languages offer several benefits, including a succinct way for representing dynamic domains. The approach proposed in this paper is also related to action description languages developed for planning, such as (Pednault, 1989; Ghallab et al., 1998).

Over the years, several action languages (e.g., $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$) have been developed, as discussed in (Gelfond and Lifschitz, 1998). Each of these languages addresses some important problems in RAC, e.g., the ramification problem, concurrency, actions with duration, and knowledge of agents. Action languages have also provided the foundations for several successful approaches to automated planning. For example, the language $\mathcal{C}$ is used in the planner C-PLAN (Castellini et al., 2001) and the language $\mathcal{B}$ is used in CPA (Son et al., 2005). The *Planning Domain Definition Language (PDDL)* (Ghallab et al., 1998), the de-facto standard language for planning systems, could also be viewed as a type of action language (a generalization of STRIPS). However, the primary focus of all these research efforts has been about reasoning within *single-agent domains.*

In single-agent domains, reasoning about actions and change mainly involves reasoning about what is true in the world, what the agent knows about the world, how the agent can manipulate the world (using world-changing actions) to reach particular states, and how the agent (using sensing actions) can learn unknown aspects of the world. In *multi-agent domains*, an agent's action may not just change the world and the agent's knowledge about the world, but also may change other agents' knowledge. Similarly, the goals of an agent in a multi-agent world may involve manipulating the knowledge of other agents—in particular, this may involve not just their knowledge about the world, but also their knowledge about other agents' knowledge about the world. Although there is a large body of research on multi-agent planning (see, e.g., (Durfee, 1999; de Weerdt et al., 2003; de Weerdt and Clement, 2009; Allen and Zilberstein, 2009; Bernstein et al., 2002; Goldman and Zilberstein, 2004; Guestrin et al., 2001; Nair et al., 2003; Peshkin and Savova, 2002)), relatively few efforts address the above aspects of multi-agent domains, which offer a number of new research challenges in representing and reasoning about actions and change. The following simple example illustrates some of these issues.

Example 1 (Three Agents and the Coin Box)**.** *Three agents, A, B, and C, are in a room. In the middle of the room there is a box containing a coin. It is common knowledge that:*

- *None of the agents knows whether the coin lies heads or tails up;*
- *The box is locked and one needs a key to open it; agent A has the key of the box and everyone knows this;*
- *In order to learn whether the coin lies heads or tails up, an agent can peek into the box—but this requires the box to be open;*
- *If one agent is looking at the box and a second agent peeks into the box, then the first agent will observe this fact and will be able to conclude that the second agent knows the status of the coin; on the other hand, the first agent's knowledge about which face of the coin is up does not change;*
- *Distracting an agent causes that agent to not look at the box;*

- *Signaling an agent to look at the box causes such agent to look at the box;*
- *Announcing that the coin lies heads or tails up will make this a common knowledge among the agents that are listening.*

Suppose that the agents A and B are allied and A would like to know whether the coin lies heads or tails up. She would also like to let the agent B know that she knows this fact. However, she would like to keep this information secret from C. Observe that the last two sentences express goals that are about agents' knowledge about other agents' knowledge. Intuitively, she could achieve her goals by:

1. *Distracting C from looking at the box;*
2. *Signaling B to look at the box if B is not looking at the box;*
3. *Opening the box; and*
4. *Peeking into the box.* □

This simple story presents a number of challenges for research in representing and reasoning about actions and their effects in multi-agent domains. In particular:

- The domain contains several types of actions:

  1. Actions that allow the agents to change the state of the world (e.g., opening the box);
  2. Actions that change the knowledge of the agents (e.g, peeking into the box, announcing heads/tails);
  3. Actions that manipulate the beliefs of other agents (e.g., peeking while other agents are looking); and
  4. Actions that change the observability of agents with respect to awareness about future actions (e.g., distract and signal actions before peeking into the box).

  We observe that the third and fourth types of actions are not considered in single agent systems.

- The reasoning process that allows agent A to verify that steps (1)-(4) will indeed achieve her goal requires A's ability to reason about the effects of actions on several aspects:

  1. *The state of the world*—e.g., opening the box causes the box to become open;
  2. *The agents' awareness of the environment and of other agents' actions*—e.g., distracting or signaling an agent causes this agent not to look or to look at the box, respectively; and
  3. *The knowledge of other agents about her own knowledge*—e.g., someone following her actions would know what she knows.

  While the first requirement is the same as for an agent in single-agent domains, the last two are specific to multi-agent domains.

With respect to planning, the above specifics of multi-agent systems raise an interesting problem:

"How can one generate a plan for the agent A to achieve her goal, given the description in Example 1?"

The above problem is an *Epistemic Planning problem in a Multi-agent domain (EPM)* (Bolander and Andersen, 2011), which refers to the generation of plans for multiple agents to achieve goals which can refer to the state of the world, the beliefs of agents, and/or the knowledge of agents. EPM has recently attracted the attention of researchers from various communities, such as planning, dynamic epistemic logic, and knowledge representation. The Dagstuhl seminars on the subject (Agotnes et al., 2014; Baral et al., 2017) provided the impetus for the development of several epistemic planners (Kominis and Geffner, 2015; Huang et al., 2017; Muise et al., 2015; Wan et al., 2015; Liu and Liu, 2018; Le et al., 2018) and extensive studies of the theoretical foundations (e.g., decidability and computational complexity) of EPM (Aucher and Bolander, 2013; Bolander et al., 2015). In spite of all these efforts, to the best of our knowledge, only two systems have been proposed that address the complete range of issues mentioned in Example 1: the dynamic epistemic modeling system called DEMO van Eijck (2004) and the recently proposed system described in (Le et al., 2018; Fabiano et al., 2020). This is in stark contrast to the landscape of automated planning for single-agent domains, where we can find several efficient automated planners capable of generating plans consisting of hundreds of actions within seconds—especially building on recent advances in search-based planning.

Among the main reasons for the lack of planning systems capable of dealing with the issues like those shown in Example 1 are: **(i)** the lack of action-based formalisms that can address the above mentioned issues and that can actually be orchestrated, and **(ii)** the fact that logical approaches to reasoning about knowledge of agents in multi-agent domains are mostly model-theoretical, and not amenable to an implementation in search-based planning systems. Indeed, both issues were raised in the recent Dagstuhl seminar (Baral et al., 2017). The issue **(i)** is considered as one of the main research topics in EPM, while **(ii)** is related to the practical and conceptual knowledge representation challenges—discussed by Herzig[2] at the second Dagstuhl seminar (Baral et al., 2017). We will discuss these issues in more detail in the next sub-section.

1.2. Related Work

In terms of related work, multi-agent actions have been explored in *Dynamic Epistemic Logics (DEL)* (e.g., Baltag and Moss (2004); Herzig et al. (2005); van Benthem (2007); van Benthem et al. (2006); van Ditmarsch et al. (2007)). However, as discussed later in the paper, DEL does not offer an intuitive view of how to orchestrate or execute a single multi-agent action. In addition, the complex representation of multi-agent actions—similar to a Kripke structure—drastically increases the number of possible multi-agent actions—thus, making it challenging to adopt a search-based approach in developing multi-agent action sequences to reach a given goal. It can be observed that several approaches to epistemic planning in multi-agent domains with focus on knowledge and beliefs of agents did employ an extension of PDDL rather than using DEL (Kominis and Geffner, 2015; Huang et al., 2017; Muise et al., 2015; Wan et al., 2015; Liu and Liu, 2018).

[2] `http://materials.dagstuhl.de/files/17/17231/17231.AndreasHerzig.Slides.pdf`

The research in DEL has also not addressed some critical aspects of multi-agent search-based planning, such as the determination of the initial state of a planning domain instance. Moreover, research in DEL did not explore the link between the state of the world and the observability encoded in multi-agent actions, and hence preventing the dynamic evolution of the observational capabilities and awareness of the agents with respect to future actions. In some ways, the DEL approach is similar to the formulation of belief updates (e.g., (Friedman and Halpern, 1999; Katsuno and Mendelzon, 1992; del Val A. and Shoham, 1994)), and most of the differences and similarities between belief updates and reasoning about actions carry over to the differences and similarities between DEL and our formulation of RAC in multi-agent domains. We will elaborate on these differences in a later section of the paper.

1.3. Contributions and Assumptions

Our goal in this paper is to develop a framework that allows reasoning about actions and their effects in a multi-agent domain; the framework is expected to address the above-mentioned issues, e.g., actions' capability to modify agents' knowledge and beliefs about other agents' knowledge and beliefs. To this end, we propose a high-level action language for representing and reasoning about actions in multi-agent domains. The language provides the fundamental components of a planning domain description language for multi-agent systems. The main contributions of the paper are:

- The action language m$\mathcal{A}^*$, which allows the representation of different types of actions—such as world-altering actions, announcement actions, and sensing actions—for formalizing multi-agent domains; the language explicitly supports actions for the dynamic modification of the awareness and observation capabilities of the agents;

- A transition function-based semantics for m$\mathcal{A}^*$, that enables hypothetical reasoning and planning in multi-agent domains. This, together with the notion of a *finitary*-**S5** *theories* for representing the initial state, introduced in (Son et al., 2014), provides a foundation for the implementation of heuristic search-based planners for domains described in m$\mathcal{A}^*$; and

- Several theoretical results relating the semantics of m$\mathcal{A}^*$ to multi-agent actions characterizations using the notion of update models from DEL Baltag and Moss (2004).

In developing m$\mathcal{A}^*$, we make several design decisions. The key decision is that actions in our formalism can be effectively executed and the outcome can be effectively determined. This is not the case, for example, in DEL (van Ditmarsch et al., 2007), where actions are complex graph structures, similar to Kripke structures, possibly representing a multi-modal formula, and it is not clear if and how such actions can be executed.[3] We also assume that actions are deterministic, i.e., the result

[3]Update models in the literature typically come with an intuitive description of the real scenario describing who executes the action, who observes (or partially observes) the action occurrence, and who is oblivious. Without such information, an update model can be interpreted in many ways. As an example, simply looking at the right most update model of Fig. 2 would not tell us whether A, B, or C executes the action. This is because this update model can also be used to represent the scenario when B makes a truthful announcement that the coin lies heads up while A is attentive and C is not attentive.

of the execution of a world altering action is unique. This assumption can be lifted in a relatively simple manner—by generalizing the techniques for handling non-deterministic actions studied in the context of single-agent domains.

Although we have mentioned both knowledge and beliefs, in this paper we will follow van Ditmarsch et al. (2007); Baltag and Moss (2004) and focus only on formalizing the changes of *beliefs* of agents after the execution of actions. Following the considerations by van Benthem (2007), the epistemic operators used in this paper can be read as *"to the best of my information."* Note that, in a multi-agent system, there may be a need to distinguish between the *knowledge* and the *beliefs* of an agent about the world. Let us consider Example 1 and let us denote with p the proposition *"nobody knows whether the coin lies heads or tails up."* Initially, the three agents know that p is true. However, after agent A executes the sequence of actions (1)-(4), A will know that p is false. Furthermore, B also knows that p is false, thanks to her awareness of A's execution of the actions of opening the box and looking into it. However, C, being unaware of the execution of the actions performed by A, will still believe that p is true. If this were considered as a part of C's knowledge, then C would result in having false knowledge.

The investigation of the discrepancy between knowledge and beliefs has been an intense research topic in dynamic epistemic logic and in reasoning about knowledge, which has lead to the development of several modal logics (e.g., (Fagin et al., 1995; van Ditmarsch et al., 2007)). Since our main aim is the development of an action language for hypothetical reasoning and planning, we will be primarily concerned with the beliefs of agents. Some preliminary steps in this direction have been explored in the context of the DEL framework (Herzig et al., 2005; Son et al., 2015). We leave the development of an action-based formalism that takes into consideration the differences between beliefs and knowledge as future work.

1.4. Paper Organization

The rest of the paper is organized as follows. Section 2 reviews the basic definitions and notation of a modal logic with belief operators and the update model based approach to reasoning about actions in multi-agent domains. This section also reviews the definition of finitary **S5**-theories whose models are finite. It also includes a short discussion for the development of m$\mathcal{A}^*$. Section 3 presents the syntax of m$\mathcal{A}^*$. Section 4 explores the modeling of the semantics of m$\mathcal{A}^*$ using the update models approach; we define the transition function of m$\mathcal{A}^*$ which maps pairs of actions and states into states; the section also presents the entailment relation between m$\mathcal{A}^*$ action theories and queries along with relevant properties. Section 5 provides an analysis of m$\mathcal{A}^*$ with respect to the existing literature, including a comparison with DEL. Section 6 provide some concluding remarks and directions for future work. For simplicity of presentation, the proofs of the main theorems are placed in Appendix A.

2. Preliminaries

We begin with a review of the basic notions from the literature on formalizing knowledge and reasoning about effects of actions in multi-agent systems. Section 2.1 presents the notion of Kripke structures. Section 2.2 reviews the notion of update models developed by the dynamic epistemic logic community for reasoning about effects of actions in multi-agent systems.

2.1. Belief Formulae and Kripke Structures

Let us consider an environment with a set $\mathcal{AG}$ of n agents. The *real state of the world* (or *real state*, for brevity) is described by a set $\mathcal{F}$ of propositional variables, called *fluents*. We are concerned with the beliefs of agents about the world and about the beliefs of other agents. For this purpose, we adapt the logic of knowledge and the notations used in (Fagin et al., 1995; van Ditmarsch et al., 2007). We associate to each agent $i \in \mathcal{AG}$ a modal operator $\mathbf{B}_i$ (to indicate a belief of agent i) and represent the beliefs of an agent as belief formulae in a logic extended with these operators. Formally, we define belief formulae using the BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \mathbf{B}_i\varphi \mid \mathbf{E}_\alpha\varphi \mid \mathbf{C}_\alpha\varphi$$

where $p \in \mathcal{F}$ is a fluent, $i \in \mathcal{AG}$, and $\emptyset \neq \alpha \subseteq \mathcal{AG}$. We often use *a fluent formula* to refer to a belief formula which does not contain any occurrence of $\mathbf{B}_i$, $\mathbf{E}_\alpha$, and $\mathbf{C}_\alpha$.

Formulae of the form $\mathbf{E}_\alpha\varphi$ and $\mathbf{C}_\alpha\varphi$ are referred to as *group formulae.* Whenever $\alpha = \mathcal{AG}$, we simply write $\mathbf{E}\varphi$ and $\mathbf{C}\varphi$ to denote $\mathbf{E}_\alpha\varphi$ and $\mathbf{C}_\alpha\varphi$, respectively. Let us denote with $\mathcal{L}_{\mathcal{AG}}$ the language of the belief formulae over $\mathcal{F}$ and $\mathcal{AG}$.

Intuitively, belief formulae are used to describe the beliefs of one agent concerning the state of the world as well as about the beliefs of other agents. For example, the formula $\mathbf{B}_1\mathbf{B}_2p$ expresses the fact that "*Agent 1 believes that agent 2 believes that p is true*," while $\mathbf{B}_1 f$ states that "*Agent 1 believes that f is true*."

In what follows, we will simply talk about "formulae" instead of "belief formulae," whenever there is no risk of confusion. In order to define the semantics of such logic language, we need to introduce the notion of a Kripke structure.

Definition 1 (Kripke Structure)**.** *A Kripke structure is a tuple $\langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n\rangle$, where*

- *S is a set of worlds,*
- *$\pi : S \mapsto 2^{\mathcal{F}}$ is a function that associates an interpretation of $\mathcal{F}$ to each element of S, and*
- *For $1 \leq i \leq n$, $\mathcal{B}_i \subseteq S \times S$ is a binary relation over S.*

A pointed Kripke structure *is a pair (M, s) where $M = \langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n\rangle$ is a Kripke structure and $s \in S$. In a pointed Kripke structure (M, s), we refer to s as the* real *(or* actual*) world.*

For the sake of readability, we use $M[S]$, $M[\pi]$, and $M[i]$ to denote the components S, π, and $\mathcal{B}_i$ of M, respectively. For $u \in S$, we write $M[\pi](u)$ to denote the interpretation associated to u via π and $M[\pi](u)(\varphi)$ to denote the truth value of a fluent formula φ with respect to the interpretation $M[\pi](u)$. In keeping with the tradition of action languages, we will often refer to $M[\pi](u)$ as the set of fluent literals defined by[4]

$$\{f \mid f \in \mathcal{F}, M[\pi](u)(f) = \top\} \cup \{\neg f \mid f \in \mathcal{F}, M[\pi](u)(f) = \bot\}.$$

[4]For simplicity of the presentation, we often omit the negative literals as well.

Given a consistent and complete set of literals X, i.e., $|\{f, \neg f\} \cap X| = 1$ for every $f \in \mathcal{F}$, we write $M[\pi](u) = X$ to indicate that the interpretation $M[\pi](u)$ is defined in such a way that $M[\pi](u) = X$.

Intuitively, a Kripke structure describes the possible worlds envisioned by the agents—and the presence of multiple worlds identifies uncertainty and the existence of different beliefs. The relation $(s_1, s_2) \in \mathcal{B}_i$ denotes that the belief of agent i about the real state of the world is insufficient for her to distinguish between the world described by s_1 and the one described by s_2. The world s in the state (M, s) identifies the world in $M[S]$ that corresponds to the actual world.

We will often view a Kripke structure $M = \langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ as a directed labeled graph, whose set of nodes is S and whose set of edges contains (s, i, t)[5] if and only if $(s, t) \in \mathcal{B}_i$. (s, i, t) is referred to as an edge coming out of (resp. into) the world s (resp. t).

Following van Ditmarsch et al. (2007), we will refer to a pointed Kripke structure (M, s) as a *state* and often use these two terms interchangeably.

The satisfaction relation between belief formulae and a state is defined next.

Definition 2. *Given a formula φ, a Kripke structure $M = \langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$, and a world $s \in S$:*

(i) $(M, s) \models \varphi$ *if p is a fluent and* $M[\pi](s) \models p$*;*

(ii) $(M, s) \models \mathbf{B}_i\varphi$ *if for each t such that* $(s, t) \in \mathcal{B}_i$, $(M, t) \models \varphi$*;*

(iii) $(M, s) \models \neg\varphi$ *if* $(M, s) \not\models \varphi$*;*

(iv) $(M, s) \models \varphi_1 \vee \varphi_2$ *if* $(M, s) \models \varphi_1$ *or* $(M, s) \models \varphi_2$*;*

(v) $(M, s) \models \varphi_1 \wedge \varphi_2$ *if* $(M, s) \models \varphi_1$ *and* $(M, s) \models \varphi_2$*.*

(vi) $(M, s) \models \mathbf{E}_\alpha\varphi$ *if* $(M, s) \models \mathbf{B}_i\varphi$ *for every* $i \in \alpha$*.*

(vii) $(M, s) \models \mathbf{C}_\alpha\varphi$ *if* $(M, s) \models \mathbf{E}^k_\alpha\varphi$ *for every* $k \geq 0$*, where*

- $\mathbf{E}^0_\alpha\varphi = \varphi$ *and*
- $\mathbf{E}^{k+1}_\alpha = \mathbf{E}_\alpha(\mathbf{E}^k_\alpha\varphi)$*.*

For a Kripke structure M and a formula φ, $M \models \varphi$ denotes the fact that $(M, s) \models \varphi$ for each $s \in M[S]$. The notation $\models \varphi$ indicates the fact that $M \models \varphi$ for every Kripke structure M.

Example 2 (State)**.** *Let us consider a simplified version of Example 1 in which the agents are concerned only with the status of the coin. The three agents A, B, C do not know whether the coin has 'heads' or 'tails' up and this is a common belief. Let us assume that the coin is heads up. The beliefs of the agents about the world and about the beliefs of other agents can be captured by the state of Figure 1.*

In the figure, a circle represents a world. The name of the world is written in the circle. Labeled edges between worlds denote the belief relations of the structure. A double circle identifies the real

[5] (s, i, t) denotes the edge from node s to node t, labeled by i.

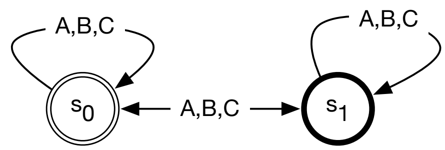

Figure 1: An example of a state

world. The interpretation of the world will be given whenever it is necessary. For example, we write $M[\pi](s_0) = \{head\}$ *to denote that* $head$ *is true in the world* s_0 *and anything else is false. Similarly,* $M[\pi](s_1) = \{\}$ *to denote that every fluent is false in the world* s_1*.*

We will occasionally be interested in Kripke structures that satisfy certain conditions. In particular, given a Kripke structure $M = \langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ we identify the following properties:

- **K**: for each agent i and formulae φ, ψ, we have that $M \models (\mathbf{B}_i\varphi \wedge \mathbf{B}_i(\varphi \Rightarrow \psi)) \Rightarrow \mathbf{B}_i\psi$.
- **T**: for each agent i and formula ψ, we have that $M \models \mathbf{B}_i\psi \Rightarrow \psi$.
- **4**: for each agent i and formula ψ, we have that $M \models \mathbf{B}_i\psi \Rightarrow \mathbf{B}_i\mathbf{B}_i\psi$.
- **5**: for each agent i and formula ψ, we have that $M \models \neg\mathbf{B}_i\psi \Rightarrow \mathbf{B}_i\neg\mathbf{B}_i\psi$.
- **D**: for each agent i we have that $M \models \neg\mathbf{B}_i \perp$.

A Kripke structure is said to be a **T**-Kripke (**4**-Kripke, **K**-Kripke, **5**-Krikpe, **D**-Kripke, respectively) structure if it satisfies property **T** (**4**, **K**, **5**, **D**, respectively). A Kripke structure is said to be an **S5** structure if it satisfies the properties **K**, **T**, **4**, and **5**. The **S5** properties have been often used to capture the notion of knowledge. Consistency of a set of formulae is defined next.

Definition 3. *A set of belief formulae* X *is said to be* p-satisfiable *(or* p-consistent*) for* $p \in \{\mathbf{S5}, \mathbf{K}, \mathbf{T}, \mathbf{4}, \mathbf{5}\}$ *if there exists a* p*-Kripke structure* M *and a world* $s \in M[S]$ *such that* $(M, s) \models \psi$ *for every* $\psi \in X$*. In this case,* (M, s) *is referred to as a* p-model *of X.*

Finally, let us introduce a notion of equivalence between states.

Definition 4. *A state* (M, s) *is* equivalent *to a state* (M', s') *if* $(M, s) \models \varphi$ *iff* $(M', s') \models \varphi$ *for every formula* $\varphi \in \mathcal{L}_{\mathcal{AG}}$*.*

2.2. Update Models

The formalism of *update models* has been used to describe transformations of (pointed) Kripke structures according to a predetermined transformation pattern. An update model is structured similarly to a pointed Kripke structure and it describes how to transform a pointed Kripke structure using an update operator defined in (Baltag and Moss, 2004; van Benthem et al., 2006).

Let us start with some preliminary definitions. An $\mathcal{L}_{\mathcal{AG}}$-substitution is a set $\{p_1 \rightarrow \varphi_1, \ldots, p_k \rightarrow \varphi_k\}$, 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.

Definition 5 (Update Model)**.** *Given a set $\mathcal{AG}$ of n agents, an* update model $\mathbf{\Sigma}$ *is a tuple* $\langle \Sigma, R_1, \ldots, R_n, pre, sub \rangle$ *where*

(i) Σ *is a set, whose elements are called* events*;*

(ii) each R_i *is a binary relation on* Σ*;*

(iii) $pre : \Sigma \rightarrow \mathcal{L}_{\mathcal{AG}}$ *is a function mapping each event* $e \in \Sigma$ *to a formula in* $\mathcal{L}_{\mathcal{AG}}$*; and*

(iv) $sub : \Sigma \rightarrow SUB_{\mathcal{L}_{\mathcal{AG}}}$ *is a function mapping each event* $e \in \Sigma$ *to a substitution in* $SUB_{\mathcal{L}_{\mathcal{AG}}}$*.*

An update instance ω *is a pair* $(\mathbf{\Sigma}, e)$ *where* $\mathbf{\Sigma}$ *is an update model* $\langle \Sigma, R_1, \ldots, R_n, pre, sub \rangle$ *and* e*, referred to as a* designated event*, is a member of* Σ*.*

Intuitively, an update model represents different views of an action occurrence which are associated with the observability of agents. Each view is represented by an event in Σ. The designated event is the one that agents who are aware of the action occurrence will observe. The relation R_i describes agent i's uncertainty on action execution—i.e., if $(\sigma, \tau) \in R_i$ and event σ is performed, then agent i may believe that event τ is executed instead. pre defines the action precondition and sub specifies the changes of fluent values after the execution of an action.

Definition 6 (Updates by an Update Model)**.** *Let* M *be a Kripke structure and* $\mathbf{\Sigma} = \langle \Sigma, R_1, \ldots, R_n, pre, sub \rangle$ *be an update model. The* update *induced by* $\mathbf{\Sigma}$ *defines a Kripke structure* $M' = M \otimes \mathbf{\Sigma}$*, where:*

(i) $M'[S] = \{(s, \tau) \mid s \in M[S], \tau \in \Sigma, (M, s) \models pre(\tau)\}$*;*

(ii) $((s, \tau), (s', \tau')) \in M'[i]$ *iff* $(s, \tau), (s', \tau') \in M'[S]$*,* $(s, s') \in M[i]$ *and* $(\tau, \tau') \in R_i$*;*

(iii) *For all* $(s, \tau) \in M'[S]$ *and* $f \in \mathcal{F}$*,* $M'[\pi]((s, \tau)) \models f$ *iff* $f \rightarrow \varphi \in sub(\tau)$ *and* $(M, s) \models \varphi$*.*

The structure M' is obtained from the component-wise cross-product of the old structure M and the update model $\mathbf{\Sigma}$, by **(i)** removing pairs (s, τ) such that (M, s) does not satisfy the action precondition (checking for satisfaction of action's precondition), and **(ii)** removing links of the form $((s, \tau), (s', \tau'))$ from the cross product of $M[i]$ and R_i if $(s, s') \notin M[i]$ or $(\tau, \tau') \notin R_i$ (ensuring that each agent's accessibility relation is updated according to the update model).

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

$$(M, s) \otimes (\mathbf{\Sigma}, \Gamma) = \{(M \otimes \mathbf{\Sigma}, (s, \tau)) \mid \tau \in \Gamma, (M, s) \models pre(\tau)\}$$

Remark 1. *In the following, we will often represent an update instance by a graph with rectangles, double rectangles, and labeled links between rectangles representing events, designated events, and the relation of agents, respectively, as in the graphical representation of a Kripke structure.*

2.3. Finitary **S5**-Theories

A finitary **S5**-theory, introduced in (Son et al., 2014), is a collection of formulae which has finitely many **S5**-models, up to a notion of equivalence (Definition 4). To define finitary **S5**-theories, we need the following notion. Given a set of propositions $\mathcal{F}$, a *complete clause* over $\mathcal{F}$ is a disjunction of the form $\bigvee_{p \in \mathcal{F}} p^*$ where p^* is either p or $\neg p$. We will consider formulae of the following forms:

$$\varphi \tag{1}$$
$$\mathbf{C}(\mathbf{B}_i \varphi) \tag{2}$$
$$\mathbf{C}(\mathbf{B}_i \varphi \vee \mathbf{B}_i \neg \varphi) \tag{3}$$
$$\mathbf{C}(\neg \mathbf{B}_i \varphi \wedge \neg \mathbf{B}_i \neg \varphi) \tag{4}$$

where φ is a fluent formula.

Definition 7. *A theory T is said to be* primitive finitary **S5** *if*

- *Each formula in T is of the form (1)-(4); and*
- *For each complete clause φ over $\mathcal{F}$ and each agent i, T contains either* **(i)** $\mathbf{C}(\mathbf{B}_i \varphi)$ *or* **(ii)** $\mathbf{C}(\mathbf{B}_i \varphi \vee \mathbf{B}_i \neg \varphi)$ *or* **(iii)** $\mathbf{C}(\neg \mathbf{B}_i \varphi \wedge \neg \mathbf{B}_i \neg \varphi)$.

A theory T is a finitary **S5**-theory *if $T \models H$ and H is a primitive finitary* **S5***-theory. T is* pure *if T contains only formulae of the form* (1)-(4).

We say that a state (M, s) is *canonical* if for every pairs of worlds $u, v \in M[S]$ and $u \neq v$, $M[\pi](u) \not\equiv M[\pi](v)$ holds. We have that

Theorem 1 (From (Son et al., 2014))**.** *Every finitary* **S5***-theory T has finitely many finite canonical models, up to equivalence. If T is pure then these models are minimal and their structures are identical up to the name of the points.*

2.4. Why an Action Language?

As mentioned earlier, the Dagstuhl seminars (Agotnes et al., 2014; Baral et al., 2017) identified one of the main research topics in EPM: *the development of an adequate specification language for EPM*. This problem arises from the fact that EPM has been defined and investigated using a DEL based approach, in which actions are represented by update models (Section 2.2). This representation has been useful for the understanding of EPM and the study of its complexity, but comes with a significant drawback—practical and conceptual knowledge representation challenges—discussed by Herzig[6] at the Dagstuhl seminar (Baral et al., 2017). Let us consider a slight modification of Example 1, where the box is open, A has looked at the coin, while both B and C are distracted, and A can announce whether the coin lies heads up or tails up. However, only agents who are attentive (to A) could listen to what A says. Assume that A announces that the coin lies heads up. Intuitively,

[6] http://materials.dagstuhl.de/files/17/17231/17231.AndreasHerzig.Slides.pdf

this action occurrence can have different effects on the beliefs of the other agents—depending on the context and the specific features of each of them, e.g., whether the agent is attentive to A's announcement. As a result, we need a variety of update models to represent this primitive action. Herzig refers to this problem as the *action type vs. action token* problem.

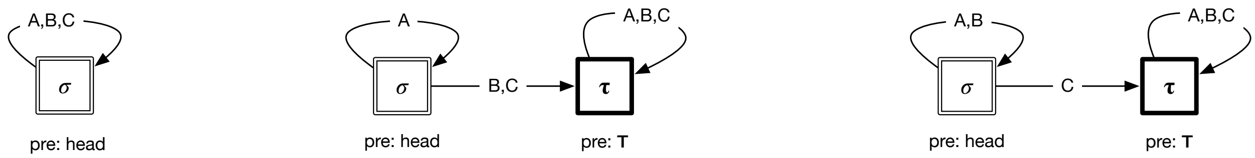

Figure 2: Update models for announcing *"the coin lies heads up"* by A in different situations

Fig. 2 shows three update models; they describe the occurrence of the announcement by A, stating that the coin lies heads up, assuming that the coin indeed lies heads up in the real state of the world. On the left is the update model when both B and C are attentive. The model in the middle depicts the situation when both B nor C are not attentive. The update model on the right captures the case of B being attentive and C being not attentive. In the figures, σ and τ are events and σ is a *designated event,* $head$ is a propositional variable denoting that the coin lies heads up.

Observe that these models are used only when the coin indeed lies heads up. The update models corresponding to the situation where $head$ is false (i.e., when A makes a false announcement) in the real state of the world are different from those in the figure and are omitted. It is easy to see that the number of update models needed to represent such simple announcement of *"the coin lies heads up"* by A is exponential in the number of agents. This is certainly an undesirable consequence of using update models and epistemic actions for representing and reasoning about effects of actions in multi-agent domains. Therefore, any specification language for representing and reasoning about the effects of actions in multi-agent domains should consider that the announcement of the coin lies heads up by A is simply a *primitive action*. In our view, the update models should be derived from the concrete state, which is a combination of the real state of the world and the state of beliefs of the agents, and not specified directly. A more detailed discussion on this issue can be found in Section 5, the related work section.

3. The language m$\mathcal{A}^*$: Syntax

In this paper, we consider multi-agent domains in which the agents are truthful and no false information may be announced or observed. Furthermore, the underlying assumptions guiding the semantics of our language are the rationality principle and the idea that beliefs of an agent are inertial. In other words, agents believe something because they have a reason to, and the beliefs of an agent remain the same unless something causes them to change.

In this section and in the next section, we introduce the language m$\mathcal{A}^*$ for describing actions and their effects in multi-agent environment. The language is built over a signature $\langle \mathcal{AG}, \mathcal{F}, \mathcal{A} \rangle$, where $\mathcal{AG}$ is a finite set of agent identifiers, $\mathcal{F}$ is a set of fluents, and $\mathcal{A}$ is a set of actions. Each action in $\mathcal{A}$ is an action the agents in the domain are capable of performing.

Similar to any action language developed for single-agent environments, m$\mathcal{A}^*$ consists of three components which will be used in describing the actions and their effects, the initial state, and the query language (see, e.g., (Gelfond and Lifschitz, 1998)). We will next present each of these components. Before we do so, let us denote the multi-agent domain in Example 1 by D_1. For this domain, we have that $\mathcal{AG} = \{A, B, C\}$. The set of fluents $\mathcal{F}$ for this domain consists of:

- $tail$: the coin lies tails up ($head$ is often used in place of $\neg tail$);
- $has\_key(x)$: agent x has the key of the box;
- $opened$: the box is open; and
- $looking(x)$: agent x is looking at the box.

The set of actions for D_1 consists of:

- $open$: an agent opens the box;
- $peek$: an agent peeks into the box;
- $signal(y)$: an agent signals agent y (to look at the box);
- $distract(y)$: an agent distracts agent y (so that y does not look at the box); and
- $shout\_tail$: an agent announces that the coin lies tails up.

where $x, y \in \{A, B, C\}$. We start with the description of actions and their effects.

3.1. Actions and effects

We envision three types of actions that an agent can perform: *world-altering actions* (also known as *ontic actions*), *sensing actions*, and *announcement actions*. Intuitively,

- A world-altering action is used to explicitly modify certain properties of the world—e.g., the agent A opens the box in Example 1, or the agent A distracts the agent C so that C does not look at the box (also in Example 1);
- A sensing action is used by an agent to refine its beliefs about the world, by making direct observations—e.g., an agent peeks into the box; the effect of the sensing action is to reduce the amount of uncertainty of the agent;
- An announcement action is used by an agent to affect the beliefs of the agents receiving the communication—we operate under the assumption that agents receiving an announcement always believe what is being announced.

For the sake of simplicity, we assume that each action $a \in \mathcal{A}$ falls in exactly one of the three categories.[7] In a multi-agent system, we need to identify an action occurrence with the agents who execute it. Given an action $a \in \mathcal{A}$ and a set of agents $\alpha \subseteq \mathcal{AG}$, we write $\mathsf{a}\langle\alpha\rangle$ to denote the joint-execution of a by the agents in α and call it an *action instance*. We will use $\mathcal{AI}$ to denote the set of possible action instances $\mathcal{A} \times 2^{\mathcal{AG}}$. Elements of $\mathcal{AI}$ will be written in sans-serif font. For simplicity of the presentation, we often use "an action" or "an action instance" interchangeably when it is clear from the context which term is appropriate. Furthermore, when α is a singleton set, $\{x\}$, we simplify $\langle\{x\}\rangle$ to $\langle x\rangle$.

In general, an action can be executed only under certain conditions, called its *executability conditions*. For example, the statement "to open the box, an agent must have its key" in Example 1 describes the executability condition of the action of opening a box. The first type of statements in m$\mathcal{A}^*$ is used to describe the executability conditions of action occurrences and is of the following form:

$$\textbf{executable } \mathsf{a} \textbf{ if } \psi \tag{5}$$

where $\mathsf{a} \in \mathcal{AI}$ and ψ is a belief formula. A statement of type (5) will be referred to as the *executability condition of the action occurrence* a. ψ is referred as the *precondition of* a. For simplicity of the presentation, we will assume that each action occurrence a is associated with exactly one executability condition. When $\psi = \top$, the statement will be omitted.

For an occurrence of a world-altering action a, such as the action of opening the box by some agent, we have statements of the following type that express the change that may be caused by such action:

$$\mathsf{a} \textbf{ causes } \ell \textbf{ if } \psi \tag{6}$$

where ℓ is a fluent literal and ψ is a belief formula. Intuitively, if the real state of the world and the beliefs match the condition described by ψ, then the real state of the world is affected by the change that makes the literal ℓ true after the execution of a. When $\psi = \top$, the part " **if** ψ" will be omitted from (6). We also use

$$\mathsf{a} \textbf{ causes } \phi \textbf{ if } \psi$$

where ϕ is a set of fluent literals as a shorthand for the set $\{\mathsf{a} \textbf{ causes } \ell \textbf{ if } \psi \mid \ell \in \phi\}$.

Sensing actions, such as the action of looking into the box, allow agents to learn about the value of a fluent in the real state of the world (e.g., learn whether the coin lies heads or tails up). We use statements of the following kind to represent effects of sensing action occurrences:

$$\mathsf{a} \textbf{ determines } \varphi \tag{7}$$

where φ is a fluent formula and $\mathsf{a} \in \mathcal{AI}$ is a sensing action. Statements of type (7) encode the occurrence of a sensing action a which enables the agent(s) to learn the value of the fluent formula φ. φ is referred to as a *sensed fluent formula* of a.

For actions such as the action of an agent telling another agent that the coin lies heads up, we have statements of the following kind:

[7] It is easy to relax this condition, but it would make the presentation more tedious.

$$\mathsf{a}\ \textbf{announces}\ \varphi \tag{8}$$

where φ is a fluent formula and $\mathsf{a} \in \mathcal{AI}$. a is called *an announcement action*.

We will next illustrate the use of statements of the form (5)-(8) to represent the actions of the domain D_1.

Example 3. The actions of domain D_1 can be specified by the following statements:

$$\begin{array}{l}
\textbf{executable}\ \ open\langle x\rangle\ \textbf{if}\ \ has_key(x)\\
\textbf{executable}\ \ peek\langle x\rangle\ \textbf{if}\ \ opened, looking(x)\\
\textbf{executable}\ \ shout_tail\langle x\rangle\ \textbf{if}\ \ \mathrm{B}_x(tail), tail\\
\textbf{executable}\ \ signal(y)\langle x\rangle\ \textbf{if}\ \ looking(x), \neg looking(y)\\
\textbf{executable}\ \ distract(y)\langle x\rangle\ \textbf{if}\ \ looking(x), looking(y)
\end{array}$$

$$\begin{array}{l}
open\langle x\rangle\ \textbf{causes}\ \ opened\\
signal(y)\langle x\rangle\ \textbf{causes}\ \ looking(y)\\
distract(y)\langle x\rangle\ \textbf{causes}\ \ \neg looking(y)\\
peek\langle x\rangle\ \textbf{determines}\ tail\\
shout_tail\langle x\rangle\ \textbf{announces}\ tail
\end{array}$$

where x and y are different agents in $\{A, B, C\}$. The first five statements encode the executability conditions of the five actions in the domain. The next three statements describe the effects of the occurrence of three world-altering actions. $peek\langle x\rangle$ is an example of an instance of a sensing action. Finally, $shout\_tail\langle x\rangle$ is an example of an instance of an announcement action.

3.2. Observability: observers, partial observers, and others

One of the key differences between single-agent and multi-agent domains lies in how the execution of an action changes the beliefs of agents. This is because, in multi-agent domains, an agent might be oblivious about the occurrence of an action or unable to observe the effects of an action. For example, watching another agent open the box would allow the agent to know that the box is open after the execution of the action; however, the agent would still believe that the box is closed if she is not aware of the action occurrence. On the other hand, watching another agent peek into the box does not help the observer in learning whether the coin lies heads or tails up; the only thing she would learn is that the agent who is peeking into the box has knowledge of the status of the coin.

m$\mathcal{A}^*$ needs to have a component for representing the fact that not all the agents may be completely aware of the occurrence of actions being executed. Depending on the action and the current situation, we can categorize agents in three classes:

- *Full observers*,
- *Partial observers*, and
- *Oblivious* (or *others*).

This categorization is dynamic—changes in the state of the world and/or the beliefs of agents may change the observability of actions. In this paper, we will consider the possible observabilities of agents for different action types as detailed in Table 1.

action type	**full observers**	**partial observers**	**oblivious/others**
world-altering actions	$\surd$		$\surd$
sensing actions	$\surd$	$\surd$	$\surd$
announcement actions	$\surd$	$\surd$	$\surd$

Table 1: Action types and agent observability

The first row indicates that, for a world-altering action, an agent can either be a *full observer*, i.e., completely aware of the occurrence of that action, or oblivious of the occurrence of the action. The assumption here is that agents are fully knowledgeable of the outcomes of a world-changing action (which makes partial observability a moot point). In the second case, the observability of the agent is categorized as *other*. Note that we assume that the observer agents know about each others' status and they are also aware of the fact that the other agents are oblivious. The oblivious agents have no clue of anything. Notice also that in multi-agent domain, an agent, who executes an action, might not be a full observer of the action occurrence.

For a sensing action, an agent can either be a *full observer*, i.e., it is aware of the occurrence of that action and of its results, it can be a *partial observer*, gaining knowledge that the full observers have performed a sensing action but without knowledge of the result of the observation, or it can be oblivious of the occurrence of the action (i.e., *other*). Once again, we assume that the observer agents know about each others' status and they also know about the agents partially observing the action and about the agents that are oblivious. The partially observing agents know about each others' status, and they also know about the observing agents and the agents that are oblivious. The oblivious agents have no clue of anything. The behavior is analogous for the case of announcement actions.

We observe that this classification is limited to individual agent's observability of action occurrences. As agents are knowledgeable about the domains, they can reason about others' observability when an action occurs and manipulate others' observability, thereby others' knowledge about the world and beliefs. As such, it is reasonable for an agent to use m$\mathcal{A}^*$ for planning purpose, e.g., in Example 1, A distracts C and signals B before opens the box. The classification, however, does not consider situations in which an agent might have uncertainty about others' observability. We discuss this limitation in Section 5.1.2. A possible way to address this issue is to remove the assumption that agents are oblivious of actions' occurrences by default. This topic of research is interesting in its own right and deserves a throughout investigation. We leave this as a future work.

Agents' observability is meant to be dynamic and the dynamic behavior is described by agent observability statements of the following forms:[8]

$$z \textbf{ observes } \mathsf{a} \textbf{ if } \varphi \tag{9}$$

[8]As discussed earlier, the " **if** $\top$" are omitted from the statements.

$$z \textbf{ aware\_of } \mathsf{a} \textbf{ if } \psi \tag{10}$$

where $z \in \mathcal{AG}$, $\mathsf{a} \in \mathcal{AI}$, and φ and ψ are fluent formulae. (9) indicates that agent z is a full observer of a if φ holds. (10) states that agent z is a partial observer of a if ψ holds. z, a, and φ (resp. ψ) are referred to as the observed agent, the action instance, and the condition of (9) (resp. (10)). The next example illustrates the use of the above statements in specifying the agents observability of the domain D_1.

Example 4 (Observability in D_1)**.** *The actions of D_1 are described in Example 3. The observability of agents in D_1 can be described by the set O_1 of statements*

$$
\begin{array}{ll}
x \textbf{ observes } open\langle x\rangle & x \textbf{ observes } peek\langle x\rangle \\
y \textbf{ observes } open\langle x\rangle \textbf{ if } looking(y) & y \textbf{ aware\_of } peek\langle x\rangle \textbf{ if } looking(y) \\
z \textbf{ observes } shout\_tail\langle x\rangle & \forall z \in \{A, B, C\} \\
x \textbf{ observes } distract(y)\langle x\rangle & x \textbf{ observes } signal(y)\langle x\rangle \\
y \textbf{ observes } distract(y)\langle x\rangle & y \textbf{ observes } signal(y)\langle x\rangle \\
z \textbf{ observes } distract(y)\langle x\rangle \textbf{ if } looking(z) & \\
z \textbf{ observes } signal(y)\langle x\rangle \textbf{ if } looking(z) &
\end{array}
$$

where x and y denote different agents in $\{A, B, C\}$. The above statements say that agent x is a fully observant agent when $open\langle x\rangle$, $peek\langle x\rangle$, $distract(y)\langle x\rangle$, $signal(y)\langle x\rangle$, or $shout\_tail\langle x\rangle$ is executed; y is a fully observant agent if it is looking (at the box) when $open\langle x\rangle$ is executed. y is a partially observant agent if it is looking when $peek\langle x\rangle$ is executed. An agent different from x and y is oblivious if $open\langle x\rangle$ or $peek\langle x\rangle$ is executed.

It is obvious that an agent cannot be both partially observable and fully observable of the execution of an action at the same time. For this reason, we will say that a statement of the form (9) is in conflict with a statement of the form (10) if for the same occurrence $\mathsf{a} \in \mathcal{AI}$ and $z \in \mathcal{AG}$, $\varphi \wedge \psi$ is consistent.

Definition 8. *An m$\mathcal{A}^*$* domain *is a collection of statements of the forms (5)-(10).*

Similarly to action domains in the language $\mathcal{A}$ introduced by Gelfond and Lifschitz (1993), an m$\mathcal{A}^*$ domain could contain two statements specifying contradictory effects of an action occurrence such as

$$\mathsf{a} \textbf{ causes } f \textbf{ if } \varphi \qquad \text{and} \qquad \mathsf{a} \textbf{ causes } \neg f \textbf{ if } \psi$$

where $\varphi \wedge \psi$ is a consistent formula, i.e., there exists some pointed Kripke structure (M, s) such that $(M, s) \models \varphi \wedge \psi$. Such a domain is not sensible and will be characterized as *inconsistent*.

Definition 9. *An m$\mathcal{A}^*$* domain *D is* consistent *if for every pointed Kripke structure (M, s) and*

- *for every pair of two statements*

$$\mathsf{a} \textbf{ causes } f \textbf{ if } \varphi \qquad \text{and} \qquad \mathsf{a} \textbf{ causes } \neg f \textbf{ if } \psi$$

in D, $(M, s) \not\models \varphi \wedge \psi$; and

- *for every pair of two statements*

$$z \textbf{ observes } \mathsf{a} \textbf{ if } \varphi \qquad \text{and} \qquad z \textbf{ aware\_of } \mathsf{a} \textbf{ if } \psi$$

in D, $(M, s) \not\models \varphi \wedge \psi$.

From now on, whenever we say an m$\mathcal{A}^*$ domain D, we will assume that D is consistent.

3.3. Initial State

A domain specification encodes the actions, and their effects, and the observability of agents in each situation. The initial state, that encodes both the initial state of the world and the initial beliefs of the agents, is specified in m$\mathcal{A}^*$ using *initial statements* of the following form:

$$\textbf{initially } \varphi \tag{11}$$

where φ is a belief formula. Intuitively, this statement says that φ is true in the initial state. We will later discuss restrictions on the formula φ to ensure the computability of the Kripke structures describing the initial state.

Example 5 (Representing Initial State of D_1)**.** Let us reconsider Example 1. The initial state of D_1 can be expressed by the following statements:

$$\begin{array}{ll} \textbf{initially } \mathbf{C}(has\_key(A)) & \\ \textbf{initially } \mathbf{C}(\neg has\_key(B)) & \\ \textbf{initially } \mathbf{C}(\neg has\_key(C)) & \\ \textbf{initially } \mathbf{C}(\neg opened) & \\ \textbf{initially } \mathbf{C}(\neg \mathbf{B}_x head \wedge \neg \mathbf{B}_x \neg head) & \text{for } x \in \{A, B, C\} \\ \textbf{initially } \mathbf{C}(looking(x)) & \text{for } x \in \{A, B, C\} \end{array}$$

These statements indicate that everyone knows that A has the key and B and C do not have the key, the box is closed, no one knows whether the coin lies heads or tails up, and everyone is looking at the box.

The notion of an action theory in m$\mathcal{A}^*$ is defined next.

Definition 10 (Action Theory)**.** *An m$\mathcal{A}^*$-*action theory *is a pair* (I, D) *where* D *is an m$\mathcal{A}^*$ domain and* I *is a set of statements of the form* (11).

In Section 4, we will define the notion of entailment between action theories and queries, in a manner similar to the notion of entailment defined for action languages in single-agent domains (e.g., (Gelfond and Lifschitz, 1998)). This requires the following definition.

Definition 11 (Initial State/Belief-State)**.** *Let* (I, D) *be an action theory. An* initial state *of* (I, D) *is a state* (M, s) *such that for every statement*

$$\textbf{initially } \varphi$$

in I, $(M, s) \models \varphi$.

(M, s) *is an initial* **S5***-state if it is an initial state and* M *is a* **S5** *Kripke structure.*

The initial belief-state *(or* initial b-state*) of* (I, D) *is the collection of all initial states of* (I, D).

The initial **S5**-belief state *of* (I, D) *is the collection of all initial* **S5***-states of* (I, D).

By definition, it is easy to see that, theoretically, there could be infinitely many initial states for an arbitrary m$\mathcal{A}^*$ theory. For example, given a state (M, s) and a set of formulae Σ such that $(M, s) \models \Sigma$, a new state (M', s) that also satisfies Σ can be constructed from M by simply stating $M'[S] = M[S] \cup \{s'\}$, where $s' \notin M[S]$, and keeping everything else unchanged. As such, it is important to identify sensible classes of action theories whose initial belief states are finite, up to a notion of equivalence (Definition 4). Fortunately, the result on finitary **S5** theories[9] (Definition 7) allows us to identify a large class of action theories satisfying this property. We call them definite action theories and define them as follows.

Definition 12 (Definite Action Theory)**.** *An action theory (I, D) is said to be* definite *if the theory $\{\varphi \mid$ **initially** φ belongs to $I\}$ is a finitary-***S5** *theory.*

Observe that Theorem 1 indicates that for definite action theories, the initial belief state is finite. An algorithm for computing the initial belief state is given in (Son et al., 2014). This, together with the definition of the transition function of m$\mathcal{A}^*$ domains in the next section, allows the implementation of search-based progression epistemic planning systems. A preliminary development can be found in (Le et al., 2018).

It is worth pointing out that Definition 11 does not impose any condition on the initial state of action theories. It merely characterizes a subgroup of action theories (**S5** action theories), for which some interesting properties can be proved. We would also like to reiterate that most of our discussion in this paper focuses on beliefs rather than knowledge. Additional steps need to be taken for answering questions related to knowledge of agents after the execution of an action sequence. For example, the ability of maintaining the KD45 properties of the resulting pointed Kripke structures after the execution of an action will be important. Preliminary investigation in this direction was presented by Son et al. (2015).

4. Update Model Based Semantics for m$\mathcal{A}^*$ Domains

An m$\mathcal{A}^*$ domain D specifies a transition system, whose nodes are states. This transition system will be described by a transition function Φ_D, which maps pairs of action occurrences and states to states. For simplicity of the presentation, we assume that only one action occurrence happens at each point in time—it is relatively simple to extend it to cases where concurrent actions are present, and this is left as future work. As we have mentioned in Section 2, we will use pointed Kripke structures to represent states in m$\mathcal{A}^*$ action theories. A pointed Kripke structure encodes three components:

- The actual world;
- The state of beliefs of each agent about the real state of the world; and
- The state of beliefs of each agent about the beliefs of other agents.

[9] To keep the paper at a reasonable length, we do not discuss the details of finitary **S5** theories. Interested readers are referred to (Son et al., 2014) for details and proof of Theorem 1.

These components are affected by the execution of actions. Observe that the notion of a state in m$\mathcal{A}^*$ action theories is more complex than the notion of state used in single-agent domains (i.e., a complete set of fluent literals).

Let $\mathcal{S}$ be the set of all possible pointed Kripke structures over $\mathcal{L}(\mathcal{F}, \mathcal{AG})$, the transition function Φ_D maps pairs of action instances and states into sets of states, i.e.,

$$\Phi_D : \mathcal{AI} \times \mathcal{S} \longrightarrow 2^{\mathcal{S}} \tag{12}$$

will be defined for each action type separately and in two steps. First, we define an update model representing the occurrence of $\mathsf{a} \in \mathcal{AI}$ in a state (M, s). Second, we use the update model and update template defined in step one to define Φ_D. We start by defining the notion of a frame of reference in order to define the function Φ_D.

4.1. Actions Visibility and Frames of Reference

Given a state (M, s) and an action occurrence a, let us define

$$\begin{aligned} \boldsymbol{F}_D(\mathsf{a}, M, s) &= \{x \in \mathcal{AG} \mid [x \textbf{ observes } \mathsf{a} \textbf{ if } \varphi] \in D \text{ such that } (M, s) \models \varphi\} \\ \boldsymbol{P}_D(\mathsf{a}, M, s) &= \{x \in \mathcal{AG} \mid [x \textbf{ aware\_of } \mathsf{a} \textbf{ if } \varphi] \in D \text{ such that } (M, s) \models \varphi\} \\ \boldsymbol{O}_D(\mathsf{a}, M, s) &= \mathcal{AG} \setminus (\boldsymbol{F}_D(\mathsf{a}, M, s) \cup \boldsymbol{P}_D(\mathsf{a}, M, s)) \end{aligned}$$

We will refer to the tuple $(\boldsymbol{F}_D(\mathsf{a}, M, s), \boldsymbol{P}_D(\mathsf{a}, M, s), \boldsymbol{O}_D(\mathsf{a}, M, s))$ as the *frame of reference* for the execution of a in (M, s). Intuitively, $\boldsymbol{F}_D(\mathsf{a}, M, s)$ (resp. $\boldsymbol{P}_D(\mathsf{a}, M, s)$ and $\boldsymbol{O}_D(\mathsf{a}, M, s)$) are the agents that are fully observant (resp. partially observant and oblivious/other) of the execution of a in the state (M, s). As we assume that for each pair of an action occurrence a and a state (M, s), the sets $(\boldsymbol{F}_D(\mathsf{a}, M, s)$, $\boldsymbol{F}_D(\mathsf{a}, M, s)$, and $\boldsymbol{P}_D(\mathsf{a}, M, s)$ are pairwise disjoint, the domain specification D and the state (M, s) determine a unique frame of reference for each action occurrence.

The introduction of frames of reference allows us to elegantly model several types of actions that are aimed at modifying the frame of reference (referred to as *reference setting actions*). Some possibilities are illustrated in the following examples.

Example 6 (Reference Setting Actions)**.** *Example 4 shows two reference setting actions signal*(y) *and distract*(y) *with instances of the form signal*$(y)\langle x\rangle$ *and distract*$(y)\langle x\rangle$ *because they change the truth value of* $looking(y)$ *to true and false, respectively, which decides whether or not the agent* y *is aware (or partially aware) of the occurrence of an action instance* $open\langle x\rangle$ *(or* $peek\langle x\rangle$*).*

The action instance signal$(y)\langle x\rangle$ *allows agent* x *to promote agent* y *into a higher level of observation for the effect of peek*$\langle x\rangle$*. On the other hand, the action instance distract*$(y)\langle x\rangle$ *allows agent* x *to demote agent* y *into a lower level of observation. The net effect of executing these actions is a change of frame.*

Let us consider signal$(y)\langle x\rangle$ *and a state* (M, s)*. Furthermore, let us assume that* (M', s') *is a state resulting from the execution of signal*$(y)\langle x\rangle$ *in* (M, s)*. The frames of reference for the execution of the action instance* $\mathsf{a} = peek\langle x\rangle$ *in these two states are related to each other by the following equations:*

$$\begin{aligned} F_{D_1}(\mathsf{a}, M', s') &= F_{D_1}(\mathsf{a}, M, s) \\ P_{D_1}(\mathsf{a}, M', s') &= P_{D_1}(\mathsf{a}, M, s) \cup \{y\} \\ O_{D_1}(\mathsf{a}, M', s') &= O_{D_1}(\mathsf{a}, M, s) \setminus \{y\} \end{aligned}$$

Intuitively, after the execution of $signal(y)\langle x\rangle$, $looking(y)$ becomes true because of the statement

$$signal(y)\langle x\rangle \textbf{ causes } looking(y)$$

in D_1. By definition, the statement

$$y \textbf{ aware\_of } peek\langle x\rangle \textbf{ if } looking(y)$$

indicates that y is partially observant.

Similar argument shows that $distract(y)\langle x\rangle$ demotes y to the lowest level of visibility, i.e., it will cause agent y to become oblivious of the successive peek$\langle x\rangle$ action. Let us assume that the execution of a *in (M, s) resulted in (M', s'). Then, the frames of reference for the execution of the action instance* $\mathsf{a} = peek\langle x\rangle$ *in these two states are related to each other by the following equations:*

$$\begin{array}{rcl} F_{D_1}(\mathsf{a}, M', s') & = & F_{D_1}(\mathsf{a}, M, s) \setminus \{y\} \\ P_{D_1}(\mathsf{a}, M', s') & = & P_{D_1}(\mathsf{a}, M, s) \setminus \{y\} \\ O_{D_1}(\mathsf{a}, M', s') & = & O_{D_1}(\mathsf{a}, M, s) \cup \{y\} \end{array}$$

4.2. Update Model for Action Occurrences

Definition 13 (Update Model/Template for World-Altering Actions)**.** *Given a world-altering action instance* a *with the precondition ψ and a frame of reference $\rho = (F, \emptyset, O)$, the update model for* a *and ρ, denoted by $\omega(\mathsf{a}, \rho)$, is defined by $\langle \Sigma, R_1, \ldots, R_n, pre, sub\rangle$ where*

- $\Sigma = \{\sigma, \epsilon\}$;
- $R_i = \{(\sigma, \sigma), (\epsilon, \epsilon)\}$ *for* $i \in F$ *and* $R_i = \{(\sigma, \epsilon), (\epsilon, \epsilon)\}$ *for* $i \in O$;
- $pre(\sigma) = \psi$ *and* $pre(\epsilon) = \top$; *and*
- $sub(\epsilon) = \emptyset$ *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

$$\Psi^-(p, \mathsf{a}) = \bigvee\{\varphi \mid [\mathsf{a} \textbf{ causes } \neg p \textbf{ if } \varphi] \in D\}.$$

The update template for the occurrence of a *and the frame of reference ρ is $(\omega(\mathsf{a}, \rho), \{\sigma\})$.*

Observe that the update model of the world-altering action occurrence has two events. Each event is associated to a group of agents in the frame of reference. The links in the update model for each group of agents reflect the state of beliefs each group would have after the execution of the action. For example, fully observant agents (in F) will have no uncertainty. The next example illustrates this definition.

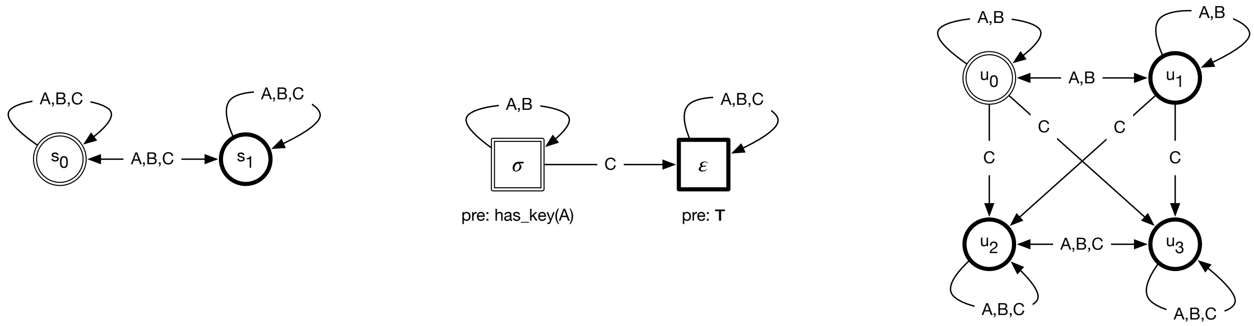

Figure 3: Update template $(\omega(\textit{open}\langle A\rangle, (\{A, B\}, \emptyset, \{C\})), \{\sigma\})$ and its application

Example 7. *Going back to our original example, consider the occurrence of the action instance* $\textit{open}\langle A\rangle$ *assuming that everyone is aware that* C *is not looking at the box while* B *and* A *are. Figure 3 (left) depicts the state* (M, s_0) *where* $M[\pi](s_0) = \{looking(A), looking(B), has\_key(A)\}$ *and* $M[\pi](s_1) = \{looking(A), looking(B), has\_key(A), head\}$*. The frame of reference for* $\textit{open}\langle A\rangle$ *in this situation is* $(\{A, B\}, \emptyset, \{C\})$*. The corresponding update template for* $\textit{open}\langle A\rangle$ *and the frame of reference* $(\{A, B\}, \emptyset, \{C\})$ *is given in Figure 3 (middle).*

The figure on the right in Figure 3 shows (M', u_0)*, the result of the application of the update template to the state* (M, s_0) *where*
$u_0 = (s_0, \sigma)$[10] *with* $M'[\pi](u_0) = \{looking(A), looking(B), has\_key(A), \textit{opened}\}$
$u_1 = (s_1, \sigma)$ *with* $M'[\pi](u_1) = \{looking(A), looking(B), head, has\_key(A), \textit{opened}\}$
$u_2 = (s_0, \epsilon)$ *with* $M'[\pi](u_2) = \{looking(A), looking(B), has\_key(A)\}$
$u_3 = (s_1, \epsilon)$ *with* $M'[\pi](u_3) = \{looking(A), looking(B), head, has\_key(A)\}$.

In the next definition, we provide the update template for a sensing or announcement action occurrence given a frame of reference. For simplicity of presentation, we will assume that the set of sensed formulae of the action is a singleton.

Definition 14 (Update Model/Template for Sensing/Announcement Actions)**.** *Let* a *be a sensing action instance that senses* φ *or an announcement action instance that announces* φ *with the precondition* ψ *and* $\rho = (F, P, O)$ *be a frame of reference. The update model for* a *and* ρ*,* $\omega(\mathsf{a}, \rho)$*, is defined by* $\langle \Sigma, R_1, \ldots, R_n, pre, sub\rangle$ *where:*

- $\Sigma = \{\sigma, \tau, \epsilon\}$*;*
- R_i *is given by*

$$R_i = \begin{cases} \{(\sigma,\sigma), (\tau,\tau), (\epsilon,\epsilon)\} & \textit{if } i \in F \\ \{(\sigma,\sigma), (\tau,\tau), (\epsilon,\epsilon), (\sigma,\tau), (\tau,\sigma)\} & \textit{if } i \in P \\ \{(\sigma,\epsilon), (\tau,\epsilon), (\epsilon,\epsilon)\} & \textit{if } i \in O \end{cases}$$

[10]This is to say that u_0 denotes the world (s_0, σ) as in Definition 6.

- *The preconditions* pre *are defined by*

$$pre(x) = \begin{cases} \psi \wedge \varphi & \textit{if } x = \sigma \\ \psi \wedge \neg\varphi & \textit{if } x = \tau \\ \top & \textit{if } x = \epsilon \end{cases}$$

- $sub(x) = \emptyset$ *for each* $x \in \Sigma$.

The update template for the sensing action occurrence a *and the frame of reference* ρ *is* $(\omega(\mathsf{a}, \rho), \{\sigma, \tau\})$ *while the update template for the announcement action occurrence* a *and the frame of reference* ρ *is* $(\omega(\mathsf{a}, \rho), \{\sigma\})$

Observe that an update model of a sensing or announcement action occurrence has three events. As we can see, an update model for an announcement action and a frame of reference is structure-wise identical to the update model for a sensing action and a frame of reference. The main distinction lies in the set of designated events in the update template for each type of actions. There is only one single designated event for announcement actions while there are two for sensing actions.

Example 8. *Let us consider the occurrence of* $peek\langle A\rangle$ *in the state described in Figure 4 (left). The frame of reference for this occurrence of* $peek\langle A\rangle$ *is* $(\{A\}, \{B\}, \{C\})$. *The corresponding update template is given in Figure 4 (middle).*

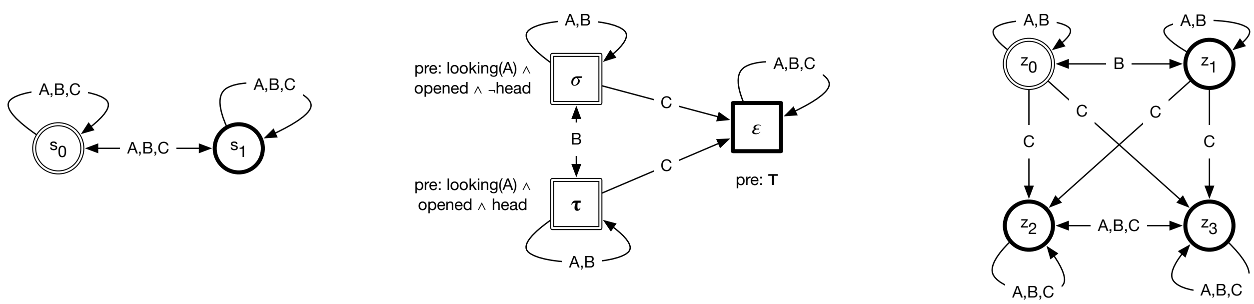

Figure 4: Update template $(\omega(peek\langle A\rangle, (\{A\}, \{B\}, \{C\})), \{\sigma, \tau\})$ and its application

In the above figure, $z_0 = (s_0, \sigma)$, $z_1 = (s_1, \tau)$, $z_2 = (s_0, \epsilon)$, *and* $z_3 = (s_1, \epsilon)$ *and the interpretation of each world* z_i *is the same as the interpretation of the world* s_j *where* $z_i = (s_j, x)$ *for* $x \in \{\sigma, \tau, \epsilon\}$.

The next example illustrates the update template of announcement actions.

Example 9. *Let us assume that* A *and* B *have agreed to a scheme of informing each other if the coin lies heads up by raising a hand.* B *can only observe* A *if* B *is looking at the box (or looking at* A*).* C *is completely ignorant about the meaning of* A*'s raising her hand. This can be modeled by* *the following statements:*[11]

$$\begin{array}{l} \textbf{executable } raising\_hand\langle A\rangle \textbf{ if } \mathbf{B}_A(head), head \\ raising\_hand\langle A\rangle \textbf{ announces } head \\ A \textbf{ observes } raising\_hand\langle A\rangle \textbf{ if } \top \\ B \textbf{ observes } raising\_hand\langle A\rangle \textbf{ if } looking(B) \end{array}$$

If A knows the coin lies heads up and raises her hand, B will be aware that the coin lies heads up and C is completely ignorant about this.

Let us consider the action occurrence raising_hand$\langle A\rangle$ and the state in which B is looking at the box and thus both A and B are aware of it. We have that the frame of reference is $(\{A, B\}, \emptyset, \{C\})$ and thus the update template for the occurrence of raising_hand$\langle A\rangle$ is shown in Figure 5.

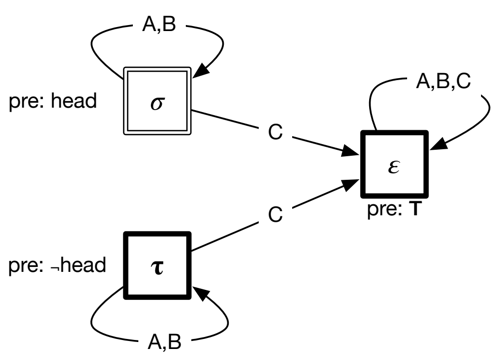

Figure 5: Update template for the *raising_hand*$\langle A\rangle$ action and $\rho = (\{A, B\}, \emptyset, \{C\})$

4.3. *Defining* Φ_D

The update models representing action occurrences can be used in formalizing Φ_D similar to the proposals in Baral et al. (2012, 2013). *However, a main issue of the earlier definitions is that the early definition does not deal with false/incorrect beliefs*. Let us address this with a novel solution, inspired by the suggestion in (van Eijck, 2017). Let us start by introducing some additional notation. For a pointed Kripke model (M, s), an agent $i \in \mathcal{AG}$, and a formula φ, we say that i has false belief about φ in (M, s) if

$$(M, s) \models \varphi \text{ and } (M, s) \models \mathbf{B}_i \neg\varphi.$$

For a set of agents S, a pointed Kripke model (M, s), and a formula φ, such that $(M, s) \models \varphi$, let $M[S, \varphi]$ be obtained from M by replacing $M[i]$ with $M[S, \varphi][i]$ where

- $M[S, \varphi][i] = (M[i] \setminus M[i]^s) \cup \{(s, s)\}$ for $i \in S$ and $(M, s) \models \mathbf{B}_i \neg\varphi$ where $M[i]^s = \{(s, u) \mid (s, u) \in M[i]\}$; and
- $M[S, \varphi][i] = M[i]$ for other agents, i.e., $i \in \mathcal{AG} \setminus S$ or $i \in S$ and $(M, s) \not\models \mathbf{B}_i \neg\varphi$.

[11]For simplicity, we ignore the effect that A's hand is raised when A raises her hand.

This process aims at correcting beliefs of agents with false beliefs. Intuitively, the links in $M[i]^s$ create the false belief of agent i. Therefore, to correct the false believe of i, we should replace them with the single link (s, s). Let us now define Φ_D.

Definition 15. *Let D be a m$\mathcal{A}^*$ domain and* a *be an action instance. Let ψ be precondition of* a*, (M, s) a state, and α a set of agents. Let us consider $\mathsf{a} \in \mathcal{AI}$. We say* a *is* executable *in (M, s) if $(M, s) \models \psi$. The result of executing* a *in (M, s) is a set of states, denoted by $\Phi_D(\mathsf{a}, (M, s))$ and defined as follows.*

- *If* a *is not executable in (M, s) then $\Phi_D(\mathsf{a}, (M, s)) = \emptyset$*
- *If* a *is executable in (M, s) and $(\mathcal{E}, E_d)$ is the representation of the occurrence of* a *in (M, s) then*
  - $\Phi_D(\mathsf{a}, (M, s)) = (M, s) \otimes (\mathcal{E}, E_d)$ *if* a *is a world-altering action instance;*
  - $\Phi_D(\mathsf{a}, (M, s)) = M_1[F_D(\mathsf{a}, M_1, s), \varphi] \otimes (\mathcal{E}, E_d)$ *where $M_1 = M[P_D(\mathsf{a}, M, s), \psi]$ if* a *is a sensing action instance that senses φ and $(M, s) \models \varphi$;*
  - $\Phi_D(\mathsf{a}, (M, s)) = M_1[F_D(\mathsf{a}, M_1, s), \neg\varphi] \otimes (\mathcal{E}, E_d)$ *where $M_1 = M[P_D(\mathsf{a}, M, s), \psi]$ if* a *is a sensing action instance that senses φ and $(M, s) \models \neg\varphi$; and*
  - $\Phi_D(\mathsf{a}, (M, s)) = M_1[F_D(\mathsf{a}, M_1, s), \varphi] \otimes (\mathcal{E}, E_d)$ *where $M_1 = M[P_D(\mathsf{a}, M, s), \psi]$ if* a *is an announcement action instance that announces φ and $(M, s) \models \varphi$.*

Finally, for a set of states $\mathcal{M}$,

- *if* a *is not executable in some $(M, s) \in \mathcal{M}$ then $\Phi_D(\mathsf{a}, \mathcal{M}) = \emptyset$;*
- *if* a *is executable in every $(M, s) \in \mathcal{M}$ then*

$$\Phi_D(\mathsf{a}, \mathcal{M}) = \bigcup_{(M,s) \in \mathcal{M}} \Phi_D(\mathsf{a}, (M, s)).$$

Observe that the definition of Φ_D for an announcement or a sensing action occurrence corrects the false beliefs of the full and partial observers[12]. $M[P_D(\mathsf{a}, M, s), \psi]$ corrects the false beliefs about the action precondition of partially observers and $M_1[F_D(\mathsf{a}, M_1, s), \varphi]$ corrects the false beliefs about the sensed (or announced) formula.

4.4. Properties of Φ_D

While the syntax and semantics of m$\mathcal{A}^*$ represent contributions on their own, of particular interest is the fact that m$\mathcal{A}^*$ satisfies certain useful properties—specifically its ability to correctly capture certain intuitions concerning the effects of various types of actions. In particular,

- If an agent is fully observant of the execution of an action instance then her beliefs will be updated with the effects of such action occurrence;

[12] We thank the anonymous reviewer for suggesting the improvement of this definition.

- An agent who is only partially observant of the action occurrence will believe that the agents who are fully observant of the action occurrence are certain about the action's effects; and
- An agent who is oblivious of the action occurrence will also be ignorant about its effects.

We will next present several theorems discussing these properties. To simplify the presentation, we will use the following notations throughout the theorems in this subsection.

- D denotes a consistent m$\mathcal{A}^*$ domain;
- (M, s) denotes a state; and
- a is an action instance, whose precondition is given by the statement

$$\textbf{executable } \mathsf{a} \textbf{ if } \psi$$

  in D, and a is executable in (M, s).
- $\rho = (F, P, O)$ is the frame of reference of the execution of a in (M, s) where $F = F_D(\mathsf{a}, M, s)$, $P = P_D(\mathsf{a}, M, s)$, and $O = O_D(\mathsf{a}, M, s)$.

We begin with a theorem about the occurrence of the instance of an world-altering action.

Theorem 2. *Assume that* a *is an world-altering action instance. It holds that:*

1. *for every agent* $x \in F_D(\mathsf{a}, M, s)$ *and* $[\mathsf{a} \textbf{ causes } \ell \textbf{ if } \varphi]$ *belongs to* D*, if* $(M, s) \models \mathbf{B}_x\varphi$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x\ell$*;*
2. *for every agent* $y \in O_D(\mathsf{a}, M, s)$ *and a belief formula* η*,* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_y\eta$ *iff* $(M, s) \models \mathbf{B}_y\eta$*; and*
3. *for every pair of agents* $x \in F_D(\mathsf{a}, M, s)$ *and* $y \in O_D(\mathsf{a}, M, s)$ *and a belief formula* η*, if* $(M, s) \models \mathbf{B}_x\mathbf{B}_y\eta$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x\mathbf{B}_y\eta$*.*

Proof. See Appendix A. □

In the above theorem, the first property discusses the changes in the beliefs of agents who are fully observant of the occurrence of an world-altering action instance. The second property shows that oblivious agents are still in the "old state," i.e., they believe nothing has happened. The third property indicates that fully observant agents are also aware that the beliefs of all oblivious agents have not changed. This is particular useful in situations where an agent would like to create false beliefs about a fluent p for other agents: she only needs to secretly execute an action that changes the truth value of p.

Theorem 3. *Let us assume that* a *is a sensing action instance and* D *contains the statement* a **determines** φ*. It holds that:*

1. *if* $(M, s) \models \varphi$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi$*;*
2. *if* $(M, s) \models \neg\varphi$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi$*;*
3. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{P_D(\mathsf{a},M,s)}(\mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi)$*;*
4. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}(\mathbf{C}_{P_D(\mathsf{a},M,s)}(\mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi))$*;*
5. *for every agent* $y \in O_D(\mathsf{a}, M, s)$ *and formula* η*,* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_y\eta$ *iff* $(M, s) \models \mathbf{B}_y\eta$*;*

6. *for every pair of agents* $x \in F_D(\mathsf{a}, M, s)$ *and* $y \in O_D(\mathsf{a}, M, s)$ *and a formula* η *if* $(M, s) \models \mathbf{B}_x \mathbf{B}_y \eta$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x \mathbf{B}_y \eta$.

Proof. See Appendix A. □

The first and second properties of the above theorem indicate that agents who are fully aware of the occurrence of the sensing action instance will be able to update their beliefs with the truth value in the real state of the world of the sensed fluent, thereby correcting any false beliefs that they might have before the execution of the action. The third property shows that agents who are partially aware of the action execution will know that agents who are fully aware of the action execution will have the correct beliefs about the sensed fluents. The fourth property indicates that fully aware agents know that partially observed agents would know that they have the correct beliefs about the sensed fluent. The fifth and sixth properties are about oblivious agents' beliefs.

Theorem 4. *Assume that* a *is an announcement action instance and* D *contains the statement* a **announces** φ. *If* $(M, s) \models \varphi$ *then it holds that*

1. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)} \varphi$;
2. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{P_D(\mathsf{a},M,s)} (\mathbf{C}_{F_D(\mathsf{a},M,s)} \varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)} \neg\varphi)$;
3. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)} (\mathbf{C}_{P_D(\mathsf{a},M,s)} (\mathbf{C}_{F_D(\mathsf{a},M,s)} \varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)} \neg\varphi))$;
4. *for every agent* $y \in O_D(\mathsf{a}, M, s)$ *and a formula* η, $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_y \eta$ *iff* $(M, s) \models \mathbf{B}_y \eta$; *and*
5. *for every pair of agents* $x \in F_D(\mathsf{a}, M, s)$ *and* $y \in O_D(\mathsf{a}, M, s)$ *and a formula* η, *if* $(M, s) \models \mathbf{B}_x \mathbf{B}_y \eta$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x \mathbf{B}_y \eta$.

Proof. The proof of this theorem is similar to the proof of Theorem 3 and is omitted for brevity. □

Similarly to Theorem 3, the first property of the above theorem indicates that a truthful announcement could help agents who are fully aware of the action instance occurrence correct their false beliefs. They also know that partially aware agents will know that they have the correct beliefs. Likewise, partially aware agents will only know that fully aware agents know the truth value of the announced formula but they might not have the real value of this formula themselves. Furthermore, as in other types of actions, the beliefs of oblivious agents do not change.

4.5. Entailment in m$\mathcal{A}^*$ Action Theories

We are now ready to define the notion of entailment in m$\mathcal{A}^*$ action theories. It will be defined between m$\mathcal{A}^*$ action theories and queries of the following form:

$$\varphi \textbf{ after } \delta \tag{13}$$

where φ is a formula and δ is a *sequence of action instances* $\mathsf{a}_1; \dots; \mathsf{a}_n$ ($n \geq 0$); we will refer to such type of sequences of action instances as *plans*. Let us observe that the entailment can be easily extended to consider more general forms of *conditional plans,* that include conditional statements (e.g., **if-then**) or loops (e.g., **while**)—as discussed in e.g. (Levesque et al., 1997; Son and Baral, 2001). We leave these relatively simple extensions for future work.

The description of an evolution of a system will deal with *belief state* (Definition 11). For a belief state B and an action instance a, we say that a is executable in B if $\Phi_D(\mathsf{a}, (M,s)) \neq \emptyset$ for every state (M,s) in B. With a slight abuse of notation, we define

$$\Phi_D(\mathsf{a}, B) = \begin{cases} \{\bot\} & \text{if } \Phi_D(\mathsf{a},(M,s)) = \emptyset \text{ in some state } (M,s) \text{ in } B \\ & \text{or } B = \{\bot\} \\ \bigcup_{(M,s)\in B} \Phi_D(\mathsf{a},(M,s)) & \text{otherwise} \end{cases} \tag{14}$$

where $\{\bot\}$ denotes that the execution of a in B fails. Note that we assume that no action instance is executable in $\bot$.

Let δ be a plan and B be a belief state. The set of belief states resulting from the execution of δ in B, denoted by $\Phi^*_D(\delta, B)$, is defined as follows:

- If δ is the empty plan $[\,]$ then $\Phi^*_D([\,], B) = B$;
- If δ is a plan of the form $\mathsf{a}; \delta'$ (with $\mathsf{a} \in \mathcal{AI}$), then $\Phi^*_D(\mathsf{a};\delta', B) = \Phi^*_D(\delta', \Phi_D(\mathsf{a}, B))$.

Intuitively, the execution of δ in B can go through several paths, each path might finish in a set of states. It is easy to see that if one of the states reached on a path during the execution of δ is $\bot$ (the failed state) then the final result of the execution of δ in B is $\{\bot\}$. $\Phi^*_D(\delta, B) = \{\bot\}$ indicates that the execution of δ in B fails.

We are now ready to define the notion of entailment.

Definition 16 (Entailment)**.** *An action theory* (I,D) *entails the query*

$$\varphi \textbf{ after } \delta$$

denoted by $(I,D) \models \varphi \textbf{ after } \delta$ *if*

1. $\Phi^*_D(\delta, I_0) \neq \{\bot\}$ *and*
2. $(M,s) \models \varphi$ *for each* $(M,s) \in \Phi^*_D(\delta, I_0)$

where I_0 *is the initial belief state of* (I,D).

We say that (I,D) **S5**-*entails the query* φ **after** δ*, denoted by* $(I,D) \models_{\mathbf{S5}} \varphi \textbf{ after } \delta$*, if the two conditions (1)–(2) are satisfied with respect to* I_0 *being the initial* **S5**-*belief state of* (I,D).

4.6. Using m$\mathcal{A}^*$: An Illustration

The next example illustrates these definitions.

Example 10. *Let* D_1 *be the domain specification given in Examples 3 and 4 and* I_1 *be the set of initial statements given in Example 5. Furthermore, let* δ_A *be the sequence of actions:*

$$\delta_A = distract(C)\langle A\rangle; open\langle A\rangle; peek\langle A\rangle.$$

We can show that

$$(I_1, D_1) \models_{\mathbf{S5}} (\mathbf{B}_A head \vee \mathbf{B}_A \neg head) \wedge \mathbf{B}_A(\mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head)) \textbf{ after } \delta_A$$
$$(I_1, D_1) \models_{\mathbf{S5}} \mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head) \wedge \neg \mathbf{B}_B head \wedge \neg \mathbf{B}_B \neg head \textbf{ after } \delta_A$$
$$(I_1, D_1) \models_{\mathbf{S5}} \mathbf{B}_C[\textstyle\bigwedge_{i\in\{A,B,C\}}(\neg \mathbf{B}_i head \wedge \neg \mathbf{B}_i \neg head)] \textbf{ after } \delta_A$$

It can be shown that (I_1, D_1) is indeed a definite action theory and any **S5***-initial state of (I_1, D_1) is equivalent to either (M_0, s_0) or (M_0, s_1) where (M_0, s_0) is drawn in Figure 6 (left) and*
$M_0[\pi](s_0) = \{has\_key(A), looking(A), looking(B), looking(C)\}$ *and*
$M_0[\pi](s_1) = \{has\_key(A), looking(A), looking(B), looking(C), head\}$.

The execution of $distract(C)\langle A\rangle$ in (M_0, s_0) results in a new state (M_1, u_0) and is shown in Figure 6 (right). The update model corresponds to the occurrence of $distract(C)\langle A\rangle$ in (M_0, s_0) is shown in the middle of Figure 6, and the interpretations associated with the worlds in M_1 are:

$u_0 = (s_0, \sigma)$ *with* $M_1[\pi](u_0) = \{looking(A), looking(B), has\_key(A)\}$
$u_1 = (s_1, \sigma)$ *with* $M_1[\pi](u_1) = \{looking(A), looking(B), head, has\_key(A)\}$.

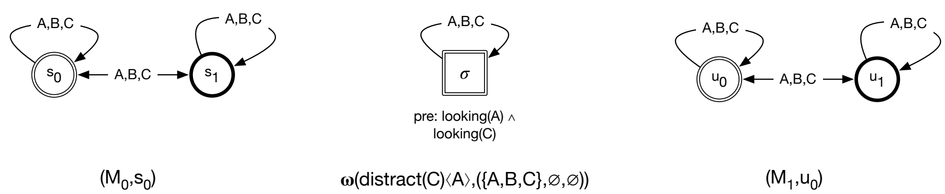

Figure 6: Execution of $distract(C)\langle A\rangle$ in (M_0, s_0) results in (M_1, u_0)

The execution of $open\langle A\rangle$ in (M_1, u_0) (left, Figure 7) results in a new state (M_2, v_0) (right, Figure 7). The update model corresponds to the occurrence of $open\langle A\rangle$ in (M_1, u_0) is shown in the middle of Figure 7. The interpretations associated to each world of (M_2, v_0) are as follows:

$v_0 = (u_0, \sigma)$ *with* $M_2[\pi](v_0) = \{looking(A), looking(B), has\_key(A), opened\}$
$v_1 = (u_1, \sigma)$ *with* $M_2[\pi](v_1) = \{looking(A), looking(B), head, has\_key(A), opened\}$
$v_2 = (u_0, \epsilon)$ *with* $M_2[\pi](v_2) = \{looking(A), looking(B), has\_key(A)\}$
$v_3 = (u_1, \epsilon)$ *with* $M_2[\pi](v_3) = \{looking(A), looking(B), head, has\_key(A)\}$.

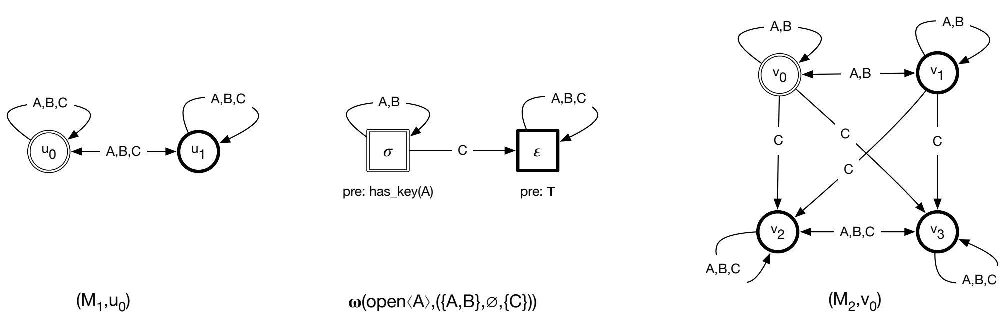

Figure 7: Execution of $open\langle A\rangle$ in (M_1, u_0) results in (M_2, v_0)

Finally, the execution of $peek\langle A\rangle$ in (M_2, v_0) results in (M_3, z_0) (Figure 8) where $z_0 = (v_0, \sigma)$, $z_1 = (v_1, \tau)$, $z_2 = (v_0, \epsilon)$, $z_3 = (v_1, \epsilon)$, $z_4 = (v_2, \epsilon)$, and $z_5 = (v_3, \epsilon)$ with $M_3[\pi](z_0) = M_2[\pi](v_0)$, $M_3[\pi](z_1) = M_2[\pi](v_1)$, $M_3[\pi](z_2) = M_2[\pi](v_0)$, $M_3[\pi](z_3) = M_2[\pi](v_1)$, $M_3[\pi](z_4) = M_2[\pi](v_2)$, and $M_3[\pi](z_5) = M_2[\pi](v_3)$.

It is easy to see that the execution of δ_A in (M_0, s_0) results in only one state (M_3, z_0). We can

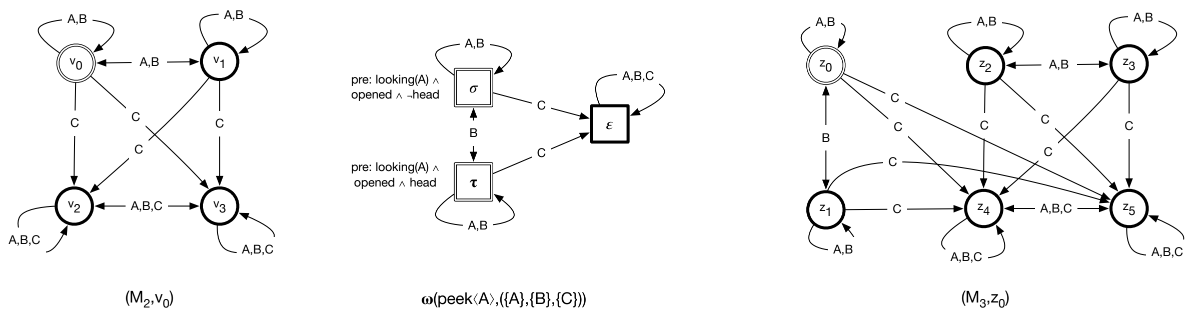

Figure 8: Execution of $peek\langle A\rangle$ in (M_2, v_0) results in (M_3, z_0)

verify that

$$
\begin{aligned}
(M_3, z_0) &\models \mathbf{B}_A \neg head \wedge \mathbf{B}_A(\mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head)) \\
(M_3, z_0) &\models \mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head) \wedge (\neg \mathbf{B}_B head \wedge \neg \mathbf{B}_B \neg head) \\
(M_3, z_0) &\models \mathbf{B}_C[\textstyle\bigwedge_{i \in \{A,B,C\}}(\neg \mathbf{B}_i head \wedge \neg \mathbf{B}_i \neg head)]
\end{aligned}
\tag{15}
$$

We conclude the example with a note that the execution of δ_A *in* (M_0, s_1) *results in a state* (V_3, z_1) *whose Krippe structures* V_3 *is identical to* M_3 *and the real state of the world is* z_1 *instead of* z_0 *(Figure 9).*

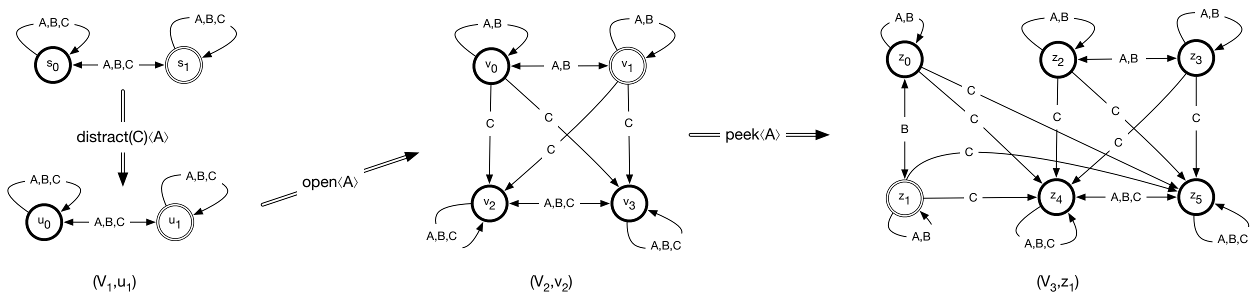

Figure 9: Execution of δ_A in (M_0, s_1) results in (V_3, z_1)

We can also verify that the following holds:

$$
\begin{aligned}
(V_3, z_2) &\models \mathbf{B}_A head \wedge \mathbf{B}_A(\mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head)) \\
(V_3, z_2) &\models \mathbf{B}_B(\mathbf{B}_A head \vee \mathbf{B}_A \neg head) \wedge (\neg \mathbf{B}_B head \wedge \neg \mathbf{B}_B \neg head) \\
(V_3, z_2) &\models \mathbf{B}_C[\textstyle\bigwedge_{i \in \{A,B,C\}}(\neg \mathbf{B}_i head \wedge \neg \mathbf{B}_i \neg head)]
\end{aligned}
\tag{16}
$$

(15) *and* (16) *prove the conclusions of the example.*

4.7. Some Considerations in Using m$\mathcal{A}^*$

The previous example shows that m$\mathcal{A}^*$ is adequate for the specifying and reasoning about the actions in the domain in the introductory example. Furthermore, it has been demonstrated that the language can be used in specifying and reasoning about the effects of typical actions in multi-agent setting (see, Appendix B in an earlier version of this paper `https://arxiv.org/pdf/1511.01960v2.pdf`). The use of m$\mathcal{A}^*$, similar to the use of action languages in single-agent setting (e.g., $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, etc.), requires some considerations. A slight change in the action or the observability specifications statements can easily lead to different results. We discuss below a variant of the theory D_1 in Example 10 that highlights this issue.

Let D_1^* be the domain specification given in Examples 3 and 4 without the two statements about the observability

$$z \textbf{ observes } distract(y)\langle x\rangle \textbf{ if } looking(z) \qquad z \textbf{ observes } signal(y)\langle x\rangle \textbf{ if } looking(z) \quad (17)$$

and I_1 be the set of initial statements given in Example 5. Furthermore, let $\delta_A^* = distract(C)\langle A\rangle; open\langle A\rangle$ and (M_0, s_0) be an initial state of (D_1^*, I_1), as described in Example 10, respectively. The execution of δ_A^* in (M_0, s_0) and the update models corresponding to the occurrences of the action occurrences in δ_A^* are shown in Figure 10. The states (M_1^*, u_0)

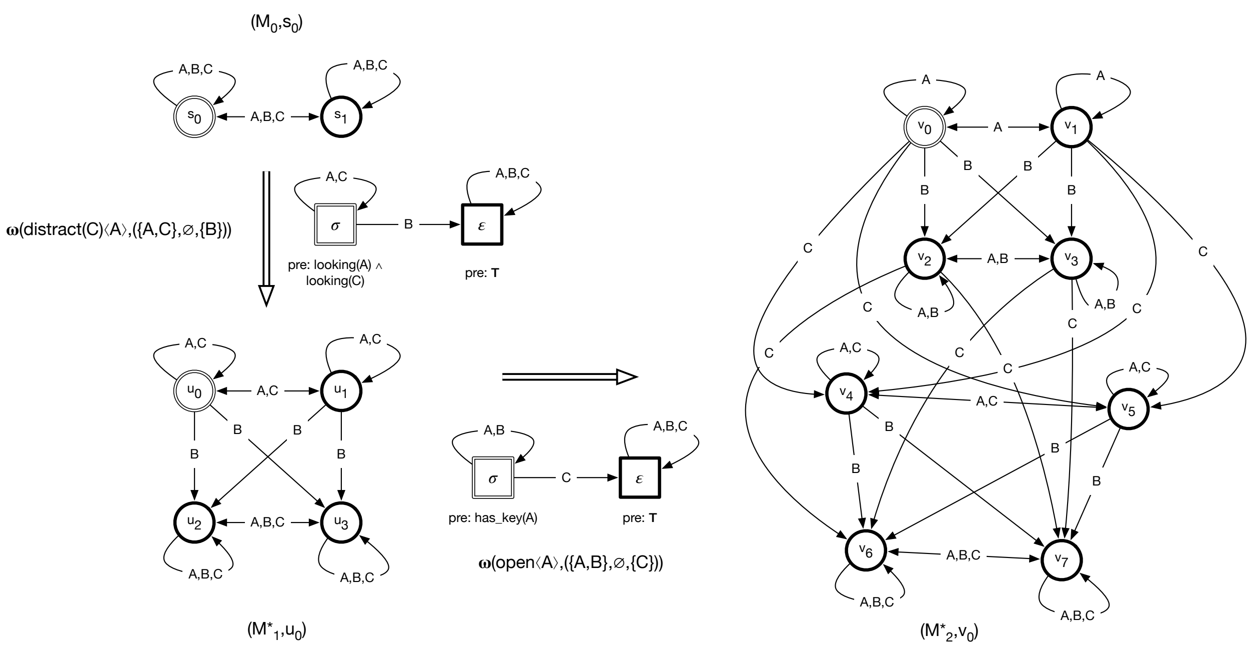

Figure 10: Executing δ_A^* in (M_0, s_0)

and (M_2^*, v_0) are results of executing $distract(C)\langle A\rangle$ and δ_A^* in (M_0, s_0), respectively, where the interpretations of the worlds in these states are:
$u_0 = (s_0, \sigma)$ with $M_1^*[\pi](u_0) = \{looking(A), looking(B), has\_key(A)\}$
$u_1 = (s_1, \sigma)$ with $M_1^*[\pi](u_1) = \{looking(A), looking(B), head, has\_key(A)\}$
$u_2 = (s_0, \epsilon)$ with $M_1^*[\pi](u_2) = \{looking(A), looking(B), looking(C), has\_key(A)\}$
$u_3 = (s_1, \epsilon)$ with $M_1^*[\pi](u_3) = \{looking(A), looking(B), looking(C), head, has\_key(A)\}$
and
$v_0 = (u_0, \sigma)$ with $M_2^*[\pi](v_0) = \{looking(A), looking(B), has\_key(A), opened\}$
$v_1 = (u_1, \sigma)$ with $M_2^*[\pi](v_1) = \{looking(A), looking(B), head, has\_key(A), opened\}$

$v_2 = (u_2, \sigma)$ with $M_2^*[\pi](v_2) = \{looking(A), looking(B), looking(C), has\_key(A), opened\}$
$v_3 = (u_3, \sigma)$ with $M_2^*[\pi](v_3) = \{looking(A), looking(B), looking(C), head, has\_key(A), opened\}$
$v_4 = (u_0, \epsilon)$ with $M_2^*[\pi](v_4) = \{looking(A), looking(B), has\_key(A)\}$
$v_5 = (u_1, \epsilon)$ with $M_2^*[\pi](v_5) = \{looking(A), looking(B), head, has\_key(A)\}$
$v_6 = (u_2, \epsilon)$ with $M_2^*[\pi](v_6) = \{looking(A), looking(B), looking(C), has\_key(A)\}$
$v_7 = (u_3, \epsilon)$ with $M_2^*[\pi](v_7) = \{looking(A), looking(B), looking(C), head, has\_key(A)\}$

Because of the differences in the observability specification between D_1^* and D_1, the update model corresponds to the execution of $distract(C)\langle A\rangle$ in (M_0, s_0) is different than its counterpart in Example 10, in which B is oblivious of the occurrence of the action A distracts C. Therefore, by the inertial principle of beliefs, B should believe that C is looking at the box after the occurrence of $distract(C)\langle A\rangle$. As such, one might argue that B should believe that C will observe the occurrence of $open\langle A\rangle$, and hence, B should believe that C believes that the box is open after the execution of δ_A^* in (M_0, s_0). In other words, the following is a reasonable conclusion[13]

$$(M_2^*, v_0) \models \mathbf{B}_B \mathbf{B}_C\ opened \tag{18}$$

On the other hand, it is easy to verify that the following holds

$$(M_2^*, v_0) \models \mathbf{B}_B \mathbf{B}_C\ \neg opened \tag{19}$$

Interestingly, this results does not affect the final outcome of the intended plan of A discussed in the introductory example. In other words, if A were to execute the action of peeking into the box, A will achieve its goal of letting B knows that A knows whether the coin lies heads up and C does not know that A knows. The detailed computation can be found in `https://arxiv.org/pdf/1511.01960v2.pdf`.

There could be different views on the difference between (19) and (18).

On the one hand, one might argue that the semantics of the language yields a counter intuitive result. One proposal, as suggested by a reviewer, is to make the action occurrence of $distract(C)\langle A\rangle$ to be observable to all agents in D_1^* as well, i.e., treating $\omega(distract(C)\langle A\rangle, (\{A, C\}, \emptyset, \{B\}))$ identical with $\omega(distract(C)\langle A\rangle, (\{A, B, C\}, \emptyset, \emptyset))$. Intuitively, this could be achieved by replacing the observability statement about this action in D_1^* with

$$z \textbf{ observes } distract(y)\langle x\rangle \qquad (*)$$

Observe that doing so would remove the problem displayed in (19). However, it is not adequate in the following situation. Consider the state (M_0', s_0') where M_0' has two worlds s_0' and s_1' and the same structure as (M_0, s_0) with $M_0'[\pi](s_0') = M_0[\pi](s_0) \setminus \{looking(B)\}$ and $M_0'[\pi](s_1') = M_0[\pi](s_1) \setminus \{looking(B)\}$, i.e., B is not looking at the box in (M_0', s_0') and this is common knowledge. The question is whether B should be allowed to be a full observer of the occurrence of $distrac(C)\langle A\rangle$ in (M_0', s_0'). If we were to allow B to be a full observer of this action occurrence in (M_1^*, s_0), then we should do so for (M_0', s_0'); and, by virtue of (*), this is something expected. This,

[13] We thank the anonymous reviewer of an earlier version of this paper who raised this question and suggested a possible solution.

in our view is a counter intuitive outcome if occurrences of reference setting actions are observable to every agent.

On the other hand, one can attribute the difference between (19) and (18) to the difference between the two action theories D_1 and D_1^*, i.e., the missing of the statements in (17) in D_1^*. This omission leads to different frame of references, and therefore, different update models for $distract(C)\langle A\rangle$ in (M_0, s_0) in the two theories. In other words, the specification in D_1^* does not represent the observability of agents adequately. As such, in designing a m$\mathcal{A}^*$ domain, it would be useful to take into consideration the agents or groups of agents who would employ the domain in its reasoning. We note that the current m$\mathcal{A}^*$ does not consider beliefs and knowledge of agents in the specification of observability statements (see, Section 5, for a more detailed discussion) and this is an interesting issue that we leave for the future development of m$\mathcal{A}^*$.

5. Related Work and Discussion

In this section, we connect our work to related efforts in reasoning about actions and their effects in multi-agent domains. The background literature spans multiple areas. We will give a quick introduction and focus our attention on the most closely related works.

5.1. Relating m$\mathcal{A}^*$ and DEL and Update Model-based Languages

Throughout the paper, we refer to update models and employ them in defining the semantics of m$\mathcal{A}^*$ as the use of update models has been well-accepted in reasoning about dynamic multi-agent systems. In the following, we will sometimes use "DEL specification" to refer to both update model- and DEL-based specification as the concept of action models originates from the study of dynamic epistemic logic. Whenever it is needed, we will refer to the specific formalization.

The key distinction between m$\mathcal{A}^*$ specification and DEL specification lies in the fact that the specification of effects of actions in m$\mathcal{A}^*$ emphasizes the distinction between *direct* and *indirect* effects of an action occurrence while DEL specification focuses on *all* effects. To reiterate, in m$\mathcal{A}^*$,

- the direct effects of a world-altering action occurrence are the changes of the world that are directly caused by the execution of an action; for instance, the direct effect of $open\langle A\rangle$ is the box is opened;
- the direct effects of a sensing action occurrence are the truth values of the sensed formula; for instance, the direct effect of $peek\langle A\rangle$, when the coin lies tails up, is "$tail$ is true";
- the direct effect of a truthful announcement action occurrence is that the truth value of the announced formula is true; for instance, the direct effect of $shout\_tail\langle A\rangle$ is that the information "$tail$ is true" was announced.

In all cases, the indirect effects of an action occurrence are the changes in the beliefs of agents who might be aware of the action occurrence. Such indirect effects are specified by the set of statements of the form (9) and (10). We note that even in single agent domains world-altering actions could also create indirect effects and this has been well studied (see, e.g., Giunchiglia et al. (1997); Shanahan (1999)). The advantage of directly dealing with indirect effects—in the form of state constraints—in planning has been discussed in (Thiebaux et al., 2003; Tu et al., 2011). This type of indirect effects of actions could be included in m$\mathcal{A}^*$ by adding statements, referred as *static causal laws*, of the form

$$\varphi \textbf{ if } \psi$$

where φ and ψ are fluent formulae. We have conducted a preliminary investigation of this issue in an earlier version of m$\mathcal{A}^*$ in Baral et al. (2013). We decided to keep it out of this paper for the simplicity of the presentation.

Contrary to the design of m$\mathcal{A}^*$, formulae in the update model language aim at specifying *all effects* of an action occurrence. To see it, let us consider the action $peek\langle A\rangle$. It is generally accepted that this action helps A—under the right circumstances—learn whether or not the coin lies heads or tails up. Since the effects of this action on the knowledge of other agents are different in different situations (pointed Kripke structure), different formulae will need to be developed to describe the effects of this particular action (more on this in Example 11). In m$\mathcal{A}^*$, such indirect effects of action occurrences are encoded in statements of the form (9) and (10). Due to the restrictions imposed on these two types of statements, some different effects of actions that could be represented using update models cannot be described in m$\mathcal{A}^*$ (c.f., Section 5.1.2).

The reader should note that in this discussion *two different kinds of objects* are touched on: *action occurrences* and *actions*. This is intentional – in the update model language, the *kinds of objects* being discussed are rightly thought of as action occurrences (i.e., individual members of a more abstract type); whereas in m$\mathcal{A}^*$ (and action language approach more generally), the fundamental abstraction is that of an *action* (seen as a *type*, of which distinct occurrences are members). The difference in focus leads to a much greater simplicity of action descriptions in the presence of multiple agents.

Let us consider the simplified version of the coin in a box problem as presented in Example 2—with three agents A, B, and C, a box containing a coin, and initially it is common knowledge that A knows whether the coin lies heads or tails up and B and C do not. Let us assume that agent A announces that the coin lies heads up. In our formalism, we express the action of A as *an instance* $shout\_head\langle A\rangle$ of the action *shout_head* and it can result in different update models (Figure 2). In DEL, the update model *for the same action occurrence* (i.e., $shout\_head\langle A\rangle$) will also need to include additional information about all three agents A, B, and C encoding their "roles" or "perspectives" (more in Section 5.1.1). By roles or perspectives we mean information such as who executes the action occurrence, who observes (partially observes the action occurrence), or who is oblivious must be accompanied with the update models; for example, without additional information, it is not clear who among A or B executes the action of announcing that the coin lies heads up given the update model on the right of Figure 2.

Thus, a critical difference between the m$\mathcal{A}^*$ approach to representing multi-agent actions and the approach used in DEL with update models lies in the way we encode the information about agents roles and perspectives, which need to be accompanied with the description of update models[14], as part of the state in m$\mathcal{A}^*$. There are some important implications of such difference and we discuss them in the following subsections.

[14]In a recent paper, (Bolander, 2018) proposed to encode this information as edge-conditions of update models.

5.1.1. Compactness of Representation

As we discussed in Section 4, each action occurrence in m$\mathcal{A}^*$ corresponds to an update model. As such, any m$\mathcal{A}^*$ domain could be represented as a theory in update model language. Since each action occurrence is associated with a pointed Kripke structure, it follows that, theoretically, we might need *exponentially many formulae* for representing a m$\mathcal{A}^*$ domain. The next example shows that we need an exponential number of update models to describe an action with a linear number of "indirect effects" in m$\mathcal{A}^*$ domain.

Example 11. *Let us consider, for example, the action occurrence peek*$\langle A \rangle$ *from domain* D_1*. To describe different scenarios related to this action occurrence, we need to have an update model for all of the following cases:*

- *Both* B *and* C *are looking;*
- *Either* B *or* C *is looking but not both; and*
- *Both* B *and* C *are not looking.*

In our approach, the above example is specified in a very different way: the action is about sensing $head$ *(or* $tail$*). The agents who sense it, who observe the sensing take place, and who are oblivious can be specified directly or can be specified indirectly in terms of* conditions, *such as which agents are near the sensing, which ones are watching from far, and which ones are looking away, respectively. As such, to specify all four cases, we write*

$$B \textbf{ aware\_of } peek\langle A\rangle \textbf{ if } looking(B) \quad \text{and} \quad C \textbf{ aware\_of } peek\langle A\rangle \textbf{ if } looking(C).$$

It is easy to see that if we have n *agents and agent* A *executes the action* $peek$ *then if we want to include the various possibilities in the finite set of action models that can be used for planning in the context of Bolander and Andersen (2011), we will need* 2^{n-1} *action models for specifying all possible consequences of* $peek\langle A\rangle$*. On the other hand, we only need* $n-1$ *statements of the form* (10) *in the action specification part, as the set of looking(X) in our framework is a part of the state.*

It is easy to see that similar conclusions can be made with regards to an occurrence of a world-altering action or an announcement action. In summary, we can say that to represent a m$\mathcal{A}^*$ domain D in the action model language, an exponential number of action models is needed in the worst cases. This advantage of m$\mathcal{A}^*$ can also be seen in representing and reasoning about action sequences. We observe that extensions of DEL have been proposed to address this issue in Bolander (2018) and Engesser et al. (2018).

Narratives and Dynamic Evolution of Multi-agent Actions. Let us consider a scenario with two agents A and B. Initially, agent B is looking at agent A. Agent A lifts a block and, after some time, agent A puts down the block. Some time later, agent B is distracted, say by A, and then agent A again lifts the block. In our formulation, this narrative can be formalized by first describing the initial situation, and then describing the sequence of actions that have occurred, which for this example is:

$$\textit{liftBlock}\langle A\rangle; \textit{putDown}\langle A\rangle; \textit{distract}(B)\langle A\rangle; \textit{liftBlock}\langle A\rangle.$$

The description of this evolution of scenario in DEL is not as simple: each action occurrence will have to be described as an update model containing information about both agents A and B. In addition, such a description (in DEL) will be partly superfluous, as it will have to record information about B looking (or not looking) at A in the update model, while that information is already part of the state. Thus, the approach used in m$\mathcal{A}^*$ to describe this narrative is more natural than the representation in DEL.

Observe that, in our narrative, the action instance *liftBlock*$\langle A \rangle$ appears twice. However, due to the difference in the roles and perspectives over time, the two occurrences of *liftBlock*$\langle A \rangle$ correspond to two different update models. This shows how, using the m$\mathcal{A}^*$ formulation, we can support the dynamic evolution of update models, as result of changes in perspective fluents in the state. In DEL, the two update models are distinct, there is no direct connection between them, and neither one does evolve from the other.

In order to further reinforce this point, let us consider another narrative example. Let us consider a scenario with three agents, A, B, and C. Initially, it is common knowledge that none of the agents knows whether the coin in the box is lying heads up or tails up. In addition, let us assume that initially A and B are looking at the box, while C is looking away. Let us consider the narrative where A peeks into the box; afterwards, A realizes that C is distracted and signals C to look at the box as well; finally A peeks into the box one more time. In m$\mathcal{A}^*$, this situation can be described again by a sequence of actions:

$$\textit{peek}\langle A \rangle ; \textit{signal}(C)\langle A \rangle ; \textit{peek}\langle A \rangle$$

The two occurrences of *peek(*A*)* correspond to two different update models; the second occurrence is an evolution of the first caused by the execution of *signal(C)*$\langle A \rangle$. In DEL, the relevance of the intermediate action *signal(C)*$\langle A \rangle$, and its impact on the second occurrence of *peek*$\langle A \rangle$, is mostly lost—and this results in the use of two distinct update models for $peek\langle A \rangle$ with complete information about the whole action scenario.

The key aspect that allows a natural representation of narratives and evolution of update models in m$\mathcal{A}^*$ is the presence of the agents' perspectives and roles encoded as perspective fluents of a state, and their use to dynamically generate the update models of the actions. While DEL can include perspective fluents as part of the states as well, it does not have a way to take advantage of them in a similar way as m$\mathcal{A}^*$.

5.1.2. Simplicity vs. Expressivity

The formulation adopted in this paper is limited in expressivity to ensure simplicity. It is limited by the (perspective) fluents we have and how we use them. On the other hand, the action model language is more complex and also more expressive. This is also evident in the simplicity of the update models used in m$\mathcal{A}^*$ (Definitions 13–14). This leads to the following limitations of m$\mathcal{A}^*$ in comparison with the action model language:

Complex action occurrences. The simple version of m$\mathcal{A}^*$ as presented here does not consider complex epistemic action occurrences. An example of this type of action occurrences is the **mayread** action in Example 5.4 from (van Ditmarsch et al., 2007). It represents an event that might or might not happen. Let us have a closer look at the action **mayread**. *A*nne and *B*ill are in a cafe.

Some agent brings a letter to Anne. The letter says that United Agents is doing well. *B* leaves the table and orders a drink at the bar so that *A* may have read the letter while he is away. In this example, *A* did not read the letter.

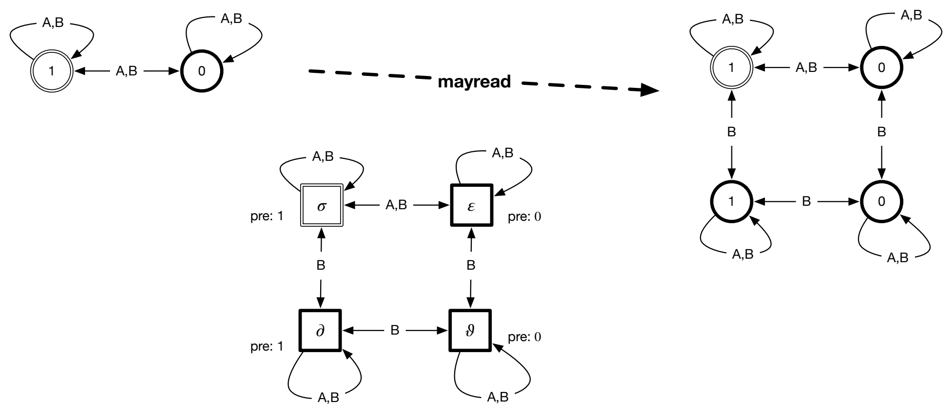

Figure 11: Update model of **mayread** without an equivalent m$\mathcal{A}^*$ representation

Initially, both *A* and *B* do not know the content of the letter sent to *A*. Figure 11 depicts the initial pointed Kripke structure (left) and the pointed Kripke structure after **mayread** (right). An update model representing the action **mayread** is given in the middle of Figure 11. It is easy to see that any update model facilitating this transition will need to have at least four events, two with 1 as the precondition and two with 0 as the precondition. Since any update model in m$\mathcal{A}^*$ needs at most three events, this shows that there exists no equivalent m$\mathcal{A}^*$ representation of **mayread**.

In our action language based framework, **mayread** can be viewed as a combination (disjunction) of two "primitive actions," one is *A reads the letter* and another one is *A does nothing*. One can then use constructs from Golog (Levesque et al., 1997) with m$\mathcal{A}^*$ actions as primitive actions to express **mayread**.

We note that m$\mathcal{A}^*$ also does not consider non-deterministic world-altering actions (e.g., the action of tossing a coin results in the coin lies heads or tails up). This type of actions has been extensively studied in action languages for single-agent domains (see, e.g., Gelfond and Lifschitz (1998)). They can be added to m$\mathcal{A}^*$ easily by allowing ℓ in (6) to be arbitrary formula. Similar to static causal laws, we decide to keep it out of this paper for the simplicity of the presentation in this paper.

Agents' Observability. In m$\mathcal{A}^*$, statements of the form (9) and (10) are used for specifying an agent's observability of action occurrences. It is expressed via fluent formulae and is evaluated with respect to the real-state of the world. In general, such observability can be beliefs of the agent about other agents. Consider the update model[15] in Figure 12 (middle). It represents an action α of an agent A who believes that after she executes the action then both A and B can see an incorrect outcome 1—i.e., A and B are fully observant. In reality, B is oblivious. The initial pointed Kripke

[15]We thank an anonymous reviewer of an earlier version of this paper who suggested a similar example.

structure is given in the left and the result of α is on the right of Figure 12. This shows that, in multi-agent domains, an agent's observability could also be considered as beliefs, and as such affect the beliefs of an agent about other agents' beliefs after the execution of an action. The present m$\mathcal{A}^*$ language does not allow for such specification.

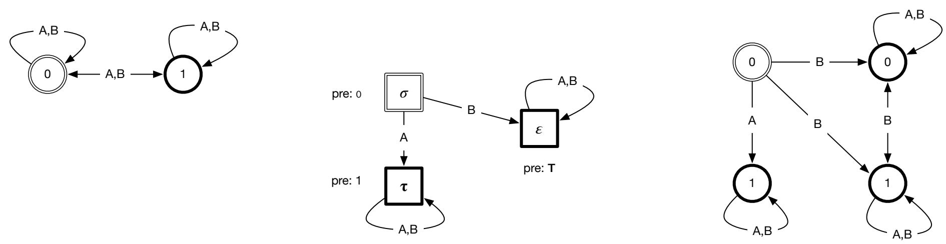

Figure 12: An update model of an action requiring different type of observability statements without an equivalent m$\mathcal{A}^*$ representation

The simplicity of our formulation is by design, and not an inherent flaw of our approach. Indeed, one could envision developing a complete encoding of the complex graph structure of an update model as part of state, using an extended collection of perspective fluents—but, at this time, we do not have a corresponding theory of change to guide us in using these more expressive perspective fluents to capture the full expressive power of update models in DEL. Hence, our current formalism is less expressive than DEL. However, the higher expressiveness of update models provides us with a target to expand m$\mathcal{A}^*$ and capture more general actions.

5.1.3. Other Differences

The previous sections detail the key differences between m$\mathcal{A}^*$ and DEL based on the differences in design and focus of m$\mathcal{A}^*$ and DEL. We next discuss their differences from other angles.

Analogy with Belief Update. Another approach to explore the differences between m$\mathcal{A}^*$ and DEL builds on the analogy to the corresponding differences between *belief updates* and the treatment of actions and change in early action languages (Gelfond and Lifschitz, 1998).

Papers on belief updates define and study the problem of updating a formula ϕ with a formula ψ. In contrast, in reasoning about actions and change, the focus is on defining the resulting state of the world after a particular action is performed in a particular world, given a description of *(i)* how the action may change the world, *(ii)* when the action can be executed; and *(iii)* how the fluents in the world may be (possibly causally) related to each other. In such a context, given a state s and an action a, it is possible to see the determination of the resulting state as the update of s by a formula φ; But, what is important to consider is that the φ is not just the collection of effects of the action a, but incorporates several other components, that take into account the static causal laws as well as which conditions (part of the conditional effects of a) are true in s.

This situation is not dissimilar to the distinction between DEL update models and m$\mathcal{A}^*$. An update model can be encoded by an action formula, and the resulting state can be obtained by updating the starting state with such formula. In DEL, such action formula has to be given directly. Instead, our considerations in m$\mathcal{A}^*$ are in the spirit of the early research in reasoning about actions and change—where we focus on describing actions and their effects, their executability conditions, and where a resulting "state" is determined by applying these descriptions to the "state" where a particular action is performed. Thus, the action formula in this latter case is not explicitly given, but derived from the description of the actions, their effects, and executability conditions.

Taking the analogy further, an important application of reasoning about actions is to determine action sequences or plan structures that achieve a given goal. This is different from searching for a sequence of formulae ψ_i's which, if used to sequentially update a given initial state, will generate a goal state.

Executing Actions. The notion of actions adopted in m$\mathcal{A}^*$ is designed to enable their executions by one or multiple agents and follows the common meaning of an action[16]. For example, the action instance *peek*$\langle A \rangle$ can be executed only by agent A. On the other hand, the notion of an update model is designed for describing the state changes and does not include the information about the actors of the update model, i.e., the agents who will execute the actions specified by the model. It is therefore not always possible to identify the agents who would execute an update model from its description. For example, by simply looking at Figure 5 or examining the definition of the corresponding update model, we cannot distinguish whether the update model is about the instance $raising\_hand(A)$ or $raising\_hand(B)$.

Hence, our representation of actions where perspective fluents are part of the state (and not part of the action) is more appropriate than the representation of actions in the initial formulations of DEL van Ditmarsch et al. (2007). We note that in the planning using DEL setting, relation between agents and action models is introduced in (Engesser et al., 2017; Löwe et al., 2011) to address this issue.

Value of Update Models:. Having discussed the differences between m$\mathcal{A}^*$ and update models, we would like to point out that update models present a very good technical tool for the understanding of effects of actions in multi-agent domains. In fact, the transition function Φ for m$\mathcal{A}^*$ action theories can be effectively characterized using update models, as described in Section 4.

5.2. Previous Work by the Authors

Early attempts to adapt action languages to formalize multi-agent domains can be found in (Baral et al., 2010b; Son et al., 2009; Son and Sakama, 2009). In these works, the action languages $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ have been extended to formalize multi-agent domains.

The research in (Son et al., 2009; Son and Sakama, 2009) investigates the use of action languages in multi-agent planning context and focus on the generation of decentralized plans for multiple agents, to either jointly achieve a goal or individual goals.

In (Baral et al., 2010b), we show that several examples found in the literature—created to address certain aspect in multi-agent systems (e.g., (Boella and van der Torre, 2005; Gerbrandy, 2006; van der Hoek et al., 2005; Herzig and Troquard, 2006; Sauro et al., 2006; Spaan et al., 2006))—can be formalized using an extension of the action language $\mathcal{C}$. Yet, most of the extensions considered

[16]For example, the relevant dictionary meaning of "action is (1) something done or performed; act; deed. (2) an act that one consciously wills and that may be characterized by physical or mental activity.

in (Baral et al., 2010b; Son et al., 2009; Son and Sakama, 2009) are inadequate for formalizing multi-agent domains in which reasoning about knowledge of other agents is critical. To address this shortcoming, we developed and investigated several preliminary versions of m$\mathcal{A}^*$ (Baral et al., 2010a; Pontelli et al., 2010; Baral and Gelfond, 2010). We started with an attempt to formulate knowledge of multiple agents in (Baral et al., 2010a); we successively extended this preliminary version of m$\mathcal{A}^*$ with the use of static observability specifications in (Pontelli et al., 2010). The language developed in this paper subsumes that of (Pontelli et al., 2010). In (Baral and Gelfond, 2010), we demonstrated the use of update models to describe the transition function for the action language of (Pontelli et al., 2010).

m$\mathcal{A}^*$ differs from all earlier versions of the language, in that it clearly differentiates the agents who execute the action from the agents who would observe (or partially observe) the effects of the actions, or are oblivious of the action execution. This is important, since an agent might execute an action without observing its effects. For example, a blind agent would not be able to observe the effects of his action of operating a light switch; an agent firing a gun in a pitch dark night would not be able to observe whether or not he hits the target.

Another, much more important, difference between m$\mathcal{A}^*$ and earlier versions of the language lies in the definition of the function Φ_D. In earlier versions, sensing (or announcement) actions *do not* help the agents in correcting their beliefs. This can be seen in Figure 13. In this example A has the false belief about f (f is true in s_0, the real state of the world, and false in s_1). A executes the action that senses f. The top part shows how earlier versions of m$\mathcal{A}^*$ treat this sensing action occurrence whose update template is in the middle of the figure. The result is the state shown on the right with four disconnected worlds $z_0 = (s_0, \sigma)$, $z_1 = (s_0, \epsilon)$, $z_2 = (s_1, \tau)$, and $z_3 = (s_1, \epsilon)$ with the interpretation of z_j identical to that of s_i where $z_j = (s_i, \_)$. As we can see, A becomes ignorant about everything in this state. The bottom part shows how m$\mathcal{A}^*$ deals with such situation: first, it corrects the beliefs of A and then applies the update. This results in the state on the right (bottom) in which A knows that f is true, which corresponds to the intuitive result of sensing.

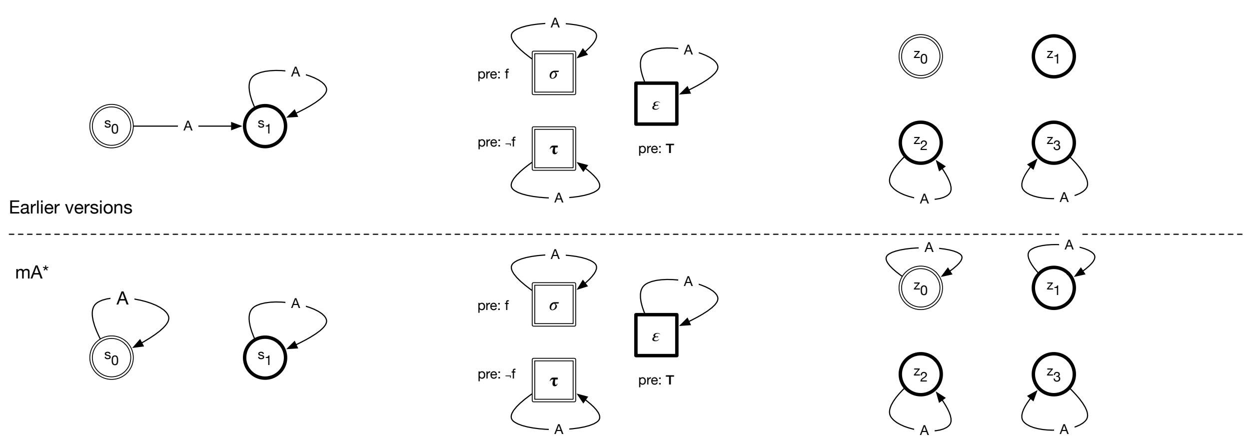

Figure 13: Sensing f helps A to correct her false beliefs in m$\mathcal{A}^*$

5.3. Other Languages

A generalized version of STRIPS, called MA-STRIPS, has been proposed for studying multi-agent classical planning model (Brafman and Domshlak, 2008). In this model, each agent possess a set of actions that they can execute. Distinctions between public and private fluents are considered, which allow for the definition of internal and public actions. This extension is closely related to our earlier extensions of action languages (e.g., (Baral et al., 2010b; Son et al., 2009)). Therefore, the key difference between m$\mathcal{A}^*$ and MA-STRIPS lies in the focus on reasoning about beliefs of agents about the world and about the beliefs of other agents in m$\mathcal{A}^*$ that is not considered in MA-STRIPS. On the other hand, the focus in (Brafman and Domshlak, 2008) is to develop plans for multiple agents to coordinate in achieving a give state of the world, which is not our focus in the development of m$\mathcal{A}^*$.

GDL-III, introduced in (Thielscher, 2017), as an epistemic game specification language, could potentially be used as a specification language for epistemic planning. Syntactically, GDL-III includes specification for two predicates for reasoning about knowledge of each agent *knows(r,p)* and common knowledge *knows(p)* among all agents as well as sensing actions (e.g., the action *peek* in Example 1) and announcement actions (e.g., the action *shout_tail* in Example 1). The effects of actions include *observations* that allow for the agents to update their knowledge. The semantics of a game specification in GDL-III is defined over knowledge states and sequences of actions, each knowledge state is a pair of a set of *true*-atoms representing the true state of the world and a collection of *knows*-atoms representing the knowledge of the agents. It is mentioned in (Thielscher, 2017) that this definition is well-defined for acyclic game descriptions. We observe that Engesser et al. (2018) proved that GDL-III is equivalent to an extension of DEL which allows for a succinct representation of events in update models and conditional effects.

Comparing to GDL-III, m$\mathcal{A}^*$ differs in the following ways. First, a GDL-III game initial state contains only fluents about the state of the world, i.e., it only consider initial state representing by a set of statements of the form (1) in which φ is a fluent. m$\mathcal{A}^*$ does allow arbitrary formulae. Finitary **S5**-theories would subsume the set of initial states permissible in GDL-III. Second, nested knowledge could be specified in GDL-III but at the cost of extra variables. Furthermore, observation tokens of GDL-III are about the real state of the world and thus it will also require some extra variables (e.g, defined predicate for *kwhether*) to specify something likes *B knows that A looks into the box and knows which side of the coin is up*. The third and perhaps most significant difference between GDL-III and m$\mathcal{A}^*$ lies in that in GDL-III, agents receive their individual perceptions after each round and these perceptions are true information about the real state of the world while agents' perceptions are dictated by the observability statements in m$\mathcal{A}^*$. As such, an agent might not know the truth value of a proposition but never has false beliefs about the world in GDL-III while it is possible in m$\mathcal{A}^*$. We believe that this stems from the fact that GDL-III focuses on knowledge of agents and m$\mathcal{A}^*$ deals with beliefs of agents.

5.4. m$\mathcal{A}^*$ and Action Languages for Single-Agent Domains

m$\mathcal{A}^*$ is a high-level action language for multi-agent domains. It is therefore instructive to discuss the connection between m$\mathcal{A}^*$ and action languages for single-agent domains. First, let us observe that m$\mathcal{A}^*$ has the following multi-agent domain specific features:

- it includes announcement actions; and
- it includes specification of the agents' observability of action occurrences.

As it turns out, if we remove all features that are specific to multi-agent domains from m$\mathcal{A}^*$, and consider the **S5**-entailment as its semantics, then the language is equivalent to the language $\mathcal{A}_K$ by Son and Baral (2001). Formally, let us consider a m$\mathcal{A}^*$ definite action theory (I, D) over the signature $\langle \mathcal{AG}, \mathcal{F}, \mathcal{A} \rangle$ such that $|\mathcal{AG}| = 1$ and D does not contain statements of the form (8) (announcement actions) and statements of the form (9)-(10). Let us define

$$I_{\mathcal{A}_K} = \{\varphi \mid \varphi \text{ appears in a statement of the form (1) or (2) in } I\}.$$

Then, the following holds

$$(I, D) \models_{\mathbf{S5}} \varphi \textbf{ after } \delta \quad \text{iff} \quad (I_{\mathcal{A}_K}, D) \models_{\mathcal{A}_K} \varphi \textbf{ after } \delta.$$

This shows that m$\mathcal{A}^*$ is indeed a generalization of action languages for single-agent domains to multi-agent domains. This also supports the claim that other elements that have been considered in action languages of single-agent domains, such as static causal laws, non-deterministic actions, or parallel actions could potentially be generalized to m$\mathcal{A}^*$.

6. Conclusions and Future Works

In this paper, we developed an action language for representing and reasoning about effects of actions in multi-agent domains. The language considers world-altering actions, sensing actions, and announcement actions. It also allows the dynamic specification of agents' observability with respect to action occurrences, enabling varying degrees of visibility of action occurrences and action effects. The semantics of the language relies on the notion of states (pointed Kripke structures), used as representations of the states of the world and states of agents' knowledge and beliefs; the semantics builds on a transition function, which maps pairs of states and actions to sets of states and employs the well-known notion of update models as the underlying machineries.

We discussed several properties of the transition function and identified a class of theories (*definite action theories*) whose set of initial **S5**-states is finite, thus allowing for the development of algorithms for the **S5**-entailment relation that is critical in applications such as planning, temporal reasoning, and diagnosis.

The development of m$\mathcal{A}^*$ is a first step towards the goal of developing scalable and efficient automated reasoning and planning systems in multi-agent domains. Important next steps include extending the language to deal with lying and/or misleading actions, refining the distinction between knowledge and beliefs of the agents, and specifying more general models of agents' observability, to capture some of the capabilities of update models that are missing from m$\mathcal{A}^*$.

Acknowledgments

The last two authors have been partially supported by NSF grants 1914635, 1757207, and 1812628.

References

Agotnes, T., Lakemeyer, G., Löwe, B., Nebel, B., 2014. Planning with epistemic goals. Tech. rep.

Allen, M., Zilberstein, S., 2009. Complexity of decentralized control: Special cases. In: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada. Curran Associates, Inc., pp. 19–27.

Aucher, G., Bolander, T., 2013. Undecidability in epistemic planning. In: Rossi, F. (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. IJCAI/AAAI.

Baltag, A., Moss, L., 2004. Logics for epistemic programs. Synthese.

Baral, C., Bolander, T., McIlraith, S., Ditmarsch, H. V., 2017. Epistemic planning. Tech. rep. URL `http://www.dagstuhl.de/17231`

Baral, C., Gelfond, G., 2010. On representing actions in multi-agent domains. In: Proceedings of the Symposium on Constructive Mathematics.

Baral, C., Gelfond, G., Pontelli, E., Son, T., 2010a. Modeling multi-agent scenarios involving agents knowledge about other's knowledge using ASP. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 259–266.

Baral, C., Gelfond, G., Pontelli, E., Son, T. C., 2012. An action language for reasoning about beliefs in multi-agent domains. In: Proceedings of NMR.

Baral, C., Gelfond, G., Pontelli, E., Son, T. C., 2013. 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 deer Torre, L., Woltran, S. (Eds.), Proceedings of the 14th International Workshop, Computational Logic in Multi-Agent Systems, CLIMA VIX, Coruna, Spain, September 16-18, 2013. Vol. 8143 of Lecture Notes in Computer Science. Springer, pp. 290–306.

Baral, C., Son, T. C., Pontelli, E., 2010b. Reasoning about multi-agent domains using action language $\mathcal{C}$: A preliminary study. In: Dix, J., Fisher, M., Novák, P. (Eds.), Computational Logic in Multi-Agent Systems - 10th International Workshop, CLIMA X, Hamburg, Germany, September 9-10, 2009, Revised Selected and Invited Papers. Vol. 6214 of Lecture Notes in Computer Science. Springer, pp. 46–63.

Bernstein, D. S., Givan, R., Immerman, N., Zilberstein, S., 2002. The complexity of decentralized control of markov decision processes. Math. Oper. Res. 27 (4), 819–840.

Boella, G., van der Torre, L. W. N., 2005. Enforceable social laws. In: Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., Wooldridge, M. (Eds.), 4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands. ACM, pp. 682–689.

Bolander, T., 2018. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In: Jaakko Hintikka on Knowledge and Game-Theoretical Semantics. Vol. 12 of Outstanding Contributions to Logic. Springer, Cham., pp. 207–236.

Bolander, T., Andersen, M., 2011. Epistemic Planning for Single and Multi-Agent Systems. Journal of Applied Non-Classical Logics 21 (1).

Bolander, T., Jensen, M. H., Schwarzentruber, F., 2015. Complexity results in epistemic planning. In: Yang, Q., Wooldridge, M. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. AAAI Press, pp. 2791–2797.

Brafman, R. I., Domshlak, C., 2008. From one to many: Planning for loosely coupled multi-agent systems. In: Rintanen, J., Nebel, B., Beck, J. C., Hansen, E. A. (Eds.), Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, September 14-18, 2008. AAAI, pp. 28–35.

Castellini, C., Giunchiglia, E., Tacchella, A., 2001. Improvements to sat-based conformant planning. In: Proceedings of 6th European Conference on Planning (ECP-01).

de Weerdt, M., Bos, A., Tonino, H., Witteveen, C., 2003. A resource logic for multi-agent plan merging. Ann. Math. Artif. Intell. 37 (1-2), 93–130.

de Weerdt, M., Clement, B., 2009. Introduction to planning in multiagent systems. Multiagent Grid Systems 5, 345–355.

del Val A., Shoham, Y., 1994. A unified view of belief revision and update. Journal of Logic and Computation, Special issue on Actions and processes, M. Georgeff (ed.).

Durfee, E., 1999. Distributed Problem Solving and Planning. In: Muliagent Systems (A Modern Approach to Distributed Artificial Intelligence). MIT Press, pp. 121–164.

Engesser, T., Bolander, T., Mattmüller, R., Nebel, B., 2017. Cooperative epistemic multi-agent planning for implicit coordination. In: Ghosh, S., Ramanujam, R. (Eds.), Proceedings of the Ninth Workshop on Methods for Modalities, M4M@ICLA 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017. Vol. 243 of EPTCS. pp. 75–90.

Engesser, T., Mattmüller, R., Nebel, B., Thielscher, M., 2018. Game description language and dynamic epistemic logic compared. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. pp. 1795–1802.

Fabiano, F., Burigana, A., Dovier, A., Pontelli, E., 2020. EFP 2.0: A Multi-Agent Epistemic Solver with Multiple E-State Representations. In: International Conference on Automated Planning and Scheduling (ICAPS). AAAI Press.

Fagin, R., Halpern, J., Moses, Y., Vardi, M., 1995. Reasoning about Knowledge. MIT press.

Fikes, R., Nilson, N., 1971. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence 2 (3–4), 189–208.

Friedman, N., Halpern, J., 1999. Modeling Belief in Dynamic Systems: Revision and Update. Journal of Artificial Intelligence Research 10, 117–167.

Gelfond, M., Lifschitz, V., 1993. Representing actions and change by logic programs. Journal of Logic Programming 17 (2,3,4), 301–323.

Gelfond, M., Lifschitz, V., 1998. Action Languages. Electronic Transactions on Artificial Intelligence 3 (6).

Gerbrandy, J., 2006. Logics of propositional control. In: Nakashima, H., Wellman, M. P., Weiss, G., Stone, P. (Eds.), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006. ACM, pp. 193–200.

Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., Wilkins, D., 1998. PDDL — the Planning Domain Definition Language. Version 1.2. Tech. Rep. CVC TR98003/DCS TR1165, Yale Center for Comp, Vis and Ctrl.

Giunchiglia, E., Kartha, G., Lifschitz, V., 1997. Representing action: indeterminacy and ramifications. Artificial Intelligence 95, 409–443.

Goldman, C. V., Zilberstein, S., 2004. Decentralized control of cooperative systems: Categorization and complexity analysis. Journal of Artificial Intelligence Resesearch (JAIR) 22, 143–174.

Guestrin, C., Koller, D., Parr, R., 2001. Multiagent planning with factored mdps. In: Dietterich, T. G., Becker, S., Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]. MIT Press, pp. 1523–1530.

Herzig, A., Lang, J., Marquis, P., 2005. Action Progression and Revision in Multiagent Belief Structures. In: Sixth Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC).

Herzig, A., Troquard, N., 2006. Knowing how to play: uniform choices in logics of agency. In: Nakashima, H., Wellman, M. P., Weiss, G., Stone, P. (Eds.), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006. pp. 209–216.

Huang, X., Fang, B., Wan, H., Liu, Y., 2017. A general multi-agent epistemic planner based on higher-order belief change. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).

Katsuno, H., Mendelzon, A., 1992. On the difference between updating a knowledge base and revising it. In: Proceedings of KR 92. pp. 387–394.

Kominis, F., Geffner, H., 2015. Beliefs in multiagent planning: From one agent to many. In: Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015. pp. 147–155.

Le, T., Fabiano, F., Son, T. C., Pontelli, E., 2018. EFP and PG-EFP: Epistemic Forward Search Planners in Multi-Agent Domains. In: International Conference on Automated Planning and Scheduling (ICAPS). AAAI Press.

Levesque, H., Reiter, R., Lesperance, Y., Lin, F., Scherl, R., April-June 1997. GOLOG: A logic programming language for dynamic domains. Journal of Logic Programming 31 (1-3), 59–84.

Lifschitz, V., 1987. On the semantics of STRIPS. In: Georgeff, M., Lansky, A. (Eds.), Reasoning about Actions and Plans. Morgan Kaufmann, San Mateo, CA., pp. 1–9.

Liu, Q., Liu, Y., 2018. Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, july 13-19, 2018, stockholm, sweden. In: Lang, J. (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. ijcai.org, pp. 1912–1920.

Löwe, B., Pacuit, E., Witzel, A., 2011. Del planning and some tractable cases. In: van Ditmarsch, H., Lang, J., Ju, S. (Eds.), Logic, Rationality, and Interaction. Vol. 6953 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 179–192.

McCarthy, J., 1959. Programs with common sense. In: Proceedings of the Teddington Conference on the Mechanization of Thought Processes. Her Majesty's Stationery Office, London, pp. 75–91.

Muise, C., Belle, V., Felli, P., McIlraith, S., Miller, T., Pearce, A. R., Sonenberg, L., 2015. Planning over multi-agent epistemic states: A classical planning approach. In: Proceedings of AAAI.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D. V., Marsella, S., 2003. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In: Gottlob, G., Walsh, T. (Eds.), IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. Morgan Kaufmann, pp. 705–711.

Pednault, E., 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In: Brachman, R., Levesque, H., Reiter, R. (Eds.), Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, pp. 324–332.

Peshkin, L., Savova, V., 2002. Reinforcement learning for adaptive routing. In: Proceedings of the Int. Joint Conf. on Neural Networks.

Pontelli, E., Son, T., Baral, C., Gelfond, G., 2010. Logic programming for finding models in the logics of knowledge and its applications: A case study. Theory and Practice of Logic Programming 10 (4-6), 675–690.

Sauro, L., Gerbrandy, J., van der Hoek, W., Wooldridge, M., 2006. Reasoning about action and cooperation. In: AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. ACM, New York, NY, USA, pp. 185–192.

Shanahan, M., 1999. The ramification problem in the event calculus. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages. pp. 140–146.

Son, T., Pontelli, E., Sakama, C., 2009. Logic Programming for Multiagent Planning with Negotiation. In: Hill, P. M., Warren, D. S. (Eds.), Logic Programming, 25th International Conference, ICLP 2009, Pasadena, CA, USA, July 14-17, 2009. Proceedings. Vol. 5649 of Lecture Notes in Computer Science. Springer, pp. 99–114.

Son, T., Sakama, C., 2009. Reasoning and planning with cooperative actions for multiagents using answer set programming. In: Baldoni, M., Bentahar, J., Lloyd, J., van Riemsdijk, B. (Eds.), Declarative Agent Languages and Technologies VI, 6th International Workshop, DALT 2009, Budapest, Hungary, 2009, Revised Selected and Invited Papers. Vol. 5948. Springer, pp. 208–227.

Son, T. C., Baral, C., January 2001. Formalizing sensing actions - a transition function based approach. Artificial Intelligence 125 (1-2), 19–91.

Son, T. C., Pontelli, E., Baral, C., Gelfond, G., 2014. Finitary s5-theories. In: Fermé, E., Leite, J. (Eds.), Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings. Vol. 8761 of Lecture Notes in Computer Science. Springer, pp. 239–252.

Son, T. C., Pontelli, E., Baral, C., Gelfond, G., 2015. Exploring the KD45n Property of a Kripke Model after the Execution of an Action Sequence. In: AAAI.

Son, T. C., Tu, P. H., Gelfond, M., Morales, R., 2005. Conformant Planning for Domains with Constraints — A New Approach. In: Proceedings of the Twentieth National Conference on Artificial Intelligence. pp. 1211–1216.

Spaan, M. T. J., Gordon, G. J., Vlassis, N. A., 2006. Decentralized planning under uncertainty for teams of communicating agents. In: Nakashima, H., Wellman, M. P., Weiss, G., Stone, P. (Eds.), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006. pp. 249–256.

Thiebaux, S., Hoffmann, J., Nebel, B., 2003. In Defense of PDDL Axioms. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03).

Thielscher, M., 2017. Gdl-iii: A description language for epistemic general game playing. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).

Tu, P., Son, T., Gelfond, M., Morales, R., January 2011. Approximation of action theories and its application to conformant planning. Artificial Intelligence Journal 175 (1), 79–119.

van Benthem, J., 2007. Dynamic logic of belief revision. Journal of Applied Non-Classical Logics 17(2), 129–155.

van Benthem, J., van Eijck, J., Kooi, B. P., 2006. Logics of communication and change. Inf. Comput. 204 (11), 1620–1662.

van der Hoek, W., Jamroga, W., Wooldridge, M., 2005. A logic for strategic reasoning. In: Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., Wooldridge, M. (Eds.), 4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands. ACM, pp. 157–164.

van Ditmarsch, H., van der Hoek, W., Kooi, B., 2007. Dynamic Epistemic Logic. Springer.

van Eijck, J., 2004. Dynamic epistemic modelling. Tech. rep.

van Eijck, J., 2017. Public announcements and public lies. Tech. rep., Lying Workshop.

Wan, H., Yang, R., Fang, L., Liu, Y., Xu, H., 2015. A complete epistemic planner without the epistemic closed world assumption. In: Yang, Q., Wooldridge, M. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. AAAI Press, pp. 3257–3263.
URL `http://ijcai.org/proceedings/2015`

Appendix A: Proofs of Theorems

Recall that the following notations are used in the presentation of the theorems.

- D denotes a consistent m$\mathcal{A}^*$ domain;
- (M, s) denotes a state; and
- a is an action instance, whose precondition is given by the statement

  $$\textbf{executable } \mathsf{a} \textbf{ if } \psi$$

  in D, and a is executable in (M, s).
- $\rho = (F, P, O)$ is the frame of reference of the execution of a in (M, s) where $F = F_D(\mathsf{a}, M, s)$, $P = P_D(\mathsf{a}, M, s)$, and $O = O_D(\mathsf{a}, M, s)$.

Theorem 2. *Assume that* a *is an ontic-action instance. It holds that:*

1. *for every agent $x \in F_D(\mathsf{a}, M, s)$ and $[\mathsf{a}$ **causes** ℓ **if** $\varphi]$ belongs to D, if $(M, s) \models \mathbf{B}_x\varphi$ then $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x\ell$;*
2. *for every agent $y \in O_D(\mathsf{a}, M, s)$ and a belief formula η, $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_y\eta$ iff $(M, s) \models \mathbf{B}_y\eta$;*
3. *for every pair of agents $x \in F_D(\mathsf{a}, M, s)$ and $y \in O_D(\mathsf{a}, M, s)$ and a belief formula η, if $(M, s) \models \mathbf{B}_x\mathbf{B}_y\eta$ then $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x\mathbf{B}_y\eta$.*

Proof. Since a is executable in (M, s), we have that $(M, s) \models \psi$. This means that

$$\Phi_D(\mathsf{a}, (M, s)) = (M, s)\otimes(\omega(\mathsf{a}, \rho), \{\sigma\})$$

where $(\omega(\mathsf{a}, \rho), \{\sigma\})$ is given in Definition 13. Assume that $(M', s') \in \Phi_D(\mathsf{a}, (M, s))$. By Definition 6, we have $s' = (s, \sigma)$. Assume that the fluent in ℓ is p, i.e., $\ell = p$ or $\ell = \neg p$.

1. Let $\Psi^+(p, \mathsf{a}) = \bigvee\{\varphi \mid [\mathsf{a}$ **causes** p **if** $\varphi] \in D\}$ and $\Psi^-(p, \mathsf{a}) = \bigvee\{\varphi \mid [\mathsf{a}$ **causes** $\neg p$ **if** $\varphi] \in D\}$ and $\theta = \Psi^+(p, \mathsf{a}) \vee (p \wedge \neg\Psi^-(p, \mathsf{a}))$. By Definition 13, $p \rightarrow \theta \in sub(\sigma)$. Furthermore, for every $u' \in M'[S]$ such that $(s', u') \in M'[x]$, it holds that $u' = (u, \sigma)$ for some $u \in M[S]$, $(M, u) \models \psi$, and $(s, u) \in M[x]$. Because $(M, s) \models \mathbf{B}_x\varphi$, we have that $(M, u) \models \varphi$. Consider two cases:

   - $\ell = p$. Then, $(M, u) \models \Psi^+(p, \mathsf{a})$, and hence, $(M, u) \models \theta$. So, $M'[\pi]((u, \sigma)) \models p$.
   - $\ell = \neg p$. Then, because $(M, u) \models \varphi$, the consistency of D implies that $(M, u) \not\models \theta$. Therefore, $M'[\pi]((u, \sigma)) \not\models p$, i.e., $M'[\pi]((u, \sigma)) \models \neg p$.

   Both cases imply that $M'[\pi]((u, \sigma)) \models \ell$. This holds for every $u' \in M'[S]$ such that $(s', u') \in M'[x]$, which implies $(M', s') \models \mathbf{B}_x\ell$.
2. By the construction of M', we have the following observations:

   - for every $u \in M[S]$ iff $(u, \epsilon) \in M'[S]$;
   - for every $z \in \mathcal{AG}$, $(u, v) \in M[z]$ iff $((u, \epsilon), (v, \epsilon)) \in M'[z]$; and
   - for every $u \in M[S]$ and $p \in \mathcal{F}$, $M'[\pi]((u, \epsilon)) \models p$ iff $(M', (u, \epsilon)) \models p$ because $sub(\epsilon) = \emptyset$.

   These observations allow us to conclude that for every formula η, $(M, u) \models \eta$ iff $(M', (u, \epsilon)) \models \eta$.
   Now, let us get back to the second item of the theorem. Consider $u' \in M'[S]$ such that $(s', u') \in M'[y]$. This holds iff there exists $u \in M[S]$, $(s, u) \in M[y]$, and $u' = (u, \epsilon)$.
   Since $(M, u) \models \eta$ iff $(M', (u, \epsilon)) \models \eta$ and this holds for every $u' \in M'[S]$ such that $(s', u') \in M'[y]$, we have that $(M, s) \models \mathbf{B}_y\eta$ iff $(M', s') \models \mathbf{B}_y\eta$.
3. Consider $u', v' \in M'[S]$ such that $(s', u') \in M'[x]$ and $(u', v') \in M'[y]$. This holds if there exist $u, v \in M[S]$, $(s, u) \in M[x]$ and $(u, v) \in M[y]$ such that $u' = (u, \sigma)$ and $v' = (v, \epsilon)$.
   Assume that $(M, s) \models \mathbf{B}_x\mathbf{B}_y\eta$. This implies that $(M, v) \models \eta$. The second item shows that $(M', (v, \epsilon)) \models \eta$, i.e., which implies $(M', s') \models \mathbf{B}_x\mathbf{B}_y\eta$. Since this holds for every $u', v' \in M'[S]$ such that $(s', u') \in M'[x]$ and $(u', v') \in M'[y]$, we have $(M', s') \models \mathbf{B}_x\mathbf{B}_y\ell$.

Since (M', s') is an arbitrary element in $\Phi_D(\mathsf{a}, (M, s))$, the theorem holds. $\square$

Theorem 3. *Let us assume that* a *is a sensing action instance and* D *contains the statement* a **determines** φ. *It holds that:*

1. *if* $(M, s) \models \varphi$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi$*;*
2. *if* $(M, s) \models \neg\varphi$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi$*;*
3. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{P_D(\mathsf{a},M,s)}(\mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi)$*;*
4. $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{C}_{F_D(\mathsf{a},M,s)}(\mathbf{C}_{P_D(\mathsf{a},M,s)}(\mathbf{C}_{F_D(\mathsf{a},M,s)}\varphi \vee \mathbf{C}_{F_D(\mathsf{a},M,s)}\neg\varphi))$*;*
5. *for every agent* $y \in O_D(\mathsf{a}, M, s)$ *and formula* η, $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_y\eta$ *iff* $(M, s) \models \mathbf{B}_y\eta$*;*
6. *for every pair of agents* $x \in F_D(\mathsf{a}, M, s)$ *and* $y \in O_D(\mathsf{a}, M, s)$ *and a formula* η *if* $(M, s) \models \mathbf{B}_x\mathbf{B}_y\eta$ *then* $\Phi_D(\mathsf{a}, (M, s)) \models \mathbf{B}_x\mathbf{B}_y\eta$*.*

Proof. We will prove the theorem for the case $(M, s) \models \varphi$. The proof of the theorem when $(M, s) \models \neg\varphi$ is similar and is omitted here. Since a is executable in (M, s), we have that $(M, s) \models \psi$. This means that

$$\Phi_D(\mathsf{a}, (M, s)) = M_1[F_D(\mathsf{a}, M_1, s), \varphi] \otimes (\omega(\mathsf{a}, \rho), \{\sigma, \tau\})$$

where $M_1 = M[P_D(\mathsf{a}, M, s), \psi]$ and $(\omega(\mathsf{a}, \rho), \{\sigma, \tau\})$ is given in Definition 14. Let us denote $M_1[F_D(\mathsf{a}, M_1, s), \varphi]$ with M^*. Observe that by the definition of M^*, for each $x \in F$ there exists some $u \in M^*[x]$ such that $(M^*, u) \models \varphi$ and for each $x \in P$ and there exists some $u \in M^*[x]$ such that $(M^*, u) \models \psi$.

We need to prove Items 1, 3, 4, 5, and 6.

Assume that $(M', s') \in \Phi_D(\mathsf{a}, (M, s))$. By Definition 6, we have $s' = (s, \sigma)$.

1. *Proof of the first item of the theorem.*
   To prove $(M', s') \models \mathbf{C}_F\varphi$, we need to show that
   $$(M', s') \models \mathbf{B}_{i_1}\mathbf{B}_{i_2}\ldots\mathbf{B}_{i_k}\varphi$$
   for any sequence $i_1, \ldots, i_k$ of agents in F, i.e., $i_j \in F$ for $j = 1, \ldots, k$.
   Let $u'_1, \ldots, u'_{k+1} \in M'[S]$ such that $(s', u'_1) \in M'[i_1]$, $(u'_j, u'_{j+1}) \in M'[i_{j+1}]$ for $j = 1, \ldots, k$. Observe that for any $x \in F$ and $u' \in M'[S]$ such that $(s', u') \in M'[x]$, it holds that $u' = (u, \sigma)$ for some $u \in M^*[S]$, $(M^*, u) \models \psi \wedge \varphi$, and $(s, u) \in M^*[x]$.
   This observation allows us to conclude that, for $u'_1, \ldots, u'_{k+1}$, there exist $u_1, \ldots, u_{k+1} \in M^*[S]$ such that $(s, u_1) \in M^*[i_1]$, $(u_j, u_{j+1}) \in M^*[i_{j+1}]$ for $j = 1, \ldots, k$, and for every $j = 1, \ldots, k+1$, $u'_j = (u_j, \sigma)$ and $u_i \models \psi \wedge \varphi$. It is easy to see that this leads to $(M', s') \models \mathbf{C}_F\varphi$.
2. *Proof of the third item of the theorem when* $(M, s) \models \varphi$.
   To prove $(M', s') \models \mathbf{C}_P(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi)$, we need to show that
   $$(M', s') \models \mathbf{B}_{i_1}\mathbf{B}_{i_2}\ldots\mathbf{B}_{i_k}(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi)$$
   for any sequence $i_1, \ldots, i_k$ of agents in P, i.e., $i_j \in P$ for $j = 1, \ldots, k$.
   Let $u'_1, \ldots, u'_{k+1} \in M'[S]$ such that $(s', u'_1) \in M'[i_1]$, $(u'_j, u'_{j+1}) \in M'[i_{j+1}]$ for $j = 1, \ldots, k$. Similar to the argument in the previous item and Definitions 6 and 14 allows us to conclude that, for $u'_1, \ldots, u'_{k+1}$, there exist $u_1, \ldots, u_{k+1} \in M^*[S]$ such that $(s, u_1) \in M^*[i_1]$, $(u_j, u_{j+1}) \in M^*[i_{j+1}]$ for $j = 1, \ldots, k$, and for every $j = 1, \ldots, k+1$, either (*a*) $u'_j = (u_j, \sigma)$ and $u_i \models \psi \wedge \varphi$ or (*b*) $u'_j = (u_j, \tau)$ and $u_i \models \psi \wedge \neg\varphi$. This leads to two cases:

(a) $u'_{k+1} = (u_{k+1}, \sigma)$ and $u_{k+1} \models \psi \wedge \varphi$. Then, similar to the proof in Item 1, we can show that $(M', u'_{k+1}) \models C_F\varphi$.

(b) $u'_{k+1} = (u_{k+1}, \sigma)$ and $u_{k+1} \models \psi \wedge \neg\varphi$. Again, similar to the proof in Item 1, we can show that $(M', u'_{k+1}) \models C_F\neg\varphi$.

The two cases imply that $(M', s') \models \mathbf{C}_P(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi)$.

3. *Proof of the fourth item of the theorem when* $(M, s) \models \varphi$.
To prove $(M', s') \models \mathbf{C}_F(\mathbf{C}_P(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi))$, we need to show that

$$(M', s') \models \mathbf{B}_{i_1}\mathbf{B}_{i_2}\ldots\mathbf{B}_{i_k}(\mathbf{C}_P(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi))$$

for any sequence $i_1, \ldots, i_k$ of agents in F, i.e., $i_j \in F$ for $j = 1, \ldots, k$. This holds because we can show that for each $u' = (u, \sigma)$ such that $u \in M^*[S]$ and $(M^*, u) \models \psi \wedge \varphi$, $(M', u') \models \mathbf{C}_P(\mathbf{C}_F\varphi \vee \mathbf{C}_F\neg\varphi)$. The arguments for this conclusion are similar to the arguments used in the proof in Item 2.

4. The proof of the fifth and sixth items of this theorem is similar to the proof of the second and third item of Theorem 2, respectively.

Since (M', s') is an arbitrary element in $\Phi_D(\mathsf{a}, (M, s))$, the theorem holds. □