

Representing and Reasoning with Event Models for Epistemic Planning

David Rajaratnam and Michael Thielscher

UNSW Sydney, Australia
{david.rajaratnam,mit}@unsw.edu.au

Abstract

The standard representation formalism for multi-agent epistemic planning has one central disadvantage: When you use event models in dynamic epistemic logic (DEL) to describe the action of one agent, the model must specify not only the actual change and the change of that agent's knowledge. Also required is the epistemic change of any agents that may be observing the first agent performing the action, plus the epistemic change for any further agents that failed to observe that anything had taken place. To overcome the gap between this complex DEL notion of events and a more commonsense notion of actions, we propose a simple high-level action description language for multi-agent epistemic planning domains with just one type of effect laws: a **causes** x **if** y . Effect x can either be a physical effect, or an *observation* from an independent set that is specific to individual agents. We formally prove that any DEL event model can be described in this way. We show how this language provides a framework for expressing a variety of executability and action models; such as describing actions that are both ontic and epistemic, partially observable, or nondeterministic. We further combine our representation of event models with a description language for finitary initial epistemic theories, and we show how this allows us to reason about the effects of a sequence of actions in a multi-agent epistemic domain by updating a single multi-pointed epistemic model.

1 Introduction

Multi-agent epistemic planning has garnered a lot of research attention recently for the purpose of controlling collaborative, or competitive, agents and robots (Baral et al. 2017). Dynamic epistemic logic (DEL) along with event models (Bolander and Andersen 2011) is considered a quasi standard as the most expressive general formalism for modelling these domains. However, despite this expressive power, a major disadvantage of DEL as a domain description language is the direct use of Kripke-style event models to specify actions and their effects (Baral and Gelfond 2011; Baral et al. 2017). In particular, event models encode both the effects as well as the observability of actions. In philosophical terms, this is known as a failure to separate action *types* from action *tokens* (Baral et al. 2017). To understand why this is problematic, it is easiest to consider an example.

In order to model the action of an agent looking into a box, the event model must specify the change of that agent's

knowledge as it perceives the content of the box. But the model must also encode the epistemic change of any agents that may be observing the first agent looking into the box. On top of this, it must also encode the preservation of knowledge for any further agents that failed to observe that anything had taken place. All these different forms of observability are encoded as distinct events within the event model. This makes for a conceptually and practically complex relationship between the DEL notion of an action, as a collection of events, and a more commonsense notion of an action.

For this reason a variety of other, compact and modular representation languages have been developed, e.g. as front-ends to automated reasoners for dynamic epistemic domains (Baral et al. 2015), planners (Le et al. 2018), or game-playing systems (Thielscher 2017). These use a variety of special-purpose language constructs, and yet most of them cannot express the full range of executability and action models of general event models.

In this paper, we will show that a surprisingly lean language construct suffices for giving compact and high-level descriptions of unrestricted epistemic event models. The construct is a simple effect law, a **causes** x **if** y , but underpinned with a concept originating from Hidden Markov Models, where an independent set of *observations* is used to provide an observer with information about the hidden system state (Baum and Petrie 1966). A similar idea was used for the epistemic game description language (GDL) and shown, under certain assumptions, to be intertranslatable with DEL specifications (Engesser et al. 2018); however, with its special-purpose design and use of the syntax and semantics of logic programming, GDL lacks the simplicity and generality of a single language construct.

We will demonstrate that a variety of aspects can be expressed in our new language in a compact and modular way; such as describing actions that are both ontic (i.e. change the world) and at the same time epistemic (i.e. change the knowledge of the agents), partially observable, or nondeterministic; and we prove that any DEL epistemic event model can be described in the language. We further combine our representation of event models with a description language for so-called *finitary* initial epistemic theories (Son et al. 2014), and we show how this allows us to reason about the effects of a sequence of actions in a multi-agent epistemic domain by updating a single multi-pointed epistemic model.

The rest of the paper is organised as follows. In the next section, we briefly recapitulate the necessary basic definitions of DEL and event models. Section 3 contains the formal definition of the syntax of our new language DER (for: dynamic epistemic representation) along with an example known from the literature (Baral et al. 2015) that illustrates a variety of features of epistemic domains and how they can be expressed with the very lean syntax of DER. In the two sections that follow we define the semantics of our language in terms of event models, and we prove that, conversely, any DEL-style event model can be described in DER. In Section 6, we extend our DER to incorporate elements for representing a sub-class of initial (epistemic) states as well as reasoning about the consequences of actions, and we illustrate how combining these two elements with the event model representation allows us to describe DEL reasoning at a purely syntactic level. In Section 7, we show how our language generalises previous epistemic action formalisms, and we conclude in Section 8.

2 Background: Epistemic Planning and DEL

Multi-agent epistemic planning (Bolander and Andersen 2011) uses the language of dynamic epistemic logic (DEL). Let Ag be a set of agents and \mathcal{F} a set of propositions (usually referred to as *fluents* in classical planning), then the epistemic language $\mathcal{L}_{\mathcal{F}, Ag}$ is given by the BNF

$$\varphi ::= f \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi$$

where $f \in \mathcal{F}$ and $i \in Ag$. $K_i\varphi$ means that agent i knows φ , and $C\varphi$ means that φ is common knowledge among all agents. Standard abbreviations $\varphi_1 \vee \varphi_2$, $\varphi_1 \rightarrow \varphi_2$, $\varphi_1 \leftrightarrow \varphi_2$, \top , \perp are also used. We leave details of epistemic models and states to Section 6 as our main focus here is with *event models*, which describe the effects of actions on the world and on the knowledge of the agents.

An event model $\mathcal{E} = \langle E, (Q_i)_{i \in Ag}, pre, eff \rangle$ consists of a finite set of *events* E ; an indistinguishability (equivalence) relation $Q_i \subseteq E \times E$ for each agent $i \in Ag$; a function $pre: E \rightarrow \mathcal{L}_{\mathcal{F}, Ag}$ defining the *preconditions* of events; and a function $eff: E \rightarrow \mathcal{L}_{\mathcal{F}, Ag}$ defining conjunctions of fluent literals (i.e. f or $\neg f$ for $f \in \mathcal{F}$) as the *effects* of events. Readers unfamiliar with the concept of event models may take a look at Figures 1 and 2 in Section 4 for two examples. The operation that determines the result of executing an event model \mathcal{E} is known as *product update*; we again refer to Bolander and Andersen (2011) for details.

It should be noted that the multi-agent epistemic planning literature does extend beyond knowledge to encompass the more general notion of *belief states* (also known as *doxastic states*) (Baltag, Moss, and Solecki 1998; Baral, Gelfond, and Son 2012). However, multi-agent doxastic planning is a significantly more complex challenge. It is typically considered only in a very restricted setting (Muise et al. 2015; Son et al. 2015) or requires the introduction of plausibility and preference relations to deal with problems such as how agents recover from false beliefs (Andersen, Bolander, and Jensen 2015; Baral et al. 2017). Consequently, epistemic planning remains the standard setting and this is also the setting that we consider.

3 A New Representation Language: Syntax

Fluents (i.e., state variables) and actions are the basic constituents of most knowledge representation formalisms for reasoning about actions and planning (Fikes and Nilsson 1971; Gelfond and Lifschitz 1993; Reiter 2001). We enrich this with a fundamental concept adapted from Hidden Markov Models (Baum and Petrie 1966), where measurable “emissions” are used to specify *observations* and how they are triggered by the hidden system state. Crucial for the generality of the theory of Hidden Markov Models is the fact that the state variables and observables are two distinct sets, connected only via rules that govern which and how observations are caused by the internal state of a system. In view of epistemic domains with multiple agents, we further generalise this concept by associating observations with individual agents, thus allowing for different agents perceiving different aspects of a partially observable state.

Definition 1 (Signature). *A signature for a dynamic epistemic domain description consists of finite sets of agents Ag , actions \mathcal{A} , fluents \mathcal{F} , and observations \mathcal{O} , where \mathcal{F} and \mathcal{O} are disjoint sets. Each observation is associated with an agent via an ownership function $\omega: \mathcal{O} \rightarrow Ag$.*

We adopt an example from Baral et al. (2015), who use it to illustrate a variety of aspects of dynamic epistemic domains, including different degrees of observability as well as common knowledge: Three agents, A, B, C , are in a room with a box that can be opened by any agent in possession of a key. The box contains a coin. Agents can peek into an opened box to determine which side of the coin is up; signal or distract another agent from looking at their action; and publicly announce that the coin shows tails.

Example 1 (Baral et al. 2015). *Consider the signature:*

- *agents* $Ag = \{A, B, C\}$
- *actions* $\mathcal{A} = \{open(x), peek(x), signal(x, y), distract(x, y), announceT(x)\}$
- *fluents* $\mathcal{F} = \{has_key(x), looking(x), opened, tail\}$
- *observations* $\mathcal{O} = \{obs(x, tail), obs(x, peek(y))\}$

Where every expression with one or more variables stands for all its ground instances with $x, y \in Ag$. Ownership of observations is determined by the first argument, that is, $\omega(obs(x, tail)) = \omega(obs(x, peek(y))) = x$.

Observation $obs(x, tail)$ informs an agent x of whether tails is up, and the role of $obs(x, peek(y))$ will be to indicate to an agent x that agent y has just peeked into the box.

The general concept of a set of observations, specific to individual agents and separate from the state variables (fluents) makes it possible to define a very simple yet powerful action description language that requires effect laws of just one type: a **causes** x **if** y , where x is either an (ontic) effect, i.e. a fluent becoming true or false, or an observation that happens as a result of action a under condition y . This uses the same syntax as in the original, basic action description language (Gelfond and Lifschitz 1993) but extended to observations.

Definition 2 (Action laws). *Consider a signature with agents Ag , fluents \mathcal{F} , actions \mathcal{A} and observations \mathcal{O} according to Definition 1.*

- A fluent literal is a fluent $f \in \mathcal{F}$ or its negation $\neg f$.
- Epistemic formulae are defined over \mathcal{F} and Ag as in DEL (cf. Section 2).
- Action laws are of the form

$$a \text{ causes } x \text{ if } \varphi$$

where $a \in \mathcal{A}$, x is either a fluent literal ℓ or an observation $o \in \mathcal{O}$, and φ is an epistemic formula.

Like in DEL planning and other action description languages, we need a means for defining the conditions under which an action can be executed; we again adopt a standard syntax (Baral et al. 2015) for these preconditions.

Definition 3. Consider a signature as in Definition 1. Precondition laws are of the form

$$\text{executable } a \text{ if } \pi$$

where $a \in \mathcal{A}$ is an action and π an epistemic formula as in Definition 2. An epistemic domain description \mathcal{D} is a set of action laws and precondition laws, one for every action $a \in \mathcal{A}$.

Example 1 (continued). The following action laws describe the effects and observations in the earlier example:

$$\begin{array}{llll} \text{open}(x) & \text{causes} & \text{opened} & \text{if } \top \\ \text{signal}(x, y) & \text{causes} & \text{looking}(y) & \text{if } \top \\ \text{distract}(x, y) & \text{causes} & \neg \text{looking}(y) & \text{if } \top \\ \\ \text{peek}(x) & \text{causes} & \text{obs}(x, \text{tail}) & \text{if } \text{tail} \\ \text{peek}(x) & \text{causes} & \text{obs}(y, \text{peek}(x)) & \text{if } \text{looking}(y) \\ \text{announceT}(x) & \text{causes} & \text{obs}(y, \text{tail}) & \text{if } \top \end{array}$$

As before, $x, y \in \{A, B, C\}$. The laws describing the effects of actions are accompanied by the precondition laws:

$$\begin{array}{llll} \text{executable } \text{open}(x) & \text{if } \text{has_key}(x) \\ \text{executable } \text{peek}(x) & \text{if } \text{opened} \wedge \text{looking}(x) \\ \text{executable } \text{signal}(x, y) & \text{if } \text{looking}(x) \wedge \neg \text{looking}(y) \\ \text{executable } \text{distract}(x, y) & \text{if } \text{looking}(x) \wedge \text{looking}(y) \\ \text{executable } \text{announceT}(x) & \text{if } K_x \text{tail} \end{array}$$

The above laws should be mostly self-explanatory. Notably, as will become clearer when we define the formal semantics, the observation “ $\text{obs}(x, \text{tail})$ if tail ” need not be accompanied by a symmetric law such as, say, $\text{peek}(x) \text{ causes } \text{obs}(x, \text{heads})$ if $\neg \text{tail}$. For when an agent x does not see $\text{obs}(x, \text{tail})$ after peeking inside the box, they know that tail must have been false. Any other agent y who is looking will see agent x executing $\text{peek}(x)$, but there is no observation to reveal the side of the coin to the first agent. In contrast to peeking, announcing “tail” provides all agents with $\text{obs}(y, \text{tail})$. In effect this signifies a public announcement, which by the last precondition axiom is possible only for an agent who knows tail .

Modularity and Representational Efficiency The small domain of Example 1 suffices to illustrate a key advantage of using action laws for epistemic planning instead of a direct description of event models. The latter would require, for each single action instance like $\text{peek}(A)$, a distinct event for every possible combination of effects on every

agent’s knowledge: The coin showing tails, agent B observing agent A while agent C is not looking gives rise to one event; the coin showing heads and both agents paying attention gives rise to another one, and so on. It is easy to see that when this example is generalised to n agents, $\{A_1, \dots, A_n\}$, then an exponential number of events is needed for a direct description as an event model. Whereas the modular description would be linear in n with the action $\text{peek}(x)$ alone, or quadratic when considering all instances of the actions $\text{signal}(x, y)$ and $\text{distract}(x, y)$.

Nondeterministic Actions Our formalisation of the main example illustrates how flexible and expressive the use of observations is when it comes to defining which aspects of an action, performed by one agent, the other agents become aware of, thus providing a lean but powerful high-level language to specify dynamic epistemic domains.

The general principle that agents do not necessarily know which action is performed can also be used to specify nondeterministic actions. In this case the executing agents themselves cannot decide which of several possible actions actually occur.

Example 2. Consider two agents $\text{Ag} = \{A, B\}$. The nondeterministic action of agent A tossing a coin may be described by the laws:

$$\begin{array}{llll} \text{toss}(A, 1) & \text{causes} & \text{tail} & \text{if } \top \\ \text{toss}(A, 2) & \text{causes} & \neg \text{tail} & \text{if } \top \\ \text{toss}(A, c) & \text{causes} & \text{obs}(A, \text{toss}, c) & \text{if } \text{looking}(A) \end{array}$$

$$\text{executable } \text{toss}(A, c) \text{ if } \top$$

where $c \in \{1, 2\}$ and $\omega(\text{obs}(A, \text{toss}, c)) = A$.

The first two laws define the two possible outcomes of agent A tossing the coin. The third law implies that A herself observes the *result* ($c = 1$ or $c = 2$) of tossing the coin provided she is looking at it. Both *actual* executions, $\text{toss}(A, 1)$ and $\text{toss}(A, 2)$, are always possible but, just like in standard DEL planning, an agent may be limited to deciding on executing only a *set* of possible actions without being able to pick a particular one (Bolander and Andersen 2011).

4 Semantics

The motivation behind enriching simple action effect laws with observations, owned by individual agents, is to provide compact, high-level descriptions of epistemic event models. In this section we show how any epistemic domain description in our new language DER can be formally interpreted by a unique event model, which is constructed in two steps.

1. An event is defined for every possible combination of conditions under which an action has one or more specific effects. For instance, the laws for $\text{peek}(A)$ in Example 1 give rise to various effects depending on which of tail , $\text{looking}(A)$, $\text{looking}(B)$, $\text{looking}(C)$ are true. Any combination that is consistent with the precondition for $\text{peek}(A)$, that is, $\text{opened} \wedge \text{looking}(A)$, determines an

event and its conditions and effects; e.g.¹

event e_1 with $pre_1 : op \wedge t \wedge l(A) \wedge l(B) \wedge \neg l(C)$
 event e_2 with $pre_2 : op \wedge \neg t \wedge l(A) \wedge l(B) \wedge \neg l(C)$

By the laws of Example 1 there are no effects on the underlying state, hence $eff_1 : \top$ and $eff_2 : \top$.

- The accessibility relations in the event model are then determined by the observations of the individual agents: Two events e and e' are indistinguishable for an agent i if, and only if, his own observations under $pre(e)$ and $pre(e')$ are the same according to the given action laws. For instance, e_1 and e_2 from above determine, respectively, the observations

$e_1 : \{obs(A, tail), obs(A, peek(A)), obs(B, peek(A))\}$
 $e_2 : \{obs(A, peek(A)), obs(B, peek(A))\}$

Hence, agent A can distinguish e_1 from e_2 while B cannot; however, B can distinguish these events from any action other than $peek(A)$, while for C any other event could have occurred that also triggers no observations for that agent.

All this is formalised as follows.

Definition 4. Consider an epistemic domain description \mathcal{D} with actions \mathcal{A} and agents Ag .

- For each action $a \in \mathcal{A}$ with action and precondition laws,

$$\begin{aligned} a \text{ causes } \ell_1^a \text{ if } \varphi_1^a \\ \vdots \\ a \text{ causes } \ell_m^a \text{ if } \varphi_m^a \\ a \text{ causes } o_1^a \text{ if } \varphi_{m+1}^a \\ \vdots \\ a \text{ causes } o_n^a \text{ if } \varphi_n^a \\ \text{executable } a \text{ if } \pi \end{aligned} \quad (1)$$

take each subset $\Phi \subseteq \{\varphi_1^a, \dots, \varphi_n^a\}$ such that

$$\bigwedge \Phi \stackrel{\text{def}}{=} \pi \wedge \bigwedge_{\varphi \in \Phi} \varphi \wedge \bigwedge_{\varphi_j^a \notin \Phi} \neg \varphi_j^a$$

is consistent, then define the event e_Φ^a by

- $pre : e_\Phi^a \mapsto \bigwedge \Phi$
- $eff : e_\Phi^a \mapsto \bigwedge \{\ell_j^a : \Phi \models \varphi_j^a, 1 \leq j \leq m\}$

- Let $E = \{e_\Phi^a\}_{a \in \mathcal{A}}$ be the set of all such events. For each agent $i \in Ag$ let Q_i be given by

$$(e_\Phi^a, e_{\Phi'}^a) \in Q_i \text{ iff } Obs_i(e_\Phi^a) = Obs_i(e_{\Phi'}^a)$$

where

$$Obs_i(e_\Phi^a) \stackrel{\text{def}}{=} \{o_j^a : \Phi \models \varphi_j^a, \omega(o_j^a) = i\}$$

The corresponding event model for domain description \mathcal{D} is $(\{e_\Phi^a\}_{a \in \mathcal{A}}, (Q_i)_{i \in Ag}, pre, eff)$.

¹Below, op , t , and l mean *opened*, *tail*, and *looking*, respectively.

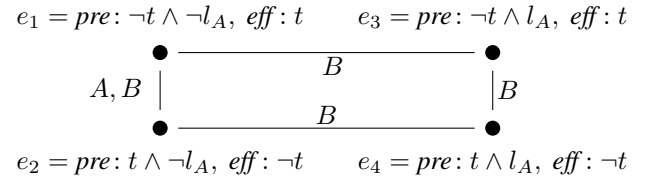


Figure 1: Event model for Example 3 (l_A means *looking*(A)).

Step 1 of Definition 4 reveals the reason behind the representational efficiency of action laws over a direct specification of an event model: Each consistent subset of conditions requires a separate event. Hence the exponential blowup of the representation in the number of agents that may or may not observe a specific action; cf. Section 3.

Since the event model for Example 1 with a total of 45 events² is too large to be given in full, let us illustrate the construction with a much smaller domain description.

Example 3. Consider agents $Ag = \{A, B\}$ and only one action, where agent A flips a coin and simultaneously learns whether it shows tail provided she is looking at the coin. This can be described as follows:

$flip(A)$ causes $tail$ if $\neg tail$
 $flip(A)$ causes $\neg tail$ if $tail$
 $flip(A)$ causes $obs(A, token)$ if $tail \wedge looking(A)$
 $flip(A)$ causes $obs(A, token)$ if $\neg tail \wedge looking(A)$

executable $flip(A)$ if \top

In step 1, there are four consistent combinations of effect conditions (after some straightforward logical simplification):

$$\begin{aligned} \Phi_1 &= \neg tail \wedge \neg looking(A) \\ \Phi_2 &= tail \wedge \neg looking(A) \\ \Phi_3 &= \neg tail \wedge looking(A) \\ \Phi_4 &= tail \wedge looking(A) \end{aligned}$$

These translate to the four events e_1 – e_4 depicted in Figure 1. Both agents cannot distinguish e_1 and e_2 since no observation is made in either case, while A , but not B , can distinguish e_3 and e_4 from each other and from e_1, e_2 .

The previous example provides an elementary case of action laws with ontic effects (i.e. which affect the state variables) combined with observations. Example 2 from Section 3 is also small enough to be illustrated here in full.

Example 2 (continued). Figure 2 depicts the event model corresponding to the nondeterministic tossing of a coin.

5 Completeness

Can every event model be described in our language? In this section we show that this is indeed the case: Any given event model can be converted into a *canonical* epistemic domain description. The construction is as follows.

First, each event is represented by a separate action. This action has the same precondition and effect as the event.

²Eight events for each instance of $peek(x)$; one event each for the 21 instances of the other actions with consistent precondition.

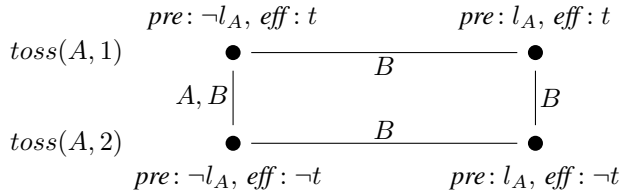


Figure 2: Event model for Example 2, with some edges omitted that follow from the transitivity of the indistinguishability relations.

Second, each action provides each agent with one observation. The observation is identical across all other actions that the agent cannot distinguish according to their individual accessibility relation. This is formalised as follows.

Definition 5 (Canonical description). *For an event model $\mathcal{E} = \langle E, (Q_i)_{i \in Ag}, pre, eff \rangle$, the canonical epistemic domain description is constructed as follows: Each event is an action, that is, $\mathcal{A} = E$. For each event $e \in E$, we define the following action laws and precondition law: Let $eff(e) = l_1 \wedge \dots \wedge l_m$. For each agent $i \in Ag$, let $\llbracket e \rrbracket_i$ be a symbol denoting the equivalence class of e in Q_i , then:*

$$\begin{aligned}
 e \text{ causes } l_1 & \quad \text{if } pre(e) \\
 & \quad \vdots \\
 e \text{ causes } l_m & \quad \text{if } pre(e) \\
 e \text{ causes } \llbracket e \rrbracket_i & \quad \text{if } pre(e) \quad \text{for all } i \in Ag \\
 \\
 \text{executable } e & \quad \text{if } pre(e)
 \end{aligned} \tag{2}$$

Where observation $\llbracket e \rrbracket_i$ is owned by agent i . We use $\mathcal{D}_{\mathcal{E}}$ to denote the set of action laws that constitute the canonical action description for \mathcal{E} .

Clearly, the canonical description is neither more compact nor easier to define than the event model itself. But the construction makes it easy to show that it can be applied to any event model and that the canonical description is interpreted by the very event model from which it has been produced.

Theorem 1 (Completeness). *Let $\mathcal{D}_{\mathcal{E}}$ be the canonical action description for an event model \mathcal{E} . Then the corresponding event model for $\mathcal{D}_{\mathcal{E}}$ is \mathcal{E} .*

Proof. First we apply Definition 4 to the laws (2) for any event e in \mathcal{E} . The only subset Φ of $\{pre(e)\}$ that is consistent with $\pi = pre(e)$ is $\Phi = \{pre(e)\}$. We thus obtain the event e with conditions $pre(e)$ and effects $l_1 \wedge \dots \wedge l_m = eff(e)$.

We then apply Definition 4 to the entire set of laws in $\mathcal{D}_{\mathcal{E}}$ in order to determine the accessibility relations Q_i for each agent $i \in Ag$. According to the only observations in (2), for any two events e and e' we have that $Obs_i(e) = Obs_i(e')$ if, and only if, $\llbracket e \rrbracket_i = \llbracket e' \rrbracket_i$. By Definition 5 this is equivalent to $Q_i(e, e')$ in \mathcal{E} . \square

This proves that DER is a complete description language in which every aspect can be modelled that can be expressed by event models. Moreover, it is easy to see that the canonical description is always polynomial in the size of an event

model \mathcal{E} : For each event e in \mathcal{E} with precondition of size n and effect of size m , we obtain $m + 2$ action laws (2) of size n . This shows that the exponential blowup, which can happen when action domain descriptions are directly represented as event models, does not occur the other way around.

6 Reasoning

We now extend our DER language to incorporate elements for representing initial (epistemic) states as well as reasoning about the consequences of actions. Combining these two elements with the event model representation allows us to describe DEL reasoning at a purely syntactic level. To achieve this we first define the required syntactic elements and then establish the semantics of how they translate into DEL epistemic models and product updates.

For specifying initial states and queries we adapt the syntax introduced by Baral et al. (2015), as well as drawing on a theory restriction from Son et al. (2014). It should be noted that this theory restriction is adequate for expressing standard benchmarks such as the Muddy Children problem (Son et al. 2014). We consider formulas of the form

$$\begin{aligned}
 \varphi & \tag{F1} \\
 \mathcal{C}(\varphi) & \tag{F2} \\
 \mathcal{C}(\mathcal{K}_i \varphi \vee \mathcal{K}_i \neg \varphi) & \tag{F3}
 \end{aligned}$$

where φ is a non-epistemic formula as in Definition 2. Intuitively, (F1) specifies properties that are true in the actual world, while (F2) and (F3) specify agent knowledge about properties of the world.

Definition 6 (Initial state). *Consider a signature with agents Ag and fluents \mathcal{F} according to Definition 1. Initial state formulas are of the form*

$$\text{initially } \varphi$$

where φ is an epistemic formula of the form (F1)-(F3).

Definition 7 (Entailment). *Consider a signature with agents Ag , fluents \mathcal{F} , and actions \mathcal{A} according to Definition 1. Entailment query formulas are of the form*

$$\varphi \text{ after } \delta$$

where φ is an epistemic formula as in Definition 2 and δ is a sequence of action instances $a_1; \dots; a_n$ such that $a_i \in \mathcal{A}$.

An initial state and a set of entailment queries can now be combined with action and precondition laws to specify a reasoning problem over a domain.

Example 3 (continued). *Recall the example of an agent A flipping a coin and learning its value depending on whether she is looking at the coin. We can describe an initial state where the coin is showing tails, although neither agent is aware of this, and it is common knowledge that agent A is looking for the coin flip.*

$$\begin{aligned}
 & \text{initially tail} \\
 & \text{initially } \mathcal{C}(\text{looking}(A))
 \end{aligned}$$

After executing the flip(A) action we can expect that agent A knows the state of the coin, while agent B know that A knows but remains ignorant of the state of the coin itself:

$$\begin{aligned}
 & \mathcal{K}_A \neg \text{tail} \quad \text{after } \text{flip}(A) \\
 & \neg \mathcal{K}_B \neg \text{tail} \quad \text{after } \text{flip}(A) \\
 & \mathcal{K}_B (\mathcal{K}_A \text{tail} \vee \mathcal{K}_A \neg \text{tail}) \quad \text{after } \text{flip}(A)
 \end{aligned}$$

From this example we can observe that formulas of the form (F1)-(F3) cannot explicitly express that agents A and B are ignorant of the state of the coin. Instead, we must rely on a *closed world assumption* to implicitly derive such negative information; the intuition being that what is not explicitly specified as known can be assumed to be unknown.

We now turn to providing the semantic underpinnings for these language constructs and intuitions. We do this by showing how they correspond to DEL epistemic states and actions. In particular we show that an initial state specification can be translated into a *primitive finitary S5-theory* (Son et al. 2014); that this corresponds directly to a DEL epistemic state; and that action entailment queries map directly to DEL product update operations.

6.1 Initial State Semantics

To capture the semantics of an initial state specification we first consider the class of *finitary S5-theories* (Son et al. 2014). **S5**-theories satisfy the standard **S5** modal axioms (**K,T,4,5**), extended to the multi-modal setting. This means that the accessibility relations are equivalence relations, which is the standard setting for specifying knowledge and also the setting that we consider for our event models.

Unfortunately, **S5**-theory semantics is defined in terms of *pointed epistemic models* (or *pointed Kripke structures*), whereas the DEL literature typically considers the extension to *multi-pointed epistemic models*. Therefore we first establish some basic results in order to align these two semantics.

Aligning Epistemic Model Semantics A common semantics is defined in terms of a satisfaction relation over a pointed epistemic model $\langle M, w \rangle$. Here $M = \langle W, R, V \rangle$ is a Kripke structure over a set of worlds W , where R is a set of accessibility relations such that R_i is the accessibility for agent $i \in Ag$, and V is a propositional valuation for each world. Finally, $w \in W$ is a *designated* world denoting the actual state of affairs.

Definition 8 (Truth in a pointed epistemic model; Bolander and Andersen 2011). *Let a Kripke structure $M = \langle W, R, V \rangle$ of $\mathcal{L}_{\mathcal{F}, Ag}$ be given. Let $i \in Ag$, $w \in W$ and $\phi, \psi \in \mathcal{L}_{\mathcal{F}, Ag}$, then*

$$\begin{aligned} \langle M, w \rangle &\models \top && \text{always} \\ \langle M, w \rangle &\models \perp && \text{never} \\ \langle M, w \rangle &\models p && \text{iff } w \in V(p) \\ \langle M, w \rangle &\models \neg\phi && \text{iff } \langle M, w \rangle \not\models \phi \\ \langle M, w \rangle &\models \phi \wedge \psi && \text{iff } \langle M, w \rangle \models \phi \text{ and } \langle M, w \rangle \models \psi \\ \langle M, w \rangle &\models \mathcal{K}_i\phi && \text{iff for all } v \in W, \\ &&& \text{if } wR_iv \text{ then } \langle M, v \rangle \models \phi \\ \langle M, w \rangle &\models \mathcal{C}\phi && \text{iff for all } v \in W, \\ &&& \text{if } w(\cup_{j \in Ag} R_j)^*v \text{ then } \langle M, v \rangle \models \phi \end{aligned}$$

where R^* is the transitive closure of R .

A pointed epistemic model is often used to capture a god's-eye view of the world, where the modeller is able to point to the actual world and the remaining worlds in the Kripke structure encode agents' knowledge. However, an important application of DEL is for epistemic planning, where planning takes place from the perspective of a specific

agent. In such a case the planning agent may not have complete knowledge of the objective world so may not be able to point to the actual world. This motivates the extension of the DEL semantics to *multi-pointed epistemic models*, where an epistemic model can have multiple designated worlds.

Definition 9 (Truth in a multi-pointed epistemic model; Bolander and Andersen 2011). *Let a Kripke structure $M = \langle W, R, V \rangle$ of $\mathcal{L}_{\mathcal{F}, Ag}$ be given. Let $i \in Ag$, $\emptyset \subset W_d \subseteq W$ and $\phi \in \mathcal{L}_{\mathcal{F}, Ag}$, then*

$$\langle M, W_d \rangle \models \phi \text{ iff } \langle M, w \rangle \models \phi \text{ for all } w \in W_d$$

Any pointed epistemic model simply maps to a corresponding multi-pointed epistemic model containing only a single designated world. Therefore, unless specifically stated we drop the single/multi-pointed descriptor when referring to arbitrary epistemic models. We also adopt the terminology that for a Kripke structure $M = \langle W, R, V \rangle$, $\langle M, W_d \rangle$ is a *finite epistemic model* iff W is a finite set. Furthermore, if every relation R_i , for each $i \in Ag$, is an equivalence relation then we say that $\langle M, W_d \rangle$ is an **S5** model.

Next, to highlight the link between sets of epistemic models, combined with sets of designated worlds, we extend the satisfaction relation to sets.

Definition 10 (Truth of a set of epistemic models). *Let $\mathcal{L}_{\mathcal{F}, Ag}$ be an epistemic language for a set of epistemic models S and $\phi \in \mathcal{L}_{\mathcal{F}, Ag}$, then*

$$S \models \phi \text{ iff } \langle M, W_d \rangle \models \phi \text{ for all } \langle M, W_d \rangle \in S$$

We can now establish a basic property of the relationship between arbitrary epistemic models and single pointed epistemic models.

Theorem 2. *For any set $S = \{\langle M, W_{d_1} \rangle, \dots, \langle M, W_{d_n} \rangle\}$ of epistemic models of a language $\mathcal{L}_{\mathcal{F}, Ag}$ we have that*

$$\begin{aligned} S \models \phi &\text{ iff } \{\langle M, w_1 \rangle, \dots, \langle M, w_n \rangle\} \models \phi, \\ &\text{ for all } \phi \in \mathcal{L}_{\mathcal{F}, Ag}, \text{ where } w_i \in \cup_{\langle M, W_d \rangle \in S} W_d. \end{aligned}$$

Proof. This follows directly from the definitions of satisfaction: $S \models \phi$ iff $\langle M, W_d \rangle \models \phi$ for all $\langle M, W_d \rangle \in S$. But for each $\langle M, W_d \rangle \in S$, $\langle M, W_d \rangle \models \phi$ iff $\langle M, w \rangle \models \phi$ for each $w \in W_d$, hence $\langle M, w \rangle \models \phi$ for every world $w \in \cup_{\langle M, W_d \rangle \in S} W_d$. \square

Theorem 2 clarifies a simple relationship between the semantics of single and multi-pointed epistemic models. In particular, while they have the same expressive power, nevertheless the multi-pointed case can provide for a more compact encoding. For example, a theory that is characterised by a set of pointed epistemic models sharing a single Kripke structure can be equivalently characterised by a single multi-pointed epistemic model.

To focus on practical applications of DEL we now consider the restricted setting of finite epistemic models and finite sets of finite epistemic models.

Theorem 3. *For any finite set S of finite epistemic models of a language $\mathcal{L}_{\mathcal{F}, Ag}$, there exists a finite epistemic model $\langle M, W_d \rangle$ such that*

$$\begin{aligned} S \models \phi &\text{ iff } \langle M, W_d \rangle \models \phi, \text{ for all } \phi \in \mathcal{L}_{\mathcal{F}, Ag}, \text{ and} \\ &\langle M, W_d \rangle \text{ is an S5 model iff } S \text{ is a set of S5 models.} \end{aligned}$$

Proof. We can construct a candidate $\langle M, W_d \rangle$ by mapping each element of S to distinct sub-graphs within M , and setting $W_d = \cup_{\langle M_x, W_{d_x} \rangle \in S} W_{d_x}$. Such a mapping is possible because for each model $\langle M_x, W_{d_x} \rangle \in S$ there exists a bisimulation equivalent model where the worlds between models of S are distinct. Since each model in S maps to disjoint sub-graphs it also preserves any equivalences in the accessibility relations, so if all models in S are **S5** then $\langle M, W_d \rangle$ is also an **S5** model. Finally, for all $\phi \in \mathcal{L}_{\mathcal{F}, Ag}$, $S \models \phi$ iff $\langle M_x, W_{d_x} \rangle \models \phi$ follows from the construction. \square

Theorem 3 establishes that any theory that can be characterised by a finite set of finite epistemic models can also be characterised by a single epistemic model. We now turn back to considering finitary **S5**Theories as one such case.

Finitary S5-Theories The class of finitary **S5**-theories is defined in multiple steps. Firstly, a language restriction is defined that includes formulas of the form (F1)-(F3) as well as formulas of the form

$$\mathcal{C}(\neg \mathcal{K}_i \varphi \wedge \neg \mathcal{K}_i \neg \varphi) \quad (F4)$$

Intuitively, formulas of the form (F4) specify what agents don't know. Next a further restriction is considered; a *complete clause over \mathcal{F}* is a disjunction of the form $\bigvee_{p \in \mathcal{F}} p^*$ where p^* is either p or $\neg p$. The definition of a *primitive finitary S5-theory* then follows.

Definition 11 (Primitive Finitary **S5**-Theory; Son et al. 2014). *A theory T is said to be primitive finitary S5 if*

- Each formula in T is of the form (F1)-(F4); and
- For each complete clause φ over \mathcal{F} and each agent i , T contains either (i) $\mathcal{C}(\varphi)$ or (ii) $\mathcal{C}(\mathcal{K}_i \varphi \vee \mathcal{K}_i \neg \varphi)$ or (iii) $\mathcal{C}(\neg \mathcal{K}_i \varphi \wedge \neg \mathcal{K}_i \neg \varphi)$.

T is said to be in disjunctive form if all statements in T are in disjunctive form.

Finally, the notion of a finitary **S5**-theory is defined in terms of entailment of a primitive finitary **S5**-theory.

Definition 12 (Finitary **S5**-Theory; Son et al. 2014). *An epistemic 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 (F1)-(F4).*

The key feature and benefit of finitary **S5**-theories is that they can be characterised by finitely many finite models.

Theorem 4 (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.*

When viewed in terms of standard DEL multi-pointed epistemic models we can take Theorem 4 further and establish the stronger result that every finitary **S5**-theory can be characterised by a single (finite) epistemic model.

Theorem 5. *For any finitary S5-theory T there exists a finite epistemic model $\langle M, W_d \rangle$ such that*

$$\text{Mod}_{\text{S5}}(T) \models \phi \text{ iff } \langle M, W_d \rangle \models \phi, \text{ for all } \phi \in \mathcal{L}_{\mathcal{F}, Ag};$$

where $\text{Mod}_{\text{S5}}(T)$ is the set of models of T .

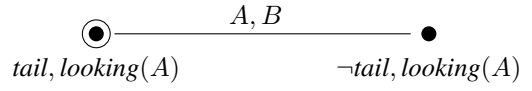


Figure 3: Epistemic models characterising the initial state for Example 3; *tail* holds but agents A and B don't know it.

Proof. Follows directly from Theorems 3 and 4. \square

However, despite the desirability of this finiteness result, Definition 12 does not actually define a specific syntactic restriction for **S5**-theories; making it difficult to use as the basis of an epistemic state specification language. Instead a syntax is only defined for the sub-class of primitive finitary **S5**-theories. But this sub-class is cumbersome to use for specifying states since it requires a formula of the form (F2)-(F4) for every complete clause and agent combination.

Compactly Representing Initial States Fortunately, Son et al. (2014) propose an approach to compactly specifying epistemic states; taking statements of the form (F1)-(F3) and generating a set of completion statements of the form (F4). We adopt this proposal here; generating a primitive finitary **S5**-theory through the application of a completion operator.

Definition 13 (Completion operator). *The completion of a set of initial state formulas I with respect to an epistemic language $\mathcal{L}_{\mathcal{F}, Ag}$ is given by $\text{comp}(I) = F_1(I) \cup F_2(I) \cup F_3(I) \cup F_4(I)$, where*

$$\begin{aligned} F_1(I) &= \{ \psi \mid \psi \in \Phi(I) \text{ is a non-epistemic formula} \}, \\ F_2(I) &= \{ \mathcal{C}(\psi) \mid \psi \text{ is a complete clause over } \mathcal{F} \text{ s.t.} \\ &\quad \varphi \models \psi \text{ where } \mathcal{C}(\varphi) \in \Phi(I) \}, \\ F_3(I) &= \{ \mathcal{C}(\mathcal{K}_i \psi \wedge \mathcal{K}_i \neg \psi) \mid \\ &\quad \psi \text{ is a complete clause over } \mathcal{F} \\ &\quad \text{s.t. } \varphi \models \psi \text{ where } \mathcal{C}(\mathcal{K}_i \varphi \wedge \mathcal{K}_i \neg \varphi) \in \Phi(I) \}, \\ F_4(I) &= \bigcup_{i \in Ag} \{ \mathcal{C}(\neg \mathcal{K}_i \varphi \wedge \neg \mathcal{K}_i \neg \varphi) \mid \varphi \in \Psi(I, i) \}, \\ \Psi(I, i) &= \{ \varphi \mid \varphi \text{ is a complete clause over } \mathcal{F} \text{ s.t.} \\ &\quad \Phi(I) \not\models \mathcal{C}(\varphi), \Phi(I) \not\models \mathcal{C}(\mathcal{K}_i \varphi \wedge \mathcal{K}_i \neg \varphi) \}, \\ \Phi(I) &= \{ \varphi \mid \text{initially } \varphi \text{ is a statement in } I \}. \end{aligned}$$

This completion operator generates a primitive finitary **S5**-theory from a set of input statements of the form (F1)-(F3). The set $F_1(I)$ consists of the objective facts of the world, $F_2(I)$ and $F_3(I)$ are the positive complete clausal statements for the (F2) and (F3) statements in I . Finally, $F_4(I)$ encodes the implicit completion of what is not known.

Example 3 (continued). *Continuing the coin flip example, the completion of the initial state specification results in the following primitive finitary S5-theory (characterised by the epistemic model of Figure 3):*

$$\begin{aligned} &tail, \mathcal{C}(tail \vee looking(A)), \mathcal{C}(\neg tail \vee looking(A)), \\ &\mathcal{C}(\neg \mathcal{K}_A(tail \vee \neg looking(A)) \wedge \neg \mathcal{K}_A \neg(tail \vee \neg looking(A))), \\ &\mathcal{C}(\neg \mathcal{K}_B(tail \vee \neg looking(A)) \wedge \neg \mathcal{K}_B \neg(tail \vee \neg looking(A))). \end{aligned}$$

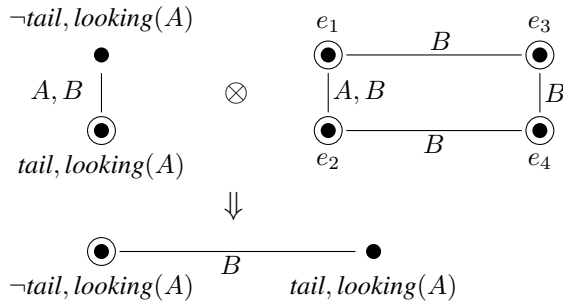
In summary, when combined with the completion operator, an initial state specification corresponds to a primitive finitary **S5**-theory. This in turn can be characterised by a single finite epistemic model (Theorem 5). We now show how this epistemic model can be used as part of a DEL product update operation to perform query answering.

6.2 Epistemic Reasoning About Actions

Query specifications (Definition 7) are defined in terms of the consequences of performing a sequence of actions, $a_1; \dots; a_n$, from some initial epistemic state. Crucially, each a_i corresponds to a DEL action.

Because of the need to capture both ontic and epistemic change, the semantics of DEL actions are more complex than their traditional non-epistemic counterparts. In particular $a = \langle \mathcal{E}, E_d \rangle$ is a DEL action where $\mathcal{E} = \langle E, (Q_i)_{i \in Ag}, pre, eff \rangle$ is an event model and $E_d \subseteq E$ is a set of distinguished events. Updating an epistemic state by an action to generate a new epistemic state involves a product update operation. We do not reproduce these definitions here, but the interested reader is referred to Definitions 8 and 9 from Bolander and Andersen (2011). Instead we highlight the behaviour of the product update operator by way of our running example.

Example 3 (continued). *The initial epistemic state, where agents A and B don't know that the coin is displaying tails but agent A is watching for the coin flip (Figure 3), is updated by the $flip(A)$ action. This action is constructed from the event model (Figure 1) combined with the distinguished events associated with the action. The product update is applied to generate a new epistemic state:*



The truth of the queries posed earlier in this example follows directly from the updated epistemic model. In the resulting model, agent A knows the actual state of the world, and therefore knows that the coin shows heads. However, agent B still does not know the coin's status, but does know that A knows it. Finally, it should be noted that the DEL action in this example consists only of distinguished events because the example itself models only a single action.

One final observation about the query specification: The action has been specified from a god's-eye perspective, where the precise action being executed is known. This can be extended to the agent's perspective, for example in the non-deterministic case where the agent does not know which action was actually executed. We leave this for future work.

7 Related Work

In this section we consider how our DER language relates to other epistemic multi-agent domain representations. The key point is that DER provides a sound and complete characterisation of DEL event models, and combined with the initial state and query specifications, provides a syntactic characterisation of reasoning about the consequences of DEL actions. In contrast, most alternative approaches consider only

a subset of DEL event models or employ a non-standard (i.e., non-DEL) semantics altogether.

7.1 The Action Language $m\mathcal{A}^*$

The approach that is syntactically closest to DER is the action language $m\mathcal{A}^*$ (Baral et al. 2015). It is of particular note because, along with an earlier version $m\mathcal{A}^+$ (Baral, Gelfond, and Son 2012), it has been the foundation for on-going extensions (Wright and Pontelli 2018) as well as being the basis for epistemic planning systems (Le et al. 2018).

There are a number of syntactic differences between DER and $m\mathcal{A}^*$. Firstly, $m\mathcal{A}^*$ does not contain a set of observations or an ownership function to map observations to agents; instead it requires multiple action and observability constructs. It distinguishes three action types: *ontic*, *sensing*, and *announcement*, specified using three language constructs: **causes**, **determines**, and **announces** (respectively). It also uses the language constructs **observes** and **aware_of** to distinguish between full and partial observations. The use of observations in DER is arguably simpler and more intuitive than having these complex language constructs.

Example 1 (continued). *Recall the box observability example. The DER and $m\mathcal{A}^*$ encodings of this example only differ in how they handle non-ontic actions (ignoring some superficial syntactic differences). These actions are encoded in DER as causing observations:*

| | | | | |
|----------------|---------------|-------------------|-----------|--------------|
| $peek(x)$ | causes | $obs(x, tail)$ | if | $tail$ |
| $peek(x)$ | causes | $obs(y, peek(x))$ | if | $looking(y)$ |
| $announceT(x)$ | causes | $obs(y, tail)$ | if | \top |

In contrast $m\mathcal{A}^$ requires a specification of both the action type and its observability consequences:*

| | | |
|----------------|-------------------|----------------------------------|
| $peek(x)$ | determines | $tail$ |
| $announceT(x)$ | announces | $tail$ |
| y | observes | $announceT(x)$ if \top |
| x | observes | $peek(x)$ if \top |
| y | observes | $open(x)$ if $looking(y)$ |
| y | aware_of | $peek(x)$ if $looking(y)$ |

Beyond syntactic differences, $m\mathcal{A}^*$ is strictly less expressive than DER, since Baral et al. (2015) show that it cannot capture all DEL event models. Finally, $m\mathcal{A}^*$ does not use the standard DEL product update but instead introduces its own *transition semantics* for state updates. The precise relationship between the transition semantics of $m\mathcal{A}^*$ and standard DEL product update has not been established.

This last point opens up a potential application for DER as a mechanism to prove the correctness of $m\mathcal{A}^*$ with respect to DEL semantics. This process could take place by establishing a translation of any $m\mathcal{A}^*$ domain specification to a DER domain specification. As DER is strictly more expressive than $m\mathcal{A}^*$, such a translation is straightforward. In particular it would require turning $m\mathcal{A}^*$ observability statements into DER observations, with accompanying ownership assignments and corresponding causal rules. Sensing and announcement statements would require additional DER observations, ownership assignments, and causal rules. Since a DER description maps directly to an event model, it would then be possible to validate the results of a DEL product update against the $m\mathcal{A}^*$ transition update.

7.2 Other Epistemic Representation Formalisms

Apart from $m\mathcal{A}^*$ and its extensions, there have been a number of other formalisms that deal with dynamic epistemic domains, especially in the context of planning. Most of these methods enforce some form of epistemic restriction with the intention of making reasoning more practical.

A useful approach to making epistemic planning more practical is to find a DEL fragment that can be encoded as a classical planning problem (Muise et al. 2015; Kominis and Geffner 2015; Cooper et al. 2016). Encoding any strict DEL fragment into DER would be possible using the technique for converting a DEL event model into a canonical DER description (Definition 5). However, a more intuitive translation could potentially also be provided based on the specific DEL restriction under consideration.

Typically the restrictions involve providing a mechanism in which to explicitly represent an agent’s knowledge (or beliefs) but to limit it in some way. For example, Muise et al. (2015) consider the case of non-disjunctive beliefs with bounded nesting of beliefs. Then they allow for action effect axioms that directly modify those beliefs. Effectively this treats the beliefs of an agent as a set of special agent-specific fluents, thus increasing the number of fluents in the representation by the depth of the desired belief nesting and the number of agents. As mentioned previously, DER is restricted to epistemic states so cannot model the belief aspects of the Muise et al. (2015) proposal. Nevertheless, as a general rule translating these types of restricted DEL formalisms into DER would involve replacing the epistemic modifying effect axioms with DER action causation statements that generate appropriate observations.

Herzig, Lorini, and Maffre’s (2015) language uses a form of (nested) “observability fluents” to express that an agent can observe the state of a property p of the world, or that an agent can observe that another agent can observe p etc. These observability fluents are then changed directly as explicit effects of actions. In DER, in contrast, observations and fluents are two independent entities, and any facts about an agent knowing the value of a fluent, or knowing that another agent knows the value of a fluent, are implicitly obtained and therefore need not be explicitly specified as the effect of an action.

Charrier and Schwarzenruber’s (2017) succinct event models provide a “procedural” way to specify event models in the form of a formal program, also possibly exponentially more compact, to construct a model. This differs from the approach taken by DER, whose main purpose is to provide, as simply as possible, a modular and declarative language for describing the actions of a multi-agent, epistemic planning domain. The declarative simplicity of the use of *observations* also contrasts DER from other languages for implicitly specifying action models, such as French, Hales, and Tay’s (2014) syntactic operators, or state transition system models such as derived from so-called knowledge-based programs. A detailed discussion on the commonalities and differences between action languages and logics that use operators or programs, is provided by Baral et al. (2015).

Finally, it is worth briefly returning to the epistemic extension to GDL (Thielscher 2017). As mentioned in the in-

roduction GDL has a special-purpose nature; it is intended for specifying games to be played as part of a larger game playing framework. This can make it cumbersome to use as a general purpose language for describing epistemic planning problems. For example, it is not possible to specify an arbitrary initial epistemic state. Nevertheless, it uses a similar observation token mechanism to our proposal, whereby each player can receive different percepts as a result of the same action. Consequently, translating these epistemic state change constructs into the DER language would be straightforward; the main challenge arising from the need to specify the joint action of all the game players.

8 Conclusion

Multi-agent epistemic planning is an important research topic with broad applicability; from modelling security games through to developing service robots that operate and interact with humans in complex environments. Dynamic epistemic logic (DEL) is a general, highly expressive, formalism for representing these environments but suffers from the requirement to encode actions and their effects using Kripke-style event models.

In this paper we introduced a high-level description language for representing actions and their effects that is both simple and compact, yet provides a (provably) full characterisation of DEL event models. We showed how this language can be used to intuitively encode actions and agent observations and reason about their effects. We then provided a comparison to previous representation formalisms, highlighting both the simplicity and power of our approach.

For future work we propose to consider two directions: applications and extension. We have shown how to use DER to reason about the consequences of actions from a god’s-eye perspective. However, for epistemic planning we need to consider alternative execution models, such as planning with non-deterministic actions or dealing with implicit cooperation amongst agents (Engesser et al. 2017). For example, for non-deterministic actions a planner must consider that the agent will, non-deterministically, execute a DEL event from some set, rather than selecting a specific event. We will seek additional language constructs to simply and compactly specify these different types of execution models.

The second direction for future work is to consider two types of extensions. Currently, our initial state representation relies on a transformation to a primitive finitary **S5**-theory (Son et al. 2014). We will explore other expressive theory restrictions that also share the finiteness property of **S5**-theories. Finally, we will explore extensions to deal with the more general case of updating belief states. As discussed in Section 2 multi-agent doxastic planning is a significant research challenge (Baral et al. 2017). However, it is also a widely applicable problem with very simple and realistic scenarios; for example a robot that leaves a room will fail to observe any changes that other agents make in that room and can therefore be lead to believing something that is not true. Recovering from false beliefs is a problem that goes beyond the standard DEL model so would require both extending our DER representation language but also tackling deeper representational challenges.

References

- Andersen, M. B.; Bolander, T.; and Jensen, M. H. 2015. Don't plan for the unexpected: Planning based on plausibility models. *Logique et Analyse* 58(230):145–176.
- Baltag, A.; Moss, L. S.; and Solecki, S. 1998. The logic of public announcements and common knowledge and private suspicions. In Gilboa, I., ed., *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98)*, Evanston, IL, USA, July 22-24, 1998, 43–56. Morgan Kaufmann.
- Baral, C., and Gelfond, G. 2011. On representing actions in multi-agent domains. In Balduccini, M., and Son, T. C., eds., *Logic Programming, Knowledge Representation, and Non-monotonic Reasoning - Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday*, volume 6565 of *Lecture Notes in Computer Science*, 213–232. Springer.
- Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2015. An action language for multi-agent domains. In *The Computing Research Repository*. CoRR abs/1511.01960.
- Baral, C.; Bolander, T.; van Ditmarsch, H.; and McIlraith, S., eds. 2017. *Epistemic Planning*, volume 17231 of *Dagstuhl Reports*. Germany: Dagstuhl Publishing.
- Baral, C.; Gelfond, G.; and Son, T. C. 2012. An action language for reasoning about beliefs in multi-agent domains. In *Proceedings of the 14th international workshop on non-monotonic reasoning*.
- Baum, L., and Petrie, T. 1966. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics* 37(6):1554–1563.
- Bolander, T., and Andersen, M. B. 2011. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics* 21(1):9–34.
- Charrier, T., and Schwarzentruher, F. 2017. A succinct language for dynamic epistemic logic. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, 123–131.
- Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; and Régnier, P. 2016. A simple account of multi-agent epistemic planning. In Kaminka, G. A.; Fox, M.; Bouquet, P.; Hüllermeier, E.; Dignum, V.; Dignum, F.; and van Harmelen, F., eds., *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, 193–201. IOS Press.
- Engesser, T.; Bolander, T.; Mattmüller, R.; and Nebel, B. 2017. Cooperative epistemic multi-agent planning for implicit coordination. In Ghosh, S., and 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*, volume 243 of *EPTCS*, 75–90.
- Engesser, T.; Mattmüller, R.; Thielscher, M.; and Nebel, B. 2018. Epistemic game description language and dynamic epistemic logic compared. In Lang, J., ed., *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1795–1802. Stockholm: AAAI Press.
- Fikes, R., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.
- French, T.; Hales, J.; and Tay, E. 2014. A composable language for action models. *Advances in Modal Logic* 197–216.
- Gelfond, M., and Lifschitz, V. 1993. Representing action and change by logic programs. *Journal of Logic Programming* 17:301–321.
- Herzig, A.; Lorini, E.; and Maffre, F. 2015. A poor man's epistemic logic based on propositional assignment and higher-order observation. In *International Workshop on Logic, Rationality and Interaction*, 156–168. Springer.
- Kominis, F., and Geffner, H. 2015. Beliefs in multi-agent planning: From one agent to many. In Brafman, R. I.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015*, 147–155. AAAI Press.
- Le, T.; Fabiano, F.; Son, T. C.; and Pontelli, E. 2018. EFP and PG-EFP: epistemic forward search planners in multi-agent domains. In de Weerd, M.; Koenig, S.; Röger, G.; and Spaan, M. T. J., eds., *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018*, 161–170. AAAI Press.
- Muise, C. J.; Belle, V.; Felli, P.; McIlraith, S. A.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2015. Planning over multi-agent epistemic states: A classical planning approach. In Bonet, B., and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 3327–3334. AAAI Press.
- Reiter, R. 2001. *Knowledge in Action*. MIT Press.
- Son, T. C.; Pontelli, E.; Baral, C.; and Gelfond, G. 2014. Finitary S5-theories. In Fermé, E., and Leite, J., eds., *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, volume 8761 of *Lecture Notes in Computer Science*, 239–252. Springer.
- Son, T. C.; Pontelli, E.; Baral, C.; and Gelfond, G. 2015. Exploring the KD45 property of a Kripke model after the execution of an action sequence. In Bonet, B., and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 1604–1610. AAAI Press.
- Thielscher, M. 2017. GDL-III: A description language for epistemic general game playing. In Sierra, C., ed., *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1276–1282.
- Wright, B., and Pontelli, E. 2018. Reasoning with doxastic attitudes in multi-agent domains. In Brawner, K., and Rus, V., eds., *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 360–365. AAAI Press.