Epistemic planning for single- and multi-agent systems

Thomas Bolander — Mikkel Birkegaard Andersen

Technical University of Denmark, DTU Informatics
Richard Petersens Plads, building 321, DK-2800 Lyngby (Denmark)
tb@imm.dtu.dk, mibi@imm.dtu.dk

ABSTRACT. *In this paper, we investigate the use of event models for automated planning. Event models are the action defining structures used to define a semantics for dynamic epistemic logic. Using event models, two issues in planning can be addressed: Partial observability of the environment and knowledge. In planning, partial observability gives rise to an uncertainty about the world. For single-agent domains, this uncertainty can come from incomplete knowledge of the starting situation and from the nondeterminism of actions. In multi-agent domains, an additional uncertainty arises from the fact that other agents can act in the world, causing changes that are not instigated by the agent itself. For an agent to successfully construct and execute plans in an uncertain environment, the most widely used formalism in the literature on automated planning is "belief states": sets of different alternatives for the current state of the world. Epistemic logic is a significantly more expressive and theoretically better founded method for representing knowledge and ignorance about the world. Further, epistemic logic allows for planning according to the knowledge (and iterated knowledge) of other agents, allowing the specification of a more complex class of planning domains, than those simply concerned with simple facts about the world. We show how to model multi-agent planning problems using Kripke-models for representing world states, and event models for representing actions. Our mechanism makes use of slight modifications to these concepts, in order to model the internal view of agents, rather than that of an external observer. We define a type of planning domain called epistemic planning domains, a generalisation of classical planning domains, and show how epistemic planning can successfully deal with partial observability, nondeterminism, knowledge and multiple agents. Finally, we show epistemic planning to be decidable in the single-agent case, but only semi-decidable in the multi-agent case.*

KEYWORDS: *automated planning, epistemic logic, dynamic epistemic logic, event models, partial observability, plan-time vs. run-time knowledge.*

DOI:10.3166/JANCL.21.9–33 © 2011 Lavoisier, Paris

1. Introduction

For most of its early life in the 60's and 70's, the field of automated planning was concerned with ways in which the problem of creating long-term plans for achieving goals could be formulated, such that solving problems of non-trivial size, would be computationally feasible. The types of planning that arose from this early work, is what is known today as Classical Planning. Classical Planning, as defined by (Ghallab *et al.*, 2004), imposes a number simplifying restrictions on the planning problem, namely that it be *finite*, *fully observable*, *deterministic* and *static*.

While there certainly are computational benefits to the above restrictions, it is also clear that such planning domains are much easier to construct theoretically sound planning algorithms for. In other words, the reason that Classical Planning became so dominant was not only due to limited computational resources, but also a limited understanding of the intricacies of how, for instance, to take the actions of other agents into account when planning, or how to naturally represent incomplete information about the world state – the complexity of automated planning is not solely computational.

In this paper, we examine a new method of planning, with which the full observability and determinism requirement can be lifted. Getting partial observability comes from the use of epistemic Kripke-models to represent knowledge about the world, recognising that partial observability and knowledge (or a lack thereof) are two sides of the same coin. Event models, taken from dynamic epistemic logic, are used in defining the ways in which actions change epistemic models, whether they are factual – changing propositional facts about the world – or epistemic – changing knowledge of the facts, but not the facts themselves – or a combination thereof. In addition, event models provide a natural way to handle nondeterminism. Epistemic planning, as we name this new approach, will be considered in both single- and multi-agent versions.

Consider the similarities between belief states, the most widely used method in the literature on automated planning for dealing with the incomplete knowledge that arises from partial observability, and Kripke-models for epistemic logic. Belief states are sets of propositions about the world, each of which represents an alternative version of the world. In epistemic modal logic, each world also represents an alternative, but with the addition of a notion of indistinguishability of these alternatives by particular agents. Even without going into details about models of epistemic logic, it is immediately obvious that epistemic logic is at least as expressive as belief states when it comes to planning, and, as the reader will learn, they are actually much more so. With the combination of epistemic logic and event models, we gain the ability to plan in nondeterministic, partially observable, multi-agent domains with knowledge, where belief states only affords us the ability to deal with the first two. Further, epistemic planning, as we call this new paradigm, internalises nondeterminism and observability in the planning language, rather than dealing with it at an algorithmic level. We find this to be a much more satisfying approach.

The remainder of this paper is organised as follows. Section 2 introduces the the well known notions of epistemic models from the literature on modal logic, and shows how they, with rather elegant modifications, can be used to model the internal view of an agent involved in the situation being modeled. Section 3 introduces our version of event models, which are largely similar to those of dynamic epistemic logic, with minor modifications to facilitate the internal view. In section 4 we show definitions of classical planning problems, epistemic planning domains and their correspondences. With epistemic planning defined, section 5 examines properties of different types of actions based on event models, and establishes a nomenclature for these. Finally, sections 6 and 7 deals with the single- and multi-agent versions of epistemic planning domains respectively, and gives decidability results for both.

2. Epistemic logic and epistemic states

In this section we present the notions from (dynamic) epistemic logic required for the remaining article. First of all, we define a language of epistemic logic. Let P be a finite set of atomic propositions (propositional symbols), and $\mathcal{A}$ a finite set of agents. We will most often use symbols $p, q, r, s, \ldots$ for atomic propositions and $i, j, k, l, \ldots$ for agents. The language $\mathcal{L}_K(P, \mathcal{A})$, the language of multi-agent epistemic logic on $(P, \mathcal{A})$, is generated by the following BNF:

$$\phi ::= \top \mid \bot \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi,$$

where $p \in P$ and $i \in \mathcal{A}$. As usual, the intended interpretation of a formula $K_i\phi$ is "agent i knows ϕ". We also consider an extended language $\mathcal{L}_{KC}(P, \mathcal{A})$ obtained by adding formulas of the type $C\phi$ intended to express common knowledge of ϕ. The semantics of $\mathcal{L}_K(P, \mathcal{A})$ and $\mathcal{L}_{KC}(P, \mathcal{A})$ is defined as usual through Kripke structures, here called *epistemic models*.

DEFINITION 1 (EPISTEMIC MODELS). — *An* epistemic model *of the languages $\mathcal{L}_K(P, \mathcal{A})$ and $\mathcal{L}_{KC}(P, \mathcal{A})$ is a triple $\mathcal{M} = (W, R, V)$, where*

– *W is the* domain*, a finite set of* worlds *(often called* states *in the literature, but we will use the word "state" for a different purpose in this paper).*

– *$R : \mathcal{A} \to 2^{W \times W}$ assigns an* accessibility relation *(or* indistinguishability relation*) R_i to each agent $i \in \mathcal{A}$. All accessibility relations are equivalence relations.*

– *$V : P \to 2^W$ assigns a set of worlds to each atomic proposition; this is the* valuation *of that variable.*

The domain W of an epistemic model $\mathcal{M} = (W, R, V)$ is often denoted $D(\mathcal{M})$. The requirement of the accessibility relations being equivalence relations ensures that the modal operators K_i capture knowledge. Most of what we do in this paper would work equally well with weaker or no conditions on the accessibility relations, e.g."belief" or even weaker notions, but for simplicity we stick to knowledge in this paper.

DEFINITION 2 (TRUTH IN AN EPISTEMIC MODEL). — *Let an epistemic model $\mathcal{M} = (W, R, V)$ of $\mathcal{L}_{KC}(P, \mathcal{A})$ be given. Let $i \in \mathcal{A}$, $w \in W$ and $\phi, \psi \in \mathcal{L}_{KC}(P, \mathcal{A})$.*

$$\begin{array}{lll}
\mathcal{M}, w \models \top & & \textit{always} \\
\mathcal{M}, w \models \bot & & \textit{never} \\
\mathcal{M}, w \models p & \textit{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\phi & \textit{iff} & \mathcal{M}, w \not\models \phi \\
\mathcal{M}, w \models \phi \wedge \psi & \textit{iff} & \mathcal{M}, w \models \phi \textit{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models K_i\phi & \textit{iff} & \textit{for all } v \in W, \textit{ if } wR_iv \textit{ then } \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models C\phi & \textit{iff} & \textit{for all } v \in W, \textit{ if } w(\cup_{j \in \mathcal{A}} R_j)^*v \textit{ then } \mathcal{M}, v \models \phi
\end{array}$$

where R^* is the transitive closure of R.

If $\mathcal{M}, v \models \phi$ holds for all epistemic models $\mathcal{M} = (W, R, V)$ and all $w \in W$, the formula ϕ is said to be valid*, denoted $\models \phi$.*

A pair $(\mathcal{M}, w)$ consisting of an epistemic model $\mathcal{M}$ and a world $w \in D(\mathcal{M})$ is often called an *epistemic state* (or *pointed epistemic model*). In an epistemic state $(\mathcal{M}, w)$, w denotes the *actual world*. Epistemic states provide a model of the world from an external point of view, where the modeler is assumed to be an omniscient and external observer of the epistemic situation (Aucher, 2010). Thus, the modeler knows which is the actual world. In this paper, we also wish to be able to represent an internal point of view, where the modeler is one of the agents represented in the epistemic model. For this purpose, we distinguish between *global (epistemic) states* representing the external view of the world and pointing out the actual state of affairs, and *local (epistemic) states* representing individual agents' view of the world. This is related to the distinction made in e.g. (Aucher, 2010; Fagin *et al.*, 1995).

DEFINITION 3 (LOCAL AND GLOBAL (EPISTEMIC) STATES). — *A pair $(\mathcal{M}, W_d)$ consisting of an epistemic model $\mathcal{M} = (W, R, V)$ of $\mathcal{L}_{KC}(P, \mathcal{A})$ and a non-empty set of* designated *worlds $W_d \subseteq W$ is called an* epistemic state *or simply a* state *(of $\mathcal{L}_{KC}(P, \mathcal{A})$). If W_d is a singleton, the state is called* global*. If W_d is closed under R_i, where $i \in \mathcal{A}$, it is called a* local state *for agent i. In general, a* local state *is any pair $(\mathcal{M}, W_d)$ which is the local state of some agent. Given a global state $(\mathcal{M}, \{w\})$, the* associated local state *of agent i is $(\mathcal{M}, \{v \mid wR_iv\})$. States $(\mathcal{M}, W_d)$ in which the domain of $\mathcal{M}$ is a singleton are called* atomic states.

Note that in a local state $(\mathcal{M}, W_d)$ for agent i, it is possible to have a pair of nodes $w, v \in W_d$ with $(w, v) \notin R_i$. We will later use this to model "plan-time indistinguishability" whereas if $(w, v) \in R_i$ it models "run-time indistinguishability". Consider a local state $s = (\mathcal{M}, W_d)$ containing only w and v, and that these are plan-time indistinguishable. This means, that while the agent is planning, it does not know whether the actual world is w or v. However, when the plan is being executed and the agent actually achieves the state of the world represented by s, it will know which of w and v is the actual world. In other words, the agent knows, that while it does not yet know the actual world, it will come to know this once the plan is being carried out. If w and v are run-time indistinguishable, then the agent is unable to distinguish them, both while planning and when carrying out the plan. These concepts will be elaborated on later, particularly in Section 6.

An alternative way to define (local) states would be to introduce an additional accessibility relation R_d and an additional world w_0 s.t. $w_0 R_d w$ iff w belongs to the set of designated worlds. In this way (local) states would become ordinary pointed models of the form $(\mathcal{M}, w_0)$. However, we stick to the definition above, as it makes some of the following definitions and constructions simpler. The only disadvantage is that one has to be a bit more careful in defining bisimulations on states. Here is the definition.

DEFINITION 4 (BISIMULATIONS BETWEEN (EPISTEMIC) STATES). — *A* bisimulation *between states* $((W, R, V), W_d)$ *and* $((W', R', V'), W'_d)$ *is a non-empty binary relation* $B \subseteq W \times W'$ *which is an ordinary bisimulation between* (W, R, V) *and* (W', R', V') *and which furthermore satisfies that the domain of* B *extends* W_d *and the image of* W_d *under* B *is* W'_d*.*

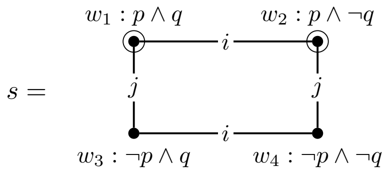
Note than when W_d and W'_d are singletons, this definition reduces to the ordinary definition of a bisimulation between pointed models. We can then, as usual, define the *bisimulation contraction* of a state as the quotient structure of the union of all autobisimulations (see e.g. (Blackburn *et al.*, 2001) for details).

DEFINITION 5 (TRUTH IN AN (EPISTEMIC) STATE). — *Let* $(\mathcal{M}, W_d)$ *be an epistemic state of* $\mathcal{L}_{KC}(P, \mathcal{A})$ *and* ϕ *a formula of* $\mathcal{L}_{KC}(P, \mathcal{A})$*. Then truth of* ϕ *in* $(\mathcal{M}, W_d)$ *is defined as follows:*

$$(\mathcal{M}, W_d) \models \phi \quad \textit{iff} \quad \mathcal{M}, w \models \phi \textit{ for all } w \in W_d$$

Note that for all local states $(\mathcal{M}, W_d)$ of some agent i and all formulas ϕ, the following holds: $(\mathcal{M}, W_d) \models K_i\phi \Leftrightarrow (\mathcal{M}, W_d) \models \phi$, as W_d is closed under R_i, and since R_i is reflexive. The property reflects the fact that a local state of some agent gives that agent's internal view of the world. When $(\mathcal{M}, W_d) \models \phi$ – or, equivalently, $(\mathcal{M}, W_d) \models K_i\phi$ – we say that agent i *knows* ϕ in $(\mathcal{M}, W_d)$. If $(\mathcal{M}, W_d) \models K_i\phi \vee K_i\neg\phi$, we say that agent i *knows whether* ϕ holds.

EXAMPLE 6. — Consider the following state:



The reflexive loops at each of the worlds have been left out for visual simplicity, which will be the case in the remainder of this paper. More generally, we will always only the show the *reflexive transitive reduction* of a state, that is, the one in which each accessibility relation R_i has been replaced by the minimal relation R'_i having the same reflexive transitive closure as R_i. The symbol $\odot$ marks designated worlds. Here, the designated worlds are w_1 and w_2. As the set of designated worlds is closed under R_i but not R_j, the state is a local state of agent i but not agent j, thus enabling its interpretation as i's view of the world. That both w_1 and w_2 are designated is due to agent i's inability to recognise which of these is the actual world. In the state, agent i knows that p holds, but doesn't know whether q holds. i does, however, know that j knows whether q holds. □

3. Event models and epistemic actions

From Dynamic Epistemic Logic (DEL), we take the concept *event model* (or *update model* or *action model*), see e.g. (van Ditmarsch *et al.*, 2008), for modeling the changes to epistemic states, brought about by the execution of actions. The exact relationship between event models and epistemic states in relation to planning will be clarified later.

DEFINITION 7 (EVENT MODELS). — *An* event model *for $\mathcal{L}_{KC}(P, \mathcal{A})$ is a quadruple $\mathcal{E} = (E, Q, pre, post)$, where*

– *E, the* domain*, is a finite non-empty set of* events*.*
– *$Q : \mathcal{A} \to 2^{E \times E}$ assigns an* accessibility relation *(or* indistinguishability relation*) to each agent $i \in \mathcal{A}$. All accessibility relations are equivalence relations.*
– *$pre : E \to \mathcal{L}_{KC}(P, \mathcal{A})$ assigns to each event a* precondition*.*
– *$post : E \to \mathcal{L}_{KC}(P, \mathcal{A})$ assigns to each event a* postcondition*. Postconditions are conjunctions of propositional literals, that is, conjunctions of atomic propositions and their negations (including $\top$ and $\bot$).*

The domain E of an event model $\mathcal{E} = (E, Q, pre, post)$ is denoted $D(\mathcal{E})$. The postcondition mapping is defined in a slightly non-standard way here. Usually, it is defined as a mapping $post' : E \to (P \to \mathcal{L}_{KC}(P, \mathcal{A}))$. As shown in (van Ditmarsch *et al.*, 2008), one can without loss of generality restrict to mappings of this type where $post'(e)(p)$ is always either $\top$, $\bot$ or p itself. Any such mapping gives rise to a mapping $post : E \to \mathcal{L}_{KC}(P, \mathcal{A})$ of the type defined above by letting:

$$post(e) = \left(\bigwedge_{post'(e)(p)=\top} p \right) \wedge \left(\bigwedge_{post'(e)(p)=\bot} \neg p \right).$$

One of the advantages of this formulation of the postcondition mapping is that it links more naturally with classical planning, as will be seen further below.

DEFINITION 8 (LOCAL AND GLOBAL (EPISTEMIC) ACTIONS). — *A pair $(\mathcal{E}, E_d)$ consisting of an event model $\mathcal{E} = (E, Q, pre, post)$ of $\mathcal{L}_{KC}(P, \mathcal{A})$ and a non-empty set of* designated *events $E_d \subseteq E$ is called an* epistemic action *or simply an* action *(of $\mathcal{L}_{KC}(P, \mathcal{A})$). If E_d is a singleton, the action is called* global*. If E_d is closed under*

R_i, where $i \in \mathcal{A}$, it is called a local action *for agent i. In general, a* local action *is any pair $(\mathcal{E}, E_d)$ which is the local state of some agent. Given a global action $(\mathcal{E}, \{e\})$, the* associated local action *of agent i is $(\mathcal{E}, \{f \mid eQ_i f\}$. Actions $(\mathcal{E}, E_d)$ in which the domain of $\mathcal{E}$ is a singleton are called* atomic actions.

The literature sometimes refer to our global actions as *pointed updates* and our non-global actions as *multi-pointed updates*, see e.g. (Sadzik, 2006). Beware that even though we sometimes refer to our actions as *epistemic* actions, they also allow the possibility of expressing factual (ontic) change via the postcondition mapping.

DEFINITION 9 (PRODUCT UPDATE OF A STATE WITH AN ACTION). — *Given are a state $(\mathcal{M}, W_d)$ and an action $(\mathcal{E}, E_d)$, where $\mathcal{M} = (W, R, V)$ and $\mathcal{E} = (E, Q, pre, post)$. The* product update *of the state $(\mathcal{M}, W_d)$ with the action $(\mathcal{E}, E_d)$ is defined as the state $(\mathcal{M}, W_d) \otimes (\mathcal{E}, E_d) = ((W', R', V'), W'_d)$, where*

– $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$
– $R'_i = \{((w, e), (v, f)) \in W' \times W' \mid wR_i v \text{ and } eQ_i f\}$
– $V'(p) = (\{(w, e) \in W' \mid \mathcal{M}, w \models p\} - \{(w, e) \in W' \mid post(e) \models \neg p\}) \cup \{(w, e) \in W' \mid post(e) \models p\}$
– $W'_d = \{(w, e) \in W' \mid w \in W_d \text{ and } e \in E_d\}$

DEFINITION 10 (APPLICABILITY OF AN ACTION IN A STATE). — *Given are a state $(\mathcal{M}, W_d)$ and an action $(\mathcal{E}, E_d)$. The action $(\mathcal{E}, E_d)$ is said to be* applicable *in the local state $(\mathcal{M}, W_d)$ if the following holds: For each world $w \in W_d$ there is a least one event $e \in E_d$ such that $\mathcal{M}, w \models pre(e)$.*

The intuition behind this definition is the following. First, note that if both the state and the action are global, that is of the form $W_d = \{w\}$ and $E_d = \{e\}$, this reduces to the condition that the precondition of the designated event e holds in the designated world w ($\mathcal{M}, w \models pre(e)$). This corresponds to the condition of "possibility" introduced with a similar purpose in (Löwe *et al.*, 2010). The point is, that the designated world w denotes the *current world* and the designated event e denotes the event that *actually* takes place, so the condition simply ensures that the precondition of the event that takes place is satisfied in the current world. If not, the pointed set W'_d of the product update $(\mathcal{M}, \{w\}) \otimes (\mathcal{E}, \{e\})$ would be empty.

Now consider the case of local states and actions. When we update a local state $(\mathcal{M}, W_d)$ of agent i with a local action $(\mathcal{E}, E_d)$ of the same agent, the local action is assumed to present agent i's view on what the action will bring about (it could be an action executed by i himself, but could also be an action executed by some other agent, or a joint action of several agents). The condition of Definition 10 then has the following meaning: For each of the worlds that agent i considers possible, the action specifies at least one applicable event that i considers possible.

LEMMA 11. — *If a local action $(\mathcal{E}, E_d)$ of an agent i is applicable in a local state $(\mathcal{M}, W_d)$ of the same agent, then the product update $(\mathcal{M}, W_d) \otimes (\mathcal{E}, E_d)$ is again a local state of i.*

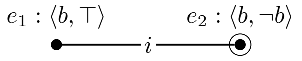
We leave the proof as an (easy) exercise for the reader.

EXAMPLE 12. — The following example is inspired by the *Sally-Ann test* used in cognitive psychology to test whether children possess a so-called *theory of mind* (Wimmer *et al.*, 1983). There are three agents, Sally (denoted by i), Ann (denoted by j) and an observer, the child (denoted by k). Sally has a basket and Ann has a box. There is a marble, which can either be in the basket or in the box. We use b to denote the proposition "the marble is in the basket". In the initial situation, the marble is in the basket, and this is common knowledge. Thus the following local state s_0 describes all three agents' initial view of the world:

$$s_0 = \odot\, w_1 : b$$

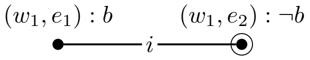
Now Sally (i) leaves the room, and in the meantime Ann (j) moves the marble to the box. The observer sees this, but Sally doesn't. However, it is common knowledge that when Sally leaves the room, Ann has the possibility of moving the marble to the box. The observer can represent the action taking place by the following local action a_1:

$$a_1 =$$


$e_1 : \langle b, \top \rangle$
$e_2 : \langle b, \neg b \rangle$
i

Labeling events by the pair $\langle \phi_1, \phi_2 \rangle$ means that the event has precondition ϕ_1 and postcondition ϕ_2. In the action a_1, event e_1 represents the possibility that Ann doesn't move the marble, and event e_2 represents the possibility that she does. Since Sally (i) has left the room, she cannot distinguish these two events, which is represented by the two events being connected by an i-relation. There is, however, no j- or k-relation connecting the two, since both Ann and the observer can distinguish between the marble being moved and not being moved. The designated event is e_2, since the observer sees the marble being moved. Taking the product update of s_0 with a_1, we then obtain the observers updated view of the world after the action has taken place:

$$s_0 \otimes a_1 =$$



We can see that the observer now knows that Sally (i) doesn't know where the marble is (doesn't know whether b or $\neg b$ holds). It has been shown in (Wimmer *et al.*, 1983) that children under the age of 4, and autistic children in general, will—when playing the role of the observer—conclude that Sally knows that the marble is now in the box ($\neg b$). □

With epistemic actions and states defined, we now show how these are employed in a planning context.

4. Epistemic planning domains and problems

Following (Ghallab *et al.*, 2004), any classical planning domain can be represented as a *restricted state-transition system* $\Sigma = (S, A, \gamma)$, where

– S is a finite or recursively enumerable set of *states*.

– A is a finite set of *actions*.

– $\gamma : S \times A \hookrightarrow S$ is a computable *state-transition function*. The state-transition function is partial, that is, for any $(s, a) \in S \times A$, either $\gamma(s, a)$ is undefined or $\gamma(s, a) \in S$.

A *classical planning problem* is then represented as a triple (Σ, s_0, S_g), where

– Σ is a *restricted state-transition system*.

– s_0 is the *initial state*, a member of S.

– S_g is the set of *goal states*, a subset of S.

A *solution* to a classical planning problem (Σ, s_0, S_g) is a finite sequence of actions (a *plan*) $a_1, a_2, \ldots, a_n$ such that

$$\gamma(\gamma(\ldots\gamma(\gamma(s_0, a_1), a_2), \ldots, a_{n-1}), a_n) \in S_g.$$

Note that finding solutions to classical planning problems is always at least semi-decidable: given a planning problem, we can compute its *state space* (the space of states reachable by a sequence of actions applied to the initial state) in a breadth-first manner, and if one of the goal states is reachable, we will eventually find it. Next is the definition of epistemic planning domains, which are special cases of classical planning domains.

DEFINITION 13 (EPISTEMIC PLANNING DOMAINS). — *Given are a finite set P of atomic propositions and a finite set $\mathcal{A}$ of agents. An* epistemic planning domain *on $(P, \mathcal{A})$ is a restricted state-transition system $\Sigma = (S, A, \gamma)$, where*

– *S is a finite or recursively enumerable set of epistemic states of $\mathcal{L}_{KC}(P, \mathcal{A})$.*

– *A is a finite set of actions of $\mathcal{L}_{KC}(P, \mathcal{A})$.*

– *γ is defined by:*

$$\gamma(s, a) = \begin{cases} s \otimes a & \textit{if } a \textit{ is applicable in } s \\ \text{undefined} & \textit{otherwise} \end{cases}$$

If all states and actions are from $\mathcal{L}_K(P, \mathcal{A})$ it is called an epistemic planning domain without common knowledge. *If $|\mathcal{A}| = 1$ it is called a* single-agent *epistemic planning domain.*

DEFINITION 14 (EPISTEMIC PLANNING PROBLEMS). — *An* epistemic planning problem *is a triple (Σ, s_0, ϕ_g), where*

– *$\Sigma = (S, A, \gamma)$ is an epistemic planning domain on $(P, \mathcal{A})$.*

– *s_0, the* initial state*, is a member of S.*

– *ϕ_g is a formula in $\mathcal{L}_{KC}(P, \mathcal{A})$ called a* goal formula*. The set of* goal states *is $S_g = \{s \in S \mid s \models \phi_g\}$.*

If all states, actions, and formulas are from $\mathcal{L}_K(P, \mathcal{A})$ it is called an epistemic planning problem without common knowledge. *If $|\mathcal{A}| = 1$ it is called a* single-agent *epistemic planning problem.*

Epistemic planning problems are special cases of classical planning problems. A *solution* to an epistemic planning is thus, according to the definition above, a sequence of actions $a_1, a_2, \dots, a_n$ s.t. $\gamma(\gamma(\dots\gamma(\gamma(s_0, a_1), a_2), \dots, a_{n-1}), a_n) \in S_g$, that is, s.t. $s_0 \otimes a_1 \otimes a_2 \otimes \cdots \otimes a_n \models \phi_g$. As noted, finding solutions is at least semi-decidable. A further look at the complexity of epistemic planning is found in later in the paper. Before examining (in the next section) some of the different types of actions that can be defined in epistemic planning problems, we briefly touch upon the relation between epistemic planning and Dynamic Epistemic Logic (DEL).

Note that in our framework for epistemic planning, our only take away from DEL is event models and product updates. We do not make use of the full DEL language, that is, epistemic logic extended with action modalities. Action modalities are used in DEL to express the logical consequences of performing actions encoded as event models. This means that we have a logical language in which it is possible to represent and reason about actions and their dynamics. In classical planning, on the other hand, the underlying logical language only describes static states of affairs. The dynamics is instead captured in a meta-language. This meta-language describes actions in terms of how they modify state descriptions given as formulas of the object language (cf. e.g. STRIPS). In other words, in classical planning the object language describing states is completely separate from the meta-language describing actions. We have here taken a similar approach, where the object language for describing states is simply standard epistemic logic, and the meta-language for describing actions is event models.

It would of course also be possible in our framework to include action modalities in the object language, that is, make it the full language of DEL. This would allow us to include formulas with action modalities in pre- and post-conditions of actions as well as in goal formulas. It would thus allow us to, for instance, express goals such as "achieve a state in which it is (im)possible for agent j to perform an action that will result in ϕ." If we were to allow goals to include statements about actions and their consequences, we would also like to be able to state goals such as: "achieve a state in which it is (im)possible for agent j to perform *any sequence of actions* that will result in ϕ." This is not possible in standard DEL, but requires us to introduce iteration of modalities. We leave this for future work.

5. Action types

An action $(\mathcal{E}, E_d)$ is called *purely epistemic* if for all events e in $D(\mathcal{E})$, the post-condition of e is implied by the precondition, that is, $\models pre(e) \rightarrow post(e)$. A purely epistemic action is one that does not make any factual (ontic) changes. An important example of such actions are *public announcements*, which are the purely epistemic atomic actions. Actions that are not purely epistemic are called *ontic*. Along an orthogonal axis, we can distinguish between observable and non-observable actions. An action $(\mathcal{E}, E_d)$ is called *fully observable* or *public*, if all the accessibility relations of $\mathcal{E}$ are identities (that is, no two distinct events are connected). If an action is not fully observable, it is called *partially observable*.

Let there be given a partially observable action $((E, Q, pre, post), E_d)$ and a group of agents $G \subseteq \mathcal{A}$. If for each $i \in G$, the accessibility relation Q_i is the identity, then the action is said to be *group observable* by G. If, in addition, for each $j \in \mathcal{A} - G$ the accessibility relation Q_j is the universal relation, then it is said to be *group observable by G alone*. An action is said to be *privately observable* by an agent i, if the action is group observable by $\{i\}$ alone. Note, that agents $j \in \mathcal{A} - G$ will know that *something* has happened, though not precisely what.

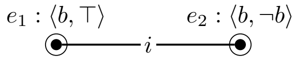
An action $((E, Q, pre, post), E_d)$ is called *globally deterministic* if all preconditions are mutually inconsistent, that is, $\models pre(e) \wedge pre(f) \rightarrow \bot$ for all distinct $e, f \in E$, in other words, only one event is possible for each world. It is called a *sensing action* if:

- it is purely epistemic
- it is globally deterministic
- its preconditions cover the logical space, that is, $\models \bigvee_{e \in E} pre(e) \leftrightarrow \top$.

Sensing actions are called *answers* in (Gerbrandy, 2007), but the word "sensing" is better in line with the taxonomy of the automated planning literature.

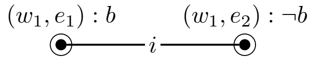
EXAMPLE 15. — Consider again the Sally-Ann example (Example 12), but now from the perspective of Sally. From her perspective, the event that takes place while she is out of the room is represented by the following local action:

$a'_1 =$


$e_1 : \langle b, \top \rangle$
$e_2 : \langle b, \neg b \rangle$
i

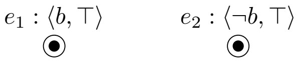
This action is *group observable by* $\{j, k\}$ *alone* (Ann and the observer). The product update of s_0 with a'_1 gives Sally's view on the world after the event has taken place:

$s_0 \otimes a'_1 =$



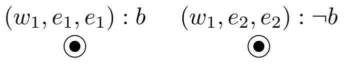
Sally now no longer knows whether the marble is in the basket or not, that is, $s_0 \otimes a'_1 \models \neg K_i b \wedge \neg K_i \neg b$. Sally might after this consider the action of entering the room again and look into the basket, where the marble used to be. This is a *sensing action*, where Sally will get to know whether the marble is in the basket or not. The sensing action looks as follows, again from the viewpoint of Sally:

$a'_2 =$


$e_1 : \langle b, \top \rangle$
$e_2 : \langle \neg b, \top \rangle$

Note that both events are still designated. This is because Sally does not know the outcome of the sensing action at the time she plans the action (see a more thorough discussion of this in the following section). Now, updating Sally's local state $s_0 \otimes a'_1$ with this sensing action, we get:

$s_0 \otimes a'_1 \otimes a'_2 =$



Now there is no longer an i-edge between the two worlds, because it represents the situation after Sally has been sensing which of the two holds. We now have that Sally knows whether the marble is in the basket, that is, $s_0 \otimes a_1' \otimes a_2' \models K_i b \vee K_i \neg b$. $\square$

In the next section, we will look at how epistemic planning domains generalise some well-known types of planning domains studied in automated planning.

6. Propositional planning and partial observability in single-agent domains

Following (Ghallab *et al.*, 2004), a *propositional planning domain* (or *set-theoretic planning domain*) on a finite set P of atomic propositions is a restricted state-transition system $\Sigma = (S, A, \gamma)$ satisfying:

– $S = 2^P$.
– A is a set of pairs $a = (\mathit{precond}(a), \mathit{effects}(a))$, where both $\mathit{precond}(a)$ and $\mathit{effects}(a)$ are finite sets of literals over P. An action a is said to be *applicable* in a state if $\mathit{precond}^+(a) \subseteq s$ and $\mathit{precond}^-(a) \cap s = \emptyset$.[1]
– γ is defined by:

$$\gamma(s,a) = \begin{cases} (s - \mathit{effects}^-(a)) \cup \mathit{effects}^+(a) & \text{if } a \text{ is applicable in } s \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that propositional planning is decidable, as the set of states is finite. Every propositional planning domain $\Sigma = (S, A, \gamma)$ is equivalent to an epistemic planning domain $\Sigma' = (S', A', \gamma')$ defined as follows:

– S' is the set of atomic states of $\mathcal{L}_{KC}(P, \mathcal{A})$ where $|\mathcal{A}| = 1$.
– A' is the set that for each $a \in A$ contains an atomic action $(E_a, Q_a, \mathit{pre}_a, \mathit{post}_a)$ given by $E_a = \{e\}$, $Q_a = \{(e,e)\}$, $\mathit{pre} = \bigwedge_{p \in \mathit{precond}^+(a)} p \;\wedge \bigwedge_{p \in \mathit{precond}^-(a)} \neg p$ and $\mathit{post} = \bigwedge_{p \in \mathit{effects}^+(a)} p \;\wedge \bigwedge_{p \in \mathit{effects}^-(a)} \neg p$.
– γ' is defined as above for general epistemic planning domains.

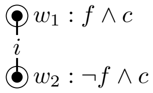
It is easy to check that Σ and Σ' are indeed equivalent. This shows that propositional planning domains are a special case of epistemic planning domains, and that the propositional planning domains can be precisely characterised as those epistemic planning domains where all states and actions are atomic, and where all actions have purely propositional preconditions. This should come as no surprise, but is still worth noting, as it clarifies the exact link between classical propositional planning and epistemic planning.

Epistemic planning domains also allow for a nice treatment of *partial observability*. Assume we are still in a single-agent domain, that is $|\mathcal{A}| = 1$. Let i denote the element of $\mathcal{A}$. Assume we have a local action $((E, Q, \mathit{pre}, \mathit{post}), E_d)$ of agent i. Let

[1] For any set of literals L, L^+ denotes the set of atoms in L and L^- denotes the set of atoms whose negations are in L.

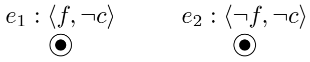
$e, e' \in E_d$. We say that e and e' are *runtime indistinguishable* if eQ_ie', otherwise they are called *runtime distinguishable* or *plan-time indistinguishable*. The point is this. Assume, for example, that the agent is facing a closed box which might either be full (denoted by f) or empty (denoted by $\neg f$), but he doesn't know which. Let c denote the proposition “the box is closed". Then his local state is this:

$s_0 =$



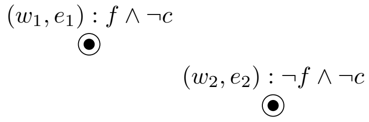
Now assume he considers the action of opening the box to see its content. At plan-time (when he is still computing the plan), he only knows that the effect of opening the box will be that he *either* learns f *or* learns $\neg f$, but not which. So the two outcomes are plan-time indistinguishable to him. However, at run-time when actually carrying out the action, he *will* know which of the two is the case. We can model this by the following local action:

openBox $=$


$e_1 : \langle f, \neg c\rangle$
$e_2 : \langle \neg f, \neg c\rangle$

Updating the state above with this action we then get:

$s_0 \otimes$ *openBox* $=$


$(w_1, e_1) : f \wedge \neg c$
$(w_2, e_2) : \neg f \wedge \neg c$

The state after the execution only differs from the state before by the substitution of $\neg c$ for c (opening the box) and the removal of the edge between the f-world and the $\neg f$-world. This means that after the action the agent will be able to distinguish between f and $\neg f$. However, as the agent at plan time still doesn't know which it will be, we need to keep both worlds in the set of distinguished worlds. This explains the need of the set of distinguished worlds, W_d, and the need of a special definition of bisimulation between states.

Constructing actions that combine runtime indistinguishable events with plan-time indistinguishable events allows us to model partial observability: if two possible outcomes (events) will be indistinguishable even when the action is performed at runtime (no observation), then they should be in the same Q_i equivalence class; if the two possible outcomes will be distinguishable when the action is performed (observable), then they should be in distinct Q_i equivalence classes. And, obviously, if the exact outcome is known already at plan-time, the action will contain only a single event representing this outcome. Note that this approach to partial observability is consistent with the definition of *fully observable actions* introduced earlier. According to this definition, a single-agent action is fully observable if and only if its accessibility relation is the identity, that is, all pairs of events are runtime distinguishable. The articles

(Bacchus *et al.*, 1998; Petrick *et al.*, 2002) argue in favour of a similar approach to partial observability.

As it can easily be seen, in the single-agent case, each local state can—modulo bisimulation—be uniquely described by a set of atomic actions and a description of which of the actions are runtime distinguishable. Thus it seems fair to say that epistemic planning with one agent captures *exactly* what is involved in propositional planning in (nondeterministic) domains with partial observability.

EXAMPLE 16. — Continuing the example with the agent and the box, consider the following action:

$$\textit{emptyBox} = \quad \underset{\odot}{e_1 : \langle f \wedge \neg c, \neg f \rangle} \qquad \underset{\odot}{e_2 : \langle \neg f, \top \rangle}$$

This is an action for emptying the box. Note that it distinguishes between two cases: one covering the case where the box is full and open, and another covering the case where it's already empty. Note that the action is only applicable when the agent knows that either the box is open or already empty. We now get:

$$s_0 \otimes \textit{openBox} \otimes \textit{emptyBox} = \quad \begin{array}{ll} \underset{\odot}{(w_1, e_1, e_1) : \neg f \wedge \neg c} & \\ & \underset{\odot}{(w_2, e_2, e_2) : \neg f \wedge \neg c} \end{array}$$

Thus a solution to the planning problem of satisfying the goal formula $\neg f$ given the initial state s_0 would be *openBox*, *EmptyBox*. Note that the branching that usually takes place when planning in partially observable domains like this is being internalised in the state descriptions. Note also that if we take the bisimulation contraction of the state $s_0 \otimes \textit{openBox} \otimes \textit{emptyBox}$, we get an atomic state. Thus if we want to plan further, e.g. close the box again, we can now work with atomic states. □

We will now prove that single-agent epistemic planning is decidable, that is, given any epistemic planning problem we can decide whether a plan exists or not. In the proof we actually show something slightly stronger, since we also show how to construct a plan if one exists.

THEOREM 17. — *Single-agent epistemic planning is decidable.*

PROOF. — Given any single-agent epistemic planning problem, we can perform a breadth-first exploration of the state space. However, after computing each new state, we make sure to replace it by its bisimulation contraction, which can be computed in linear time (Dovier *et al.*, 2001). Now it suffices to prove that when $|\mathcal{A}| = 1$, there are only finitely many distinct bisimulation minimal states of $\mathcal{L}_{KC}(P, \mathcal{A})$ (recall that P is always assumed to be finite). Consider first connected states of $\mathcal{L}_{KC}(P, \mathcal{A})$, that is, states with only one equivalence class. Since the accessibility relation is an equivalence relation, there can be no two worlds satisfying the same atomic propositions in a bisimulation minimal state (the two worlds would be bisimilar). Thus all bisimulation minimal connected states are substates of the following state (up to isomorphism):[2]

$$((2^P, 2^P \times 2^P, V), 2^P), \text{where } V(p) = \{w \mid p \in w\}.$$

There can obviously only be finitely many such substates (up to isomorphism). Now consider the case of non-connected states. Note that we can not immediately reduce these to connected states due to the way we defined bisimulations on states. In any case, each equivalence class in the state must again be a substate of the state defined above. Furthermore, there can be no two bisimilar equivalence classes, by bisimulation minimality. Thus, there can also only be finitely many bisimulation minimal non-connected states (up to isomorphism). This is the required conclusion. ■

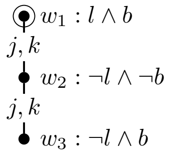
In this section we have only been considering the single-agent case, but obviously the multi-agent case is the most interesting, and, as we will see next, also far more challenging.

7. Multi-agent epistemic planning

We will start this section by giving an example of a multi-agent epistemic planning domain inspired by the well-known *Byzantine Agreement* problem (or *coordinated attack* problem) (Fagin *et al.*, 1995).

EXAMPLE 18. — There are three logicians, a philosopher (i), a computer scientist (j), and a mathematician (k). They work at the same university, so usually they go together in the same car, allowing them to discuss logic on the way. One day at work it happens that agent i suddenly recalls that he forgot to turn off the lights of the car, so that the battery is now flat. Let l denote the proposition “the lights are on" and let b denote the proposition “the battery is flat". A possible local state describing agent i’s internal view of the world immediately after having realised that the light were left on could then be:

$s_0 =$



Recall that, by convention, we only show the reflexive transitive reduction of the state, so there is an implicit j, k-edge from w_1 to w_3. In this local state, it is assumed to be common knowledge that j and k are still unaware of whether the lights are on or not. The fact that there is no world labelled $l \wedge \neg b$ means that it is also common knowledge that *if* the lights are on, *then* the battery is flat, that is, $l \rightarrow b$.

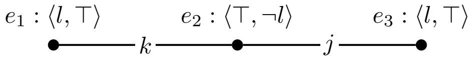
[2] A state $((W', R', V'), W'_d)$ is called a *substate* of a state $((W, R, V), W_d)$ if (W', R', V') is a submodel of (W, R, V) and $W'_d = W_d \cap W'$.

Assume further, to keep the example manageable (and because leaving the lights on happens more often than the three logicians would wish to think about), that it is common knowledge that from now on agent i can repeatedly only choose between the following three actions:

1) tell agent j that l:
2) go to the car and turn the lights off if they are still on;
3) tell agent k that l.

Actions 1 and 3 are both announcement of l, so they can both be expressed by the event $\bullet\ \langle l, \top \rangle$. Action 2 is an ontic action that can be expressed by the event $\bullet\ \langle \top, \neg l \rangle$. Agent k cannot distinguish 1 from 2 and agent j cannot distinguish 2 from 3. So we obtain an event model $\mathcal{E}$ looking like this:

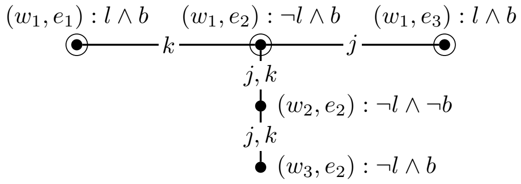
$\mathcal{E} =$


$e_1 : \langle l, \top \rangle$
$e_2 : \langle \top, \neg l \rangle$
$e_3 : \langle l, \top \rangle$
k
j

The three local actions available to agent i are then $(\mathcal{E}, \{e_1\})$, $(\mathcal{E}, \{e_2\})$ and $(\mathcal{E}, \{e_3\})$ (note that agent i has full observability). The three possible actions are modelled by the same event model, only differing in the designated set. This is often the case in multi-agent epistemic planning domains, and it means that the branching that usually takes place in the search for a plan is here internalised in a single epistemic action, and the branching factor of the plan search becomes 1 (if we make sure to label worlds with their update history to be able to keep track of the designated sets). Of course, what you pay for this is that the epistemic models might grow exponentially in size as you move down through the state space, but this is essentially no different from classical planning, where the number of reachable states can grow exponentially with the depth.

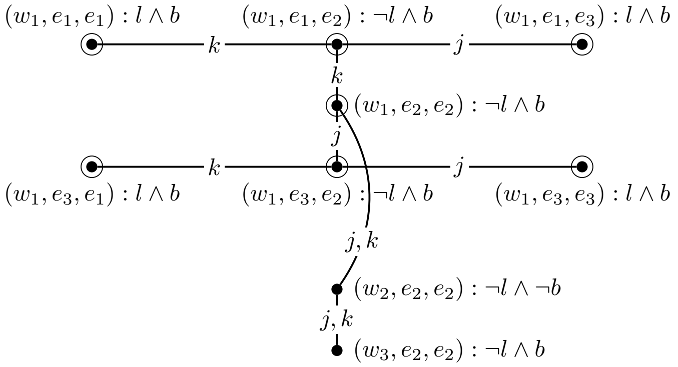
Now consider the product update of the initial state s_0 with the local action $a = (\mathcal{E}, \{e_1, e_2, e_3\})$, where we have chosen all three events as designated to postpone the decision of which to pick (corresponds to nondeterministic, but observable, choice, cf. Section 5):

$s_0 \otimes a =$


$(w_1, e_1) : l \wedge b$
$(w_1, e_2) : \neg l \wedge b$
$(w_1, e_3) : l \wedge b$
k
j
j, k
$(w_2, e_2) : \neg l \wedge \neg b$
j, k
$(w_3, e_2) : \neg l \wedge b$

It can be seen that if agent i e.g. chooses to do e_1 (tell l to j), then afterward $K_j b$ and $\neg K_k b$ (since these formulas hold in the world (w_1, e_1)). Now consider a second update with the action a:

$s \otimes a \otimes a =$


$(w_1, e_1, e_1) : l \wedge b$
$(w_1, e_1, e_2) : \neg l \wedge b$
$(w_1, e_1, e_3) : l \wedge b$
k
j
k
$(w_1, e_2, e_2) : \neg l \wedge b$
j
k
j
$(w_1, e_3, e_1) : l \wedge b$
$(w_1, e_3, e_2) : \neg l \wedge b$
$(w_1, e_3, e_3) : l \wedge b$
j, k
$(w_2, e_2, e_2) : \neg l \wedge \neg b$
j, k
$(w_3, e_2, e_2) : \neg l \wedge b$

If agent i first chooses e_1 and then e_3, the designated world (actual world) will become (w_1, e_1, e_3). We here have $K_k K_j b$ but not $K_j K_k K_j b$. It might at first seem intuitively puzzling that k knows that j knows the battery to be flat after i having performed only the action sequence e_1, e_3 (tell l to j, tell l to k). To explain the intuition, first note that by choice of event model, whenever agent i performs an action, all three agents will "know" that one of e_1, e_2 or e_3 has happened, but not necessarily which. Thus after i having performed first e_1 and then e_3, we can intuitively think of agent k as being able to perform the following line of reasoning: "Agent i's second action was to tell me l. Thus the lights must still be on. Therefore agent i's first action can not have been to turn off the lights. Since his first action wasn't to let me know about the lights either, his first action must have been to let agent j know that the lights are on. From this, j must have been able to conclude that the battery is flat." This reasoning leads k to conclude that j knows b (no sequence of actions can change the truth value of b). This provides the informal intuition behind why $K_k K_j b$ holds at (w_1, e_1, e_3).

It can easily be shown that in general in the world

$$(w_1, \overbrace{e_1, e_3, e_1, e_3, \ldots, e_1, e_3}^{2(n+1)})$$

of the state $s_0 \otimes a^{2(n+1)}$ we have $(K_k K_j)^{n+1} b$ but not $K_j (K_k K_j)^{n+1} b$. It's a bit like an inverse Muddy Children puzzle: instead of iteratively decreasing the depth of the agents' uncertainty, it is iteratively increased. This is similar to the situation obtained in the Byzantine Agreement problem. From this we can infer that there is no upper bound on the size of the models in the chain $s_0 \otimes a, s_0 \otimes a^2, s_0 \otimes a^3, \ldots$, not even if we take the bisimulation contractions of the states. Similar to Byzantine Agreement, it is also easy to infer that no matter which sequence of choices agent i makes, it will never become common knowledge that the battery is flat (there will always be exactly one world in which $\neg l \wedge \neg b$ holds, and this world is accessible from all other worlds by some path). □

The example we just gave made use of ontic actions. However, in (Sadzik, 2006) it is shown that even allowing only *purely epistemic* actions with propositional preconditions, we can still get iterated updates of arbitrary size (using a variant of the coordinated attack problem). From the fact that in multi-agent epistemic planning problems there is in general no upper bound on the size of the reachable epistemic states, one might fear that planning is not even decidable in the general case. Indeed, this is exactly the case, as we will now show.

THEOREM 19. — *Multi-agent epistemic planning is undecidable (even without common knowledge).*

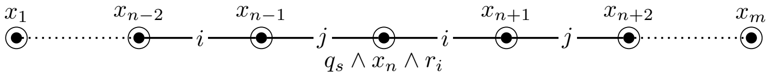
PROOF. — Undecidability here means that there is no decision procedure that for arbitrary multi-agent epistemic planning problems can determine whether a solution exists or not. We give the proof by showing that for any Turing machine M we can construct an epistemic planning problem P_M that has a solution if and only if M halts. As the halting problem is undecidable, so is epistemic planning. The underlying idea is this. Given any Turing machine M, we can encode its configurations (state, tape content and head position) as epistemic models—models containing exactly one world per non-blank tape cell. Furthermore, we can encode the possible transitions of M as epistemic actions. In this way, we achieve that any run of M can be simulated by a sequence of epistemic actions applied to the epistemic state representing the initial configuration of M. Suppose M has only a single halting state which we represent in the epistemic language by a special propositional symbol q_f. We can then conclude that M halts if and only if there is a sequence of epistemic actions leading from the (representation of the) initial epistemic state to an epistemic state in which q_f holds in one of the worlds. In this way, the Turing machine halts if and only if there is a solution to the planning problem in which the goal is that q_f should hold in one of the worlds. This gives us the required planning problem P_M that has a solution if and only if M halts.

We now proceed with the details. Let there be given a deterministic Turing machine M with two-way infinite tape, and states $q_0, q_1, \dots, q_f$, where q_0 is the initial state and q_f is the (only) halting state. The set of tape symbols is some finite set Γ including a blank symbol, b. We will now show how to construct an epistemic planning problem in which the Turing machine's configurations (state, tape content and head position) are encoded as epistemic states, and the transitions of the Turing machine are encoded as epistemic actions. First we will make use of common knowledge, but later we will show how the role of common knowledge can be replaced by the introduction of an additional agent. We build the planning problem on the language $\mathcal{L}_{KC}(\{q_0, \dots, q_f\} \cup \Gamma \cup \{r_i, r_j\}, \{i, j\})$.

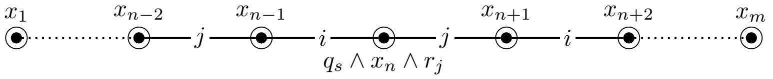
Assume we are given a configuration of M with instantaneous description (ID)

$$x_1 \cdots x_{n-2} x_{n-1} q_s x_n x_{n+1} \cdots x_m,$$

where $q_s \in \{q_0, \dots, q_f\}$ and all $x_i \in \Gamma$ (see (Hopcroft *et al.*, 2006) for details on Turing machines and instantaneous descriptions). Then this instantaneous description will be encoded as either of the following local states:



(1)



(2)

We call these local states the states *representing* the ID. There is exactly one world to represent each of the non-blank tape cells of the Turing machine, and this world is labelled by the symbol representing the content of the cell. In addition, the world representing the current tape cell is labelled by two additional atomic propositions: the name of the current state (q_s) and either the proposition r_i or the proposition r_j. The purpose of the propositions r_i and r_j will be explained in a moment. First note the alternation of i- and j-edges in these models. This is to ensure that the local states represent linear structures where each world has exactly one left and one right neighbour. If instead all edges were i-edges, all pairs of worlds would be each others neighbours, as all accessibility relations are assumed to be equivalence relations. Thus, if all edges were i-edges, we would be representing a set rather than a linear structure. The linear structure is required, since this is the only way we can encode the tape of a Turing machine. This also indicates why the current proof wouldn't work in the single-agent case, at least as long as we insist on using only equivalence classes, that is, insist on representing *knowledge*.

Now back to the propositions r_i and r_j. The purpose of r_i and r_j is to mark which indistinguishability relation (either i or j) will lead to the tape cell to the right of the current one (the "next" tape cell). If r_i holds at the world representing the cell currently scanned, it means that the tape cell to the right is represented by the neighbouring world reached by following the i-edge—and vice versa for r_j. Since in 1, r_i holds at the world representing the current tape cell, it means that the world representing the tape cell to the right is the one labelled x_{n+1}. If we replaced r_i by r_j in 1, it would correspond to changing the direction of the tape, and the right neighbour would instead become x_{n-1}.

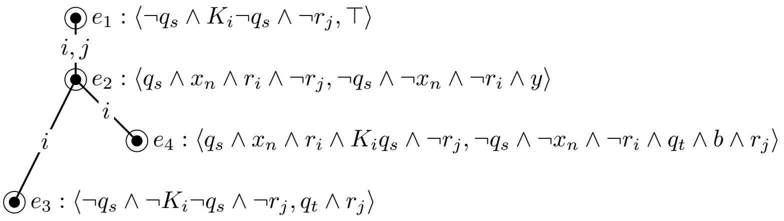
The initial configuration of M (empty tape) will be represented by the singleton local state

$$s_0 = \odot\, q_0 \land b \land r_i$$

This will be the initial state of our planning problem P_M. In the planning problem, we put two (symmetric) local actions for each of the transitions of the Turing machine. We will only show the local actions for transitions of the form

$$\delta(q_s, x_n) = (q_t, y, R), \text{where } x_n \neq y, \tag{3}$$

as transitions of the form $\delta(q_s, x_n) = (q_t, y, L)$ and transitions with $x_n = y$ can be handled similarly. There are two local actions corresponding to (3), where one is obtained from the other by interchanging i with j everywhere. We thus only show one of the two:


$e_1 : \langle \neg q_s \wedge K_i \neg q_s \wedge \neg r_j, \top \rangle$
i, j
$e_2 : \langle q_s \wedge x_n \wedge r_i \wedge \neg r_j, \neg q_s \wedge \neg x_n \wedge \neg r_i \wedge y \rangle$
i
i
$e_4 : \langle q_s \wedge x_n \wedge r_i \wedge K_i q_s \wedge \neg r_j, \neg q_s \wedge \neg x_n \wedge \neg r_i \wedge q_t \wedge b \wedge r_j \rangle$
$e_3 : \langle \neg q_s \wedge \neg K_i \neg q_s \wedge \neg r_j, q_t \wedge r_j \rangle$

(4)

We call these local actions the actions *representing* the transitions of the Turing machine. Suppose M can perform a move from an instantaneous description ID_1 to an instantaneous description ID_2, and let a be the local action representing the transition used in the move. The point is now that if s is a local state representing ID_1, then the product update $s \otimes a$ will be representing ID_2. Before providing the details, we will try to explain the intuition behind the construction of the local actions, and how they can simulate the moves of the Turing machine.

Let a denote an action of the form shown above, representing a transition of type (3). Let s denote a local state representing an instantaneous description of M, that is, s is on the form (1) or (2). Suppose a is applicable in the local state s. Then, by the applicability condition (Definition 10), s can only be of the form (1), as none of the events of a have preconditions that satisfy r_j. Now consider what happens when a is applied to s, that is, when we form the product update $s \otimes a$. We denote the world of s in which q_s holds by w_c. The world w_c represents the current tape cell of the Turing machine (before the update). We now consider how the events $e_1, \ldots, e_4$ of a affect the product update.

Event e_1 has its precondition satisfied in all worlds of s except w_c (because of the conjunct $\neg q_s$) and its right neighbour (because of the conjunct $K_i \neg q_s$). Since e_1 has an empty postcondition, this implies that in $s \otimes a$ all worlds of s except w_c and its right neighbour will be kept unchanged (paired with e_1). In other words, the tape excluding the current cell and its right neighbour remain the same after the update, as it should.

Event e_2 has its precondition satisfied in w_c and this world only. Its postcondition deletes q_s, x, and r_i, and instead adds y. Thus, event e_2 makes sure to change the symbol at the current tape cell from x to y, and remove the head from this cell.

Event e_3 has its precondition satisfied in the right neighbour of w_c, if such a right neighbour exists in s. In case it exists, q_t and r_j will be added as conjuncts to it. Thus e_3 makes sure to place the head at the right neighbour of the previous current cell, and to update the state from q_s to q_t.

In case the right neighbour of w_c doesn't exist, the action a makes sure to construct such a right neighbour. This is done via the event e_4. In case w_c has no right neighbour, the precondition of e_4 will be satisified in w_c. This implies that the product update $s \otimes a$ will contain both a world (w_c, e_2) and a world (w_c, e_4). The first of these is the "updated version" of w_c, whereas (w_c, e_4) is a "new" world. This new world is accessible from (w_c, e_2) by an i-edge. It is the new right neighbour of w_c. The postcondition of e_4 makes sure that $q_t \wedge b \wedge r_j$ will hold in this new right neighbour. Thus e_4 makes sure to construct a new right neighbour cell (if needed), make this the new current cell, put a blank symbol into it, and update the state.

Note that given any state s, either it will contain a world satisfying $\mathsf{pre}(e_3)$ or a world satisfying $\mathsf{pre}(e_4)$, but not both. There will be a world satisfying $\mathsf{pre}(e_3)$ if the head stays within the previously used part of the tape when the transition represented by a is executed, otherwise there will be a world satisfying $\mathsf{pre}(e_4)$.

We can now finalise the proof. The set of local actions of the planning problem P_M is taken to be the set of local actions representing the transitions of M. Now suppose there is a move of the Turing machine from an instantaneous description ID_1 to an instantaneous description ID_2. We then need to prove that if s_1 is a local state representing ID_1, then for all actions a applicable in s_1, the local state $s_1 \otimes a$ represents ID_2. We need to split the proof into cases, distinguishing the cases where the tape head moves into the previously unused part of the tape, and those where it doesn't. We will only cover one of the cases here, as they are largely similar. Let us consider the most tricky case, where the tape gets extended. So assume ID_1 is an instantaneous description of the following form:

$$\mathrm{ID}_1 = x_1 \cdots x_{n-2} x_{n-1} q_s x_n,$$

and the move performed is a result of the following transition:

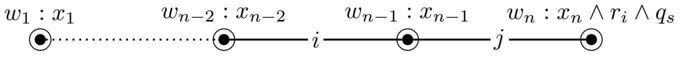
$$\delta(q_s, x_n) = (q_t, y, R), \text{where } x_n \neq y. \tag{5}$$

The move will then result in the following instantaneous description:

$$\mathrm{ID}_2 = x_1 \cdots x_{n-2} x_{n-1} y q_t b.$$

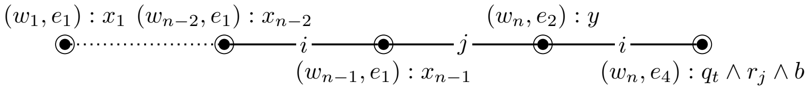
There are two local states that can represent ID_1, but they are symmetric, so we can without loss of generality assume that the local state s_1 representing ID_1 is the following:

$s_1 =$



Now, the only actions applicable in this state are actions containing at least one event with its preconditions being satisfied by $x_n \wedge r_i \wedge q_s$. This implies that the only action applicable is the one representing the transition (5) (recall that the Turing machine is deterministic). This is the action shown in (4). Taking the product update of s_1 with the action (4), we get the following local state s_2:

$s_2 =$


$(w_1, e_1) : x_1$
$(w_{n-2}, e_1) : x_{n-2}$
i
$(w_{n-1}, e_1) : x_{n-1}$
j
$(w_n, e_2) : y$
i
$(w_n, e_4) : q_t \wedge r_j \wedge b$

The reader is encouraged to check that this is indeed the correct product update of s_1 with the action (4). Action s_2 is immediately seen to be a representation of the instantaneous description ID_2, as required.

It now follows that if $a_1, a_2, \ldots$ is any sequence of actions in P_M where each a_i is applicable in $s_0 \otimes a_1 \otimes \cdots \otimes a_{i-1}$, then the sequence $s_0, s_0 \otimes a_1, s_0 \otimes a_1 \otimes a_2, \ldots$ will be a representation of the sequence of the moves of the Turing machine M. Now choose the goal formula of P_M to be $\neg C \neg q_f$. This formula expresses that there is world accessible by some path at which q_f holds. It holds exactly in those epistemic states representing halting states of the Turing machine. Thus we now have that the planning problem has a solution if and only if M halts. This is the required result.

The proof just given makes use of common knowledge. We can do away with common knowledge by introducing a third agent, k, instead. The idea is quite simple: whenever there is an i- or j-edge in any of the states or actions introduced above, we also add a k-edge. Since all accessibility relations are equivalence relations, this means that if a world of a state is accessible by *any* path, then it is accessible by a single k-edge. Thus we can replace the goal formula $\neg C \neg q_f$ by the formula $\neg K_k \neg q_f$. The rest of the proof remains unchanged. Whether epistemic planning with only two agents and no common knowledge is decidable or not is an open problem. ■

There seem to be no direct equivalents of this result in the existing literature, although there are obviously some connections to the non-stabilisation results of iterated updates over various types of purely epistemic actions in (Sadzik, 2006), and to the undecidability of the logic of iterated public announcements in (Miller *et al.*, 2005). The result above is of course not encouraging for epistemic planning in the general case, however, semi-decidability can sometimes be sufficient for a planner, as a planner embedded in an agent architecture (e.g. a BDI agent) would usually in any case rely on being timed out if a plan is not found within a reasonable time. From a positive perspective, this result shows that we have been introducing a very expressive planning framework, more expressive than previous frameworks suggested for planning based on epistemic logic (van der Hoek *et al.*, 2002; Petrick *et al.*, 2002) (since these other frameworks are known to be decidable, and ours is only semi-decidable in the most general case). In any case, an interesting problem of course becomes to find fragments of epistemic planning that *are* decidable. We already saw one such fragment, the single-agent case. Another decidable fragment is the one that only allows globally deterministic actions (cf. Section 5). This covers e.g. public announcements, atomic ontic actions, and sensing actions. That this fragment is decidable follows trivially from the fact that updates with actions having mutually inconsistent preconditions can never increase the model size. In (Löwe *et al.*, 2010), it is shown that interesting planning problems can be expressed even within such restrictive fragments. A more thorough investigation of which fragments of multi-agent epistemic planning are decidable is left for future work.

8. Related and future work

Work on using Dynamic Epistemic Logic in planning was recently independently initiated by Löwe, Pacuit and Witzel (Löwe *et al.*, 2010). Their work however differs from ours in a number of ways. They only consider purely epistemic actions (no postconditions), but on the other hand they allow arbitrary accessibility relations in models. Both accounts can surely be extended to cover both ontic actions and arbitrary accessibility relations with a bit of extra work. In (Löwe *et al.*, 2010), a restricted planning fragment is shown to be decidable by giving an upper bound on model size, but general algorithms and decidability issues are not covered. Our work also differs in allowing the *internal perspective* on planning, where the epistemic models represent the planning agent's internal view of the world. We showed that this gives a very nice and natural way to deal with partial observability in planning (even relevant in the single-agent case). The closest relative to our idea of epistemic models from an internal perspective appears to be the recent work by Aucher (Aucher, 2010), however, his approach is technically slightly different.

Another line of research considers *planning as model checking*. The idea is here to represent the state space of the planning problem as a model in a suitable temporal logic, and then recast the planning problem as a model checking problem (model checking of a formula expressing reachability of the goal state). The article (van der Hoek *et al.*, 2002) considers epistemic planning from this perspective. It however assumes that the state space is already given, and that it is finite. Thus, it doesn't consider the problem of how to express actions in a convenient formalism, and it doesn't allow the expressiveness we have in our formalism. In this approach, the treatment of epistemic and ontic change is similar—either way it is just a next step in a run, and how the valuation between different point changes is not essential to define or describe the transition (van Ditmarsch *et al.*, 2008). Note also that the word "tractable" in the article (van der Hoek *et al.*, 2002) refers to the fact that the model checking algorithm is polynomial in the *size of the model*, that is, in the size of the entire state space. Usually the complexity of planning problems is rather stated as functions of the size of the descriptions of the available actions, and from this perspective even classical propositional planning is PSPACE-complete.

Future work includes generalising the framework to arbitrary accessibility relations, in particular for representing belief rather than knowledge. However, for this to be realistic in planning scenarios, one should also take belief revision into account, e.g. along the lines of (Baltag *et al.*, 2007). We would also like to carry through a more thorough investigation of the decidable fragments of epistemic planning. Finally, and probably most importantly, we wish to develop languages suitable for describing local actions in a form which is more manageable and closer to the traditional planning languages like STRIPS. Such languages should not necessarily possess the full expressivity of local actions, but should be tailored for expressing a non-trivial subset relevant for actual planning problems in multi-agent domains (e.g. with different predefined action types for dealing with sensing, announcements, and ontic change, and parameters for specifying the observers of the action).

Acknowledgements

The authors would like to thank the following persons for valuable comments and ideas for improvement of the original draft: Torben Braüner, Valentin Goranko, Jens Ulrik Hansen, Martin Holm Jensen, and two anonymous reviewers. The work is partially funded by the Danish Natural Science Research Council under project HYLOCORE.

9. References

Aucher G., "An Internal Version of Epistemic Logic", *Studia Logica*, vol. 94, num. 1, pp. 1-22, 2010.

Bacchus F., Petrick R. P. A., "Modeling an Agent's Incomplete Knowledge During Planning and During Execution", *KR*, pp. 432-443, 1998.

Baltag A., Smets S., "A qualitative theory of dynamic interactive belief revision", *in* G. Bonanno, W. van der Hoek, M. Wooldridge (eds), *Texts in Logic and Games*, vol. 3, Amsterdam University Press, pp. 11–58, 2007.

Blackburn P., de Rijke M., Venema Y., *Modal Logic*, vol. 53 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, Cambridge, UK, 2001.

Dovier A., Piazza C., Policriti A., "A Fast Bisimulation Algorithm", *in* G. Berry, H. Comon, A. Finkel (eds), *CAV*, vol. 2102 of *Lecture Notes in Computer Science*, Springer, pp. 79-90, 2001.

Fagin R., Halpern J. Y., Moses Y., Vardi M. Y., *Reasoning About Knowledge*, MIT Press, 1995.

Gerbrandy J., "Communication Strategies in Games", *Journal of Applied Non-Classical Logics*, 2007.

Ghallab M., Nau D. S., Traverso P., *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004.

Hopcroft J. E., Motwani R., Ullman J. D., *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

Löwe B., Pacuit E., Witzel A., Planning based on dynamic epistemic logic, technical report, ILLC, University of Amsterdam, May, 2010.

Miller J. S., Moss L. S., "The Undecidability of Iterated Modal Relativization", *Studia Logica*, vol. 79, num. 3, pp. 373-407, 2005.

Petrick R. P. A., Bacchus F., "A Knowledge-Based Approach to Planning with Incomplete Information and Sensing", *in* M. Ghallab, J. Hertzberg, P. Traverso (eds), *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, AAAI Press, Menlo Park, CA, pp. 212–221, April, 2002.

Sadzik T., Exploring the iterated update universe, technical report, ILLC, University of Amsterdam, 2006.

van der Hoek W., Wooldridge M., "Tractable multiagent planning for epistemic goals", *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, ACM, New York, NY, USA, pp. 1167–1174, 2002.

van Ditmarsch H., Kooi B., "Semantic Results for Ontic and Epistemic Change", *in* G. Bonanno, W. van der Hoek, M. Wooldridge (eds), *Logic and the Foundation of Game and Decision Theory (LOFT 7)*, Texts in Logic and Games 3, Amsterdam University Press, pp. 87–117, 2008.

Wimmer H., Perner J., "Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception", *Cognition*, vol. 13, pp. 103–128, 1983.