Dynamic term-modal logics for first-order epistemic planning

Andrés Occhipinti Liberman [a], Andreas Achen [b], Rasmus Kræmmer Rendsvig [c],*

[a] *DTU Compute, Technical University of Denmark, Denmark*
[b] *London School of Economics, United Kingdom of Great Britain and Northern Ireland*
[c] *Center for Information and Bubble Studies, University of Copenhagen, Karen Blixens Plads 8, 2300 Copenhagen S, Denmark*

ARTICLE INFO

ABSTRACT

Many classical planning frameworks are built on first-order languages. The first-order expressive power is desirable for compactly representing actions via schemas, and for specifying quantified conditions such as $\neg \exists x \mathsf{blocks\_door}(x)$. In contrast, several recent epistemic planning frameworks are built on propositional epistemic logic. The epistemic language is useful to describe planning problems involving higher-order reasoning or epistemic goals such as $K_a \neg \mathsf{problem}$.

This paper develops a first-order version of Dynamic Epistemic Logic (DEL). In this framework, for example, $\exists x K_x \exists y \mathsf{blocks\_door}(y)$ is a formula. The formalism combines the strengths of DEL (higher-order reasoning) with those of first-order logic (lifted representation) to model multi-agent epistemic planning. The paper introduces an epistemic language with a possible-worlds semantics, followed by novel dynamics given by first-order action models and their execution via product updates. Taking advantage of the first-order machinery, epistemic action schemas are defined to provide compact, problem-independent domain descriptions, in the spirit of PDDL.

Concerning metatheory, the paper defines axiomatic normal term-modal logics, shows a Canonical Model Theorem-like result which allows establishing completeness through frame characterization formulas, shows decidability for the finite agent case, and shows a general completeness result for the dynamic extension by reduction axioms.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Most classical planning languages are first-order. Standard formalisms like PDDL [58] and ADL [69], for example, are first-order. One major reason for using a first-order language over a propositional one is that variables can be used to describe actions compactly. For instance, in the PDDL description of BlocksWorld, the *action schema* stack(X, Y) uses variables X and Y to represent generic blocks and state the preconditions and effects of all actions of the form: "put block X on top of block Y". This is possible because the action of stacking block A on block B has the same type of effects as the action of stacking block C on D; only the names of the blocks are different. Action schemas use variables to exploit this repeated structure in actions, resulting in action representations whose size is independent of the number of objects in a domain. While stack(X, Y) describes the preconditions and effects of performing a stack action on any two blocks, regardless of total number of blocks, with a propositional language each stack action has to be represented by a separate model, yielding $n^2 - n$ propositional models for a domain with n blocks. Generally, given an action schema with k variables and n constant

symbols standing for domain objects, the schema has up to n^k different instantiations, each requiring a separate model in a propositional representation.

Dynamic Epistemic Logic (DEL) has proved to be a very expressive framework for *epistemic planning*, i.e., planning explicitly involving e.g. knowledge or belief. DEL uses the language of propositional epistemic logic to describe the knowledge or belief held by a community of agents. This language is built from a set of propositional atoms, standard logical connectives, and modal operators K_i for each agent i in a fixed set of agent indices $I = \{1, \ldots, n\}$. An example of a formula is $K_1 p \wedge K_2 K_1 p$, which expresses that agent 1 knows the propositional atom p and that agent 2 knows that agent 1 knows p. Actions in DEL are described by so-called *action models* [6,7] or variants thereof. Action models describe preconditions and effects of events, and provide a rich framework for representing the agents' uncertainty about such events. However, as action models are based on the propositional epistemic language, propositional DEL cannot achieve the generality of action schemas. Variabilized, general descriptions are not possible, so one action model is required for each action. Hence, while propositional DEL adds a great deal of expressivity to planning, this comes at a cost in terms of representational succinctness.

This paper presents a DEL-based epistemic planning framework built on *epistemic term-modal logic*. The underlying language is first-order and includes modalities indexed by first-order terms. Examples of formulas include $K_c\, \mathrm{on}(A, B)$ (agent c knows that block A is on block B), $K_c \exists x\, \mathrm{on}(x, B)$ (c knows that there is a block on top of B) and $\forall y K_y \exists x\, \mathrm{on}(x, B)$ (all agents know that there is a block on top of B). Term-modal languages thus extend the expressive power of first-order modal languages by treating modal operators *both* as operators *and* as predicates.

In addition to higher-order knowledge expressions, the first-order apparatus of epistemic term-modal logic allows for domain descriptions in terms of objects and relations, as well as abstract reasoning via variables and quantification. The term-modal aspect ensures that these first-order aspects also extend to agents and their knowledge. Importantly, the presence of variables enables the definition of *epistemic action schemas*. Epistemic action schemas can be exponentially more succinct than standard DEL event models (see Section 5.3). Moreover, epistemic action schemas provide an action representation whose size is independent of the number of agents and objects in the domain. We consider the development of this epistemic planning framework our first main contribution.

Our second main contribution is the development of term-modal logic, its dynamic extension and the metatheory for both. Many papers have been dedicated to term-modal logic and its metatheory (see Section 9.3 for a detailed review), but due to the many complications that may arise in such generalized first-order modal systems, no general completeness results have been shown. In this paper, we define a rich but well-behaved semantics that allow us to define axiomatic normal term-modal logics and show a Canonical Model Theorem-like result that allow completeness results through frame characterization formulas. Adding reduction axioms to the term-modal logics then allow us to show completeness for the dynamic extension.

The paper progresses as follows. Section 2 presents *SelectiveCommunication* used as running example of epistemic planning. Section 3 presents term-modal logical languages and Section 4 defines state representations: first-order Kripke models where the agent set is a part of the domain of quantification. Section 5 introduces action representations (action models) and how these may be succinctly represented as action schemas. The action representations are used in Section 6 to define epistemic planning problems and related notions, and an example describing a term-modal planning domain and problem using a 'PDDL-like syntax' is there given. Section 7 details how to extend the term-modal language to allows reasoning about actions, Section 8 turns to axiomatic systems and metatheory, while Section 9 turns to related work on epistemic planning, dynamic epistemic logic and term-modal logic, respectively. Section 10 contains final remarks and open questions. All proofs may be found in Appendix A.

2. A running example

Throughout the paper, we illustrate the planning formalism with a simple running example in a variant of the *SelectiveCommunication* (SC) domain, adapted from [45]. Here we describe it informally, but it will serve as an example for the various formal notions throughout the paper. In the $SC(n, m, k, \ell)$ domain, there are n agents. Each agent is initially in one of m rooms arranged in a corridor. There are k boxes distributed in the rooms, each having one of ℓ available colors. See Fig. 1 for an example. Agents can perform four types of actions:

- Move(*agent*, *room*1, *room*2): agents can move from a room to a contiguous room, by going right or left. In this adaptation of the domain, we model the move actions as partially observable: if agent α is in room ρ_i and moves to room ρ_j, only the agents in either of the rooms can see that α's location has changed.
- SenseLoc(*agent*, *agent_or_box*, *room*): while in a room, agents can sense the location of other agents or boxes in that room. Other agents in the room notice the sensing action.
- SenseCol(*agent*, *color*, *box*, *room*): agents can sense the color of a box when they are in the same room as the box. Other agents in the room notice the sensing action.
- Announce(*agent*, *color*, *box*, *room*): agents can make announcements concerning the colors of boxes. If α makes an announcement in a room, all agents in the same room or in a contiguous room will hear what was announced. α can use announcements to ensure that some agents get to know the truth value of some φ while the remaining agents do not.
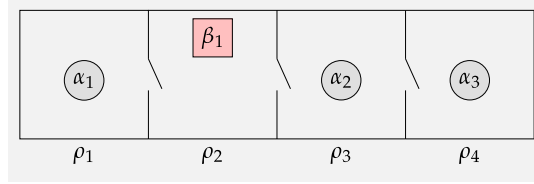
Fig. 1. A depiction of a possible state in $SC(3, 4, 1, 2)$, where a red box β_1 is in room ρ_2. (For colored figures, the reader is referred to the web version of this article.)

A specific choice for the parameters n, m, k, ℓ yields an *instance* of the SC domain. For example, $SC(3, 4, 1, 2)$ is the instance of *SelectiveCommunication* involving three agents (α_1, α_2 and α_3), four rooms (ρ_1, ρ_2, ρ_3 and ρ_4), one box (β_1) and two possible colors for the box (e.g., red and green). Fig. 1 depicts a *possible state* of the environment in this domain.

A possible *goal* $g = g_1 \wedge g_2 \wedge g_3$ in this domain is given by the conjunction of the following subgoals:

- g_1: α_1 and α_2 know the color of β_1
- g_2: α_1 knows that α_2 knows the color of β_1
- g_3: α_1 knows that α_3 does not know the color of β_1

That is, g requires α_1 to learn the color of β_1 and privately communicate this information to α_2. I.e., the goal is epistemic; it requires α_1 to achieve *first-order knowledge* about the environment (g_1) as well as *higher-order knowledge* about what others know (g_2 and g_3).

The nature of a *plan* for achieving g depends on the *initial state* as well as the assumptions made about the planning problem. For illustrative purposes, we consider a simple problem. Suppose that only α_1 can act and that the initial state s_0 satisfies the following conditions:

- c_1: each agent knows the location of all agents and the box β_1.
- c_2: no agent knows the color of β_1 (which is in fact red).
- c_3: conditions c_1 and c_2 are common knowledge among the agents.

In this case, α_1 can easily reach a state satisfying g from s_0. The following plan achieves g: α_1 *moves to* ρ_2, α_1 *senses the color of* β_1, α_1 *announces the color of* β_1. Of course, more initial uncertainty, or allowing other agents to act (sequentially or in parallel), results in more complex tasks. Such tasks can be defined with the formalism presented in this paper; however, for a first take on the formalism, this toy problem will be considered.

For additional examples, in [53] we use the framework to model social networks with epistemic dynamics.

3. Language

As term-modal logical languages include first-order aspects, they are parameterized by a *signature* specifying the *non-logical symbols* and their *type*—i.e., the constants and relation- and function symbols, and the sort and order of arguments (agent or object) they apply to. Also variables are here assigned a type.

Notation 1. For a vector $v = (x_1, ..., x_n)$, let *len*(v) denote its length, let v_i denote its i'th element, i.e., $v_i := x_i$, and let $v_{|k}$ denote its restriction to its prefix of length k, i.e., $v_{|k} := (x_1, ..., x_k)$.

Definition 1. A *signature* is a tuple $\Sigma = (V, C, R, F, t)$ with V a countably infinite set of variables, and C, R and F countable sets of respectively constants, relation symbols and function symbols with the one requirement that $\{=\} \subseteq R$. Finally, t is a type assignment map that satisfies

1. For $x \in V$, $t(x) \in \{\text{agt}, \text{obj}\}$ such that both $V \cap t^{-1}(\text{agt})$ and $V \cap t^{-1}(\text{obj})$ are countably infinite.
2. For $c \in C$, $t(c) \in \{\text{agt}, \text{obj}\}$.
3. For $r \in R$,
   (a) for some $n \in \mathbb{N}$, $t(r) \in \{\text{agt}, \text{obj}, \text{agt\_or\_obj}\}^n$, and
   (b) for $= \in R$, $t(=) = (\text{agt\_or\_obj}, \text{agt\_or\_obj})$.
4. For $f \in F$,
   (a) for some $n \in \mathbb{N}$, $t(f) \in \{\text{agt}, \text{obj}, \text{agt\_or\_obj}\}^n \times \{\text{agt}, \text{obj}\}$, and
   (b) if $t(f)_{|n} = (t(t_1), ..., t(t_n))$, then $t(f(t_1, ..., t_n)) = t(f)_{n+1}$.

Identity is treated as a relation symbol; for it, infix notation is used with $= (a, b)$ written $a = b$.

Example 1 *(Signature for SelectiveCommunication).* The following signature $\Sigma = (\text{V}, \text{C}, \text{R}, \text{F}, \text{t})$ can be used to specify the $SC(n, m, k, \ell)$ domain introduced in Section 2:

- Variables $\text{V} = \{x^\star, x, y, z, x_1, x_2, x_3, \dots\}$.
- Constants $\text{C} = Agents_{con} \cup Rooms_{con} \cup Boxes_{con} \cup Colors_{con}$, where $Agents_{con} = \{a_1, \dots, a_n\}$, $Rooms_{con} = \{r_1, \dots, r_m\}$ $Boxes_{con} = \{b_1, \dots, b_k\}$ and $Colors_{con} = \{c_1, \dots, c_\ell\}$.
- Relation symbols $\text{R} = \{\text{In}, \text{Color}, \text{Adj}, =\}$ where $\text{In}(x, y)$ states that agent or box x is in room y, $\text{Color}(x, y)$ states that box x has color y and $\text{Adj}(x, y)$ states that room x is adjacent to room y.
- Function symbols $\text{F} = \emptyset$.
- Type assignment t with constant types $\text{t}(x) = \text{agt}$ for $x \in Agents_{con}$, $\text{t}(x) = \text{obj}$ for $x \in Rooms_{con} \cup Boxes_{con} \cup Colors_{con}$, relation types $\text{t}(\text{In}) = (\text{agt\_or\_obj}, \text{obj})$, and $\text{t}(\text{Color}) = \text{t}(\text{Adj}) = (\text{obj}, \text{obj})$.

Definition 2. The set of *terms* T of a signature $\Sigma = (\text{V}, \text{C}, \text{R}, \text{F}, \text{t})$ is given by the grammar

$$t ::= x \mid c \mid f(t_1, \dots, t_n)$$

for $x \in \text{V}$, $c \in \text{C}$ and $f \in \text{F}$, provided that $t_1, \dots, t_n \in \text{T}$ and $\text{t}(f)_{|len(\text{t}(f))-1} = (\text{t}(t_1), \dots, \text{t}(t_n))$.

A term is *ground* if it does not contain any variables; it is *free* if all its terms are (i) variables or (ii) function symbols all whose arguments are free terms.

By the definitions of type assignments and terms, it is the case that for all $t \in \text{T}$, $\text{t}(t) \in \{\text{agt}, \text{obj}\}$. This allows for a uniform definition of formulas in term-modal languages:

Definition 3. Let $\Sigma = (\text{V}, \text{C}, \text{R}, \text{F}, \text{t})$ be a signature. Let $t_1, \dots, t_n \in \text{T}$ and $r \in \text{R}$ with $\text{t}(r) = (\text{t}(t_1), \dots, \text{t}(t_n))$, let $\dagger \in \text{T}$ with $\text{t}(\dagger) = \text{agt}$, and let $x \in \text{V}$. The *language* $\mathcal{L}$ is then given by the grammar

$$\varphi ::= r(t_1, \dots, t_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_\dagger\varphi \mid \forall x\varphi$$

An *atom* is a formula obtained by the first clause. An atom is *ground* if all its terms are ground; it is free if all its terms are *free*. Denote by $\texttt{GroundAtoms}(\mathcal{L})$ and $\texttt{FreeAtoms}(\mathcal{L})$ the set of all ground and free atoms in $\mathcal{L}$, respectively.

Throughout, the standard Boolean connectives as well as $\top$, $\bot$ and $\exists$ are used as meta-linguistic abbreviations as usual. We abbreviate inequality expressions of the form $\neg(t_1 = t_2)$ by $(t_1 \neq t_2)$. *Free* and *bound* variables may be defined recursively as usual with the free variables of $K_t\varphi$ the free variables of φ plus the variables in t. A formula is a *sentence* if it has no free variables. With $\varphi \in \mathcal{L}, t \in \text{T}, x \in \text{V}, \text{t}(x) = \text{t}(t)$ and no bound variables of φ occurring in t, the result of replacing all occurrences of x in φ with t is denoted $\varphi(x \mapsto t)$.

Remark 1. $K_t\varphi$ is read as "agent t knows that φ". Epistemic expressions are only well-defined when t is an agent term. The language $\mathcal{L}$ neither enforces nor requires a fixed-size agent set, in contrast with standard epistemic languages, where the set of operators is given by reference to some index set. Fixed-size agent sets are discussed throughout.

4. State representation

In planning frameworks based on epistemic logic, states are often represented using *possible-worlds models*, tracing back to the work of Hintikka [39] and Kripke [49]. The standard epistemic interpretation of such models—employed here in all examples—is one of *indistinguishability*, as follows. A model contains a set of worlds, each representing a physical state of affair. For each agent, a model contains a binary relation on the set of worlds. Under the indistinguishability interpretation, this relation is taken to be an equivalence relation. If two worlds are related for agent α, then α cannot distinguish them given her current information. I.e., they are *informationally indistinguishable* for α. Hence, when α in fact is in some world w, she cannot tell which of the worlds related to w she is in fact in. The set of worlds indistinguishable from w for agent α is therefore sometimes referred to as agent α's *range of uncertainty* (at w). The term *information cell* is used to cover the same, and a world in α's range of uncertainty is said to be considered *possible* by α (at w). An agent's range of uncertainty determines its knowledge: an agent knows φ in world w if φ is true in all the worlds in the agent's range of uncertainty at w. For instance, if the agent has no information about two blocks A and B, and therefore cannot tell whether one of them is stacked on the other or not, she will consider at least three worlds possible: one in which A is indeed stacked on B, one in which it is not, and one in which B is stacked on A. Possible-worlds models represent also all levels of *higher-order knowledge*. E.g. agent α knows that agent β knows φ if α does not consider it possible that β considers it possible that φ is false.

A possible-worlds model is formally defined as a structure in general called a *Kripke model*. Kripke models need not enforce any properties on the agents' relations. Our results hold for the general case, with equivalence relations a special case. Under the indistinguishability interpretation, Kripke models are often called *epistemic models* or *epistemic states*. For a

thorough explanation of the components of a Kripke model, we refer the reader to [8,27]. When the context makes it clear, such a structure may simply be called a *model*. A model consists of a *frame* and an *interpretation*. Two things differentiate the frame used here from the standard, propositional version. First, a frame here contains a constant domain of elements existing in each world. Working with distinct agents and objects, the domain is a disjoint union of two sets, the agent domain and the object domain. Second, the accessibility relations over worlds are directly associated with elements in the agent domain. The agent domain thereby makes reference to an index set—as used in non-term-modal logical frames—redundant. The definition of a frame is thereby self-contained.

Definition 4. A ***frame*** F is a triple $F = (D, W, R)$ where

1. $D := D_{\texttt{agt\_or\_obj}} := D_{\texttt{agt}} \dot{\cup} D_{\texttt{obj}}$, called the *domain*, is the disjoint union of the non-empty sets $D_{\texttt{agt}}$ and $D_{\texttt{obj}}$, called the *agent domain* and the *object domain*, respectively.
2. W is a non-empty set of *worlds*.
3. R is a map associating to each agent $i \in D_{\texttt{agt}}$ a binary *accessibility relation* on W. I.e., $R : D_{\texttt{agt}} \longrightarrow \mathcal{P}(W \times W)$.

Write R_i for $R(i)$, write $w R_i w'$ for $(w, w') \in R_i$ and write $R_i(w)$ for $\{w' \in W : w R_i w'\}$. If $|D_{\texttt{agt}}| = n$ and $|D_{\texttt{obj}}| = m$, $(n, m \in \mathbb{N})$, say F is of *size* (n, m). Denote by $\textbf{F}$ the class of all frames and by $\textbf{F}_{(n,m)}$ the class of all frames of size (n, m).

For propositional modal logic, a frame is augmented by a valuation assigning an extension of worlds to each propositional symbol. In the first-order and term-modal cases, each non-logical symbol is assigned an extension in the domain. Here, this extension is assigned *world-relatively* for both relation symbols, function symbols and constants. In particular the last is note-worthy: the constants are thereby *non-rigid*—they may refer to different objects (and agents!) in different worlds. The non-rigidity of constants allows for uncertainty about identity cf. the example of Section 6.2 and play an important role concerning the validity of frame-property characterizing axioms, cf. Section 8.1.4.

Constants that do not vary with worlds—so-called *rigid* constants—often come in handy when referring to agents. A rigid constant provides a syntactic name for a semantic agent. Rigid constants are a special case: a constant c may be forced rigid by assuming its interpretation $I(c, w)$ to be constant over all worlds, i.e., by $I(c, w) = I(c, w')$ for all $w, w' \in W$.

Definition 5. Let a signature $\Sigma = (\texttt{V}, \texttt{C}, \texttt{R}, \texttt{F}, \texttt{t})$ and a frame $F = (D, W, R)$ be given. An ***interpretation*** of Σ over F is a map I satisfying for each $w \in W$:

1. $I(=, w)$ is the set $\{(d, d) : d \in D\}$.
2. For $c \in \texttt{C}$, $I(c, w) \in D_{\texttt{t}(c)}$.
3. For $r \in \texttt{R}$, $I(r, w) \subseteq \prod_{i=1}^{len(\texttt{t}(r))} D_{\texttt{t}_i(r)}$.
4. For $f \in \texttt{F}$, $I(f, w) \subseteq \prod_{i=1}^{len(\texttt{t}(f))} D_{\texttt{t}_i(f)}$ such that $I(f, w)$ is a (possibly partial) map: i.e., if $(d_1, ..., d_k, d_{len(\texttt{t}(f))}), (d_1, ..., d_k, d'_{len(\texttt{t}(f))}) \in I(f, w)$, then $d_{len(\texttt{t}(f))} = d'_{len(\texttt{t}(f))}$.

With $F = (D, W, R)$ a frame and I an interpretation of Σ over F, the tuple $M = (D, W, R, I)$ is a ***model***. Both $w \in F$ and $w \in M$ states that $w \in W$. When $w \in M$, the pair (M, w) is a ***pointed model*** with w called the *designated world*.

Satisfaction for all formulas without variable occurrences may be defined over pointed models. To specify satisfaction for the full language, variables must also be assigned extension. Letting variable valuations be world independent—or *rigid*—trans-world identification of objects and agents may be made using suitable bound variables, cf. e.g. the *de re* knowledge in Example 2.

Definition 6. Let a signature $\Sigma = (\texttt{V}, \texttt{C}, \texttt{R}, \texttt{F}, \texttt{t})$ and a frame $F = (D, W, R)$ be given. A ***valuation*** of Σ over F is a map $v : \texttt{V} \longrightarrow D$ such that $v(x) \in D_{\texttt{t}(x)}$. An ***x-variant*** v' of v is a valuation such that $v'(y) = v(y)$ for all $y \in \texttt{V}, y \neq x$.

Jointly, an interpretation and a valuation assigns an extension to every term t of Σ relative to every world of a frame. The following involved, but uniform, notation will be used throughout to denote the extension of terms:

Definition 7. Let a signature $\Sigma = (\texttt{V}, \texttt{C}, \texttt{R}, \texttt{F}, \texttt{t})$, a model $M = (D, W, R, I)$ and a valuation v be given. The ***extension*** of the term $t \in \texttt{T}$ in M under v is

$$\llbracket t \rrbracket_w^{I, v} = \begin{cases} v(t) & \text{if } t \in \texttt{V} \\ I(t, w) & \text{if } t \in \texttt{C} \\ d \text{ with } (d_1, ..., d_n, d) \in I(f, w) & \text{if } t = f(t_1, ..., t_n) \end{cases}$$
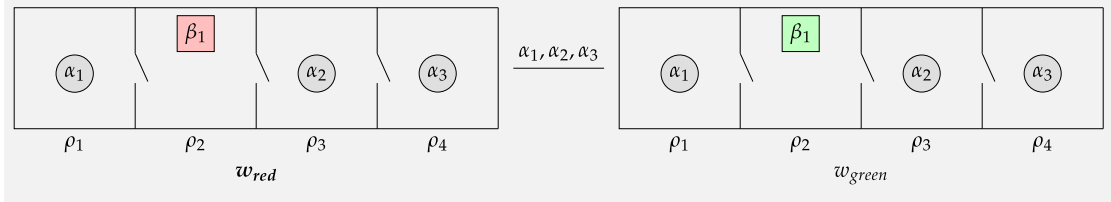
Fig. 2. (M_0, w_{red}), a pointed epistemic model for the initial state s_0 described in Section 2. The agents are uncertain about the color of β_1, which may be red (w_{red}) or green (w_{green}). This is captured by the edge linking w_{red} and w_{green}. Reflexive edges are not drawn. The name of the actual world, w_{red}, is marked with boldface letters.

For exactly the terms t with $\mathtt{t}(t) = \mathtt{agt}$, $R_{[\![t]\!]_w^{I,v}}$ is then an accessibility relation in M. The extension of terms thus play a key role in the satisfaction of modal formulas:

Definition 8. Let $\Sigma = (\mathtt{V}, \mathtt{C}, \mathtt{R}, \mathtt{F}, \mathtt{t})$, $M = (D, W, R, I)$ and v be given. The **_satisfaction_** of formulas of $\mathcal{L}$ is given by

$M, w \vDash_v r(t_1, ..., t_n)$ iff $([\![t_1]\!]_w^{I,v}, ..., [\![t_n]\!]_w^{I,v}) \in I(r, w)$ for all $r \in \mathtt{R}$, including $=$.
$M, w \vDash_v \neg\varphi$ iff not $M, w \vDash_v \varphi$.
$M, w \vDash_v \varphi \wedge \psi$ iff $M, w \vDash_v \varphi$ and $M, w \vDash_v \psi$.
$M, w \vDash_v \forall x\varphi$ iff $M, w \vDash_u \varphi$ for every x-variant u of v.
$M, w \vDash_v K_t\varphi$ iff $M, w' \vDash_v \varphi$ for all w' such that $(w, w') \in R_{[\![t]\!]_w^{I,v}}$.

Example 2 (_Epistemic model for SC(3, 4, 1, 2)_). Fig. 2 depicts an epistemic model $M_0 = (D, W, R, I)$ for the initial state s_0 described in Section 2. Formally, the model $M_0 = (D, W, R, I)$ has

- $D = D_{\mathtt{agt}} \dot\cup D_{\mathtt{obj}}$, with $D_{\mathtt{agt}} = \{\alpha_1, \alpha_2, \alpha_3\}$ and $D_{\mathtt{obj}} = Rooms_{obj} \cup Boxes_{obj} \cup Colors_{obj}$, where $Rooms_{obj} = \{\rho_1, \rho_2, \rho_3, \rho_4\}$, $Boxes_{obj} = \{\beta_1\}$ and $Colors_{obj} = \{Red, Green\}$.
- $W = \{w_{red}, w_{green}\}$.
- $R(\alpha_i) = W \times W$, for $i \in \{1, 2, 3\}$.
- The interpretation of all constants is the same in w_{red} and w_{green}, i.e., all constants are rigid: $I(a_i, u) = \alpha_i$, $I(r_i, u) = \rho_i$, $I(b_i, u) = \beta_i$, $I(green, u) = Green$ and $I(red, u) = Red$ for all $u \in W$.
- The interpretation of the predicates is as follows: $I(\mathsf{In}, u) = \{(\alpha_1, \rho_1), (\alpha_2, \rho_3), (\alpha_3, \rho_4), (\beta_1, \rho_2)\}$, for all $u \in W$, $I(\mathsf{Color}, w_{red}) = \{(\beta_1, red)\}$, $I(\mathsf{Color}, w_{green}) = \{(\beta_1, green)\}$. The interpretation of the Adj predicate is as expected.

Following the semantics from Definition 8, it can be seen that $M_0, w_{red} \vDash_v \forall x(K_x\mathsf{In}(b_1, r_2))$, i.e., every agent knows the location of box β_1. Similarly, every agent knows that all agents know this, since $M_0, w_{red} \vDash_v \forall y\forall x(K_y K_x\mathsf{In}(b_1, r_2))$. Moreover, the agents know that the box has a color, but not what color it is. They thus have what is called _de dicto_ knowledge of the coloring of the box, but not _de re_ knowledge. Agent α_3's de dicto knowledge is captured by $M_0, w_{red} \vDash_v K_{a_3}\exists x\mathsf{Color}(b_1, x)$, while its lack of de re knowledge is captured by $M_0, w_{red} \vDash_v \neg\exists x K_{a_3}\mathsf{Color}(b_1, x)$. Finally, agent α_3 has de re knowledge of the box, or as Hintikka [39] puts it, α_3 knows what the box is, captured by $M_0, w_{red} \vDash_v \exists x K_{a_3}(x = b_1)$.

5. Action representation

In automated planning, a distinction is often drawn between _action schemas_, which describe classes of actions in a general way, and _ground actions_, which represent a specific action with a fixed set of agents and objects [35,79]. Action schemas use so-called _action parameters_ or _variables_, which are instantiated into constants to define an action. For example, a schema may be used to represent all actions of the form 'agent x tells y that object z has color u', where x, y, z and u are variables standing for agents and objects. A corresponding ground action is obtained by replacing all free variables by names referring to specific agents and objects. For example, a schema instance could be '_ann_ tells _bob_ that _box_1 has color _red_'.

In DEL, the descriptions of concrete actions are called _action models_. That is, DEL action models correspond to a ground action in classical planning. Following the DEL naming conventions, models of concrete actions will be called action models, whereas variabilized models in the spirit of PDDL will be called action schemas. Action models and action schemas, as well as a suitable notion of schema instantiation relating the two, are introduced next.

5.1. Action models

Formally and intuitively, action models are closely related to Kripke models. Where Kripke models contains worlds and relations, action models instead contain _events_ and relations. Under the standard epistemic interpretation, the relations again represent indistinguishability and are again assumed to be equivalence relations. Again, this is a special case of the models introduced here.

The DEL-style action models we add to the term-modal logic setting include *preconditions* ([6,7]), *postconditions* ([11,15, 26]) as well as *edge-conditions* similarly to [13]. Preconditions specify when an event is executable (e.g., a precondition of opening the door is that it is closed.) Postconditions describe the physical effects of events (e.g., the door is open after the event). Edge-conditions are used to represent how an agent's observation of an action depends on the agent's circumstances. For example, the way in which an agent α_i observes an action performed by agent α_j may depend on α_i's proximity to α_j or to the objects affected by the action (e.g., to α_i, the events of opening and closing the door are distinguishable if α_i can see or hear the door, but else not). Edge-conditions provide a general way to describe actions whose observability is context-dependent. The epistemic effects of an action model is encoded by the *product update* operation by which action models are applied to Kripke models (defined below).

In more detail, the components of term-modal action models (Definition 9 below) play the following roles. E represents the set of events that might occur as the action is executed. Q is a map that assigns to each edge $(e, e') \in E \times E$ an edge-condition: a formula with a single free variable $x^\star$. Given a model M describing the situation in which the action is applied, an agent α cannot distinguish e from e' iff the edge-condition from e to e' is true in M *when the free variable $x^\star$ is mapped to* α. Intuitively, if the situation described by the edge-condition is true for α, the way α observes the action does not allow her to tell whether e or e' is taking place. The precondition restricts the applicability of an event e to those states satisfying the precondition formula $\mathsf{pre}(e)$. Precondition formulas contain no free variables to ensure that their effects are conditional only on the model, but not the variable valuation. The postcondition $\mathsf{post}(e)$ describes the physical changes induced by the event. If both $\mathsf{pre}(e)$ and $\mathsf{post}(e)(r(t_1, \ldots, t_n))$ are true in a state s of a model M, then the event e occurs, and after its occurrence, $r(t_1, \ldots, t_n)$ is true in the updated version of s. That is, $r(t_1, \ldots, t_n)$ is a *conditional effect* of event e with condition $\mathsf{post}(e)(r(t_1, \ldots, t_n))$.

The language used to state pre- and postconditions in action models is an extension of $\mathcal{L}$, denoted $\mathcal{L}_{AM}$, to be introduced in Section 7. This extended language has formulas of the form $[A, e]\varphi$, which are interpreted as: 'after event e of action A, φ holds'. This type of formula makes it possible to mention other actions in the pre- and postconditions of actions, i.e., to express syntactically some dependencies or interactions between actions. However, the action model construction does not require or depend on the use of $\mathcal{L}_{AM}$ rather $\mathcal{L}$, so the reader can safely ignore the details of $\mathcal{L}_{AM}$ for now.

Definition 9. An **action model** A is a tuple $A = (E, Q, \mathsf{pre}, \mathsf{post})$ where

1. E is a non-empty, finite set of possible *events*.
2. $Q : (E \times E) \to \mathcal{L}_{AM}$, where for each pair (e, e') the formula $Q(e, e')$ has exactly one free variable $x^\star$.
3. $\mathsf{pre} : E \to \mathcal{L}_{AM}$ is a map that assigns to each event $e \in E$ a precondition formula with no free variables.
4. $\mathsf{post} : E \to (\texttt{GroundAtoms}(\mathcal{L}) \to \mathcal{L}_{AM})$ is a map that assigns to each event $e \in E$ a postcondition for each ground atom.

It is required that $\mathsf{post}(e)(= (t, t)) = \top$ for each event e, to preserve the meaning of equality. A pair (A, e) consisting of the action and an event from E is called a **pointed action**.

Notation 2. Let $A = (E, Q, \mathsf{pre}, \mathsf{post})$ be an action model. We denote by $\mathsf{dom}(\mathsf{post}(e))$ the set of atoms for which $\mathsf{post}(e)(r(t_1, \ldots, t_k)) \neq r(t_1, \ldots, t_k)$. We denote any $\mathsf{post}(e)$ that maps every atom to itself by *id* (the identity function). When convenient, we add the superscript "A" to the components of A, so that $A = (E^A, Q^A, \mathsf{pre}^A, \mathsf{post}^A)$.

To ensure that postcondition functions are finite objects, each $\mathsf{post}(e)$ is often required to be only finitely different from the identity function. That is, $\mathsf{dom}(\mathsf{post}(e))$ is required to be finite. This allows for a finite encoding of postconditions, as only pairs with $post(e)(\varphi) \neq \varphi$ need to be stored in memory. Especially in planning, it should be possible to write down a sequence of symbols that completely specifies any given action model in the language. For the sake of generality, we do not impose this restriction in the definition of an action model. But for all practical purposes, this standard restriction will be needed.

Notation 3. Let $A = (E, Q, \mathsf{pre}, \mathsf{post})$ be an action model. When A is illustrated as a labelled graph, for each node $e \in E$, we write the precondition and postconditions for e as a pair $\langle \mathsf{pre}(e); \mathsf{post}(e)(\psi_1) = \varphi_1 \land \cdots \land \mathsf{post}(e)(\psi_n) = \varphi_n \rangle$. We write postconditions such as $\mathsf{post}(e)(\varphi) = \top \land \mathsf{post}(e)(\psi) = \bot$ using the notation $\varphi \land \neg\psi$ (indicating that the action makes φ true and ψ false unconditionally). In graphs, we omit the postconditions for atoms φ with $post(e)(\varphi) = \varphi$.

We do not include the edge-conditions for reflexive loops in illustrations, but always assume that for all $e \in E$, $Q(e, e) = (x^\star = x^\star)$, to the effect that all agents retain reflexive relations following updates. When two events e, e' are connected by a line without arrowheads labeled by a single formula φ, this means that $Q(e, e') = Q(e', e) = \varphi$, retaining symmetry.

Example 3 (*Action models for SC(3, 4, 1, 2)*). Figs. 3, 4 and 5 depict graphically the action models for the three actions in the plan from in Section 2, i.e., the following movement, sensing and announcement actions: α_1 *moves to* ρ_2, α_1 *senses the color of* β_1, α_1 *announces the color of* β_1.
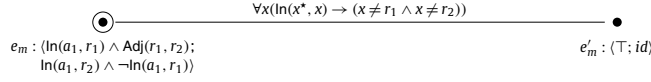
Fig. 3. Move(a_1, r_1, r_2), the action model for α_1 moving from ρ_1 to ρ_2. Event e_m describes what is actually taking place (in the drawing for the model, the actual event is marked with a double circle). The precondition formula says that α_1 is in ρ_1 and that ρ_2 is next to ρ_1. The action changes α_1's location to ρ_2, as captured by the postcondition. The event e'_m describes the situation in which nothing happens. This is how the action looks to any agent that is neither in the room α_1 is currently in, nor in the room the agent is moving to. The edge-condition linking the two events captures this observability constraint.
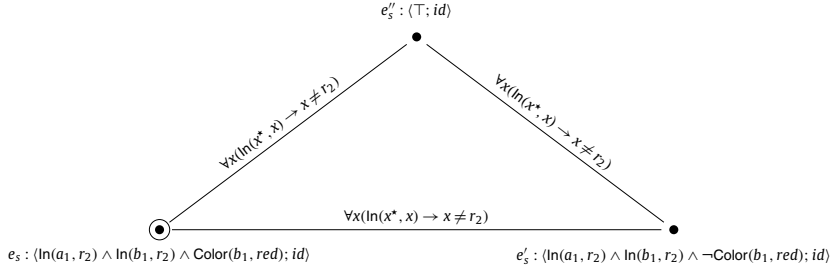

Fig. 4. SenseCol(a_1, red, b_1, r_2), the action model for α_1 sensing in room ρ_2 whether box β_1 is red or not. Event e_s describes what is actually taking place, i.e., α_1 seeing that the box is red. The action is a purely epistemic action, i.e., it does not change the physical state of the environment, and therefore the postcondition post(e_s) is id. e'_s represents the event in which α_1 sees that the box is not red, while e''_s represents the event in which nothing happens. The agents that are not in ρ_2 cannot observe what α_1 is doing. More precisely, they cannot distinguish between α_1 seeing that the box is red, α_1 seeing that it is not red, and α_1 doing nothing. This is captured by the edge-conditions.
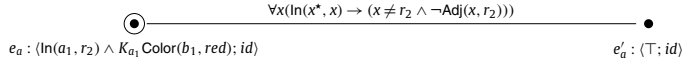
Fig. 5. Announce(r_1, red, b_1, r_2), the action model for α_1 announcing that β_1 is red while in room ρ_2. Event e_a describes what is actually taking place. The precondition formula pre(e_a) says that α_1 is in ρ_2 and that a_1 knows that the color of β_1 is red. Event e'_a describes the event in which nothing occurs. This is what the announcement looks like to any agent that cannot hear the announcement. An agent α_i cannot hear α_1's announcement if α_i is neither in α_1's room nor in a room that is adjacent to it. This is captured by the (identical) edge-conditions $Q(e_a, e'_a)$ and $Q(e'_a, e_a)$.

5.2. Product update

Having defined epistemic models and action models, we introduce an operation that computes the epistemic model M' reached by applying action A in model M. The operation is a first-order variant of *product update* [7]. Under the indistinguishability interpretation, the core epistemic intuition is that to tell two worlds apart after an update, either the agent could tell them apart beforehand, or it could tell them apart by something happening in one, but not the other. In slightly more detail: Assume that *after* an update, a model contains worlds (w, e) and (w', e'), representing that event e occurred in world w, and e' occurred in w'. Then (w, e) is indistinguishable from (w', e') for agent α iff α found both w and w' indistinguishable and events e and e' indistinguishable. Formally, (term-modal) product update is defined below. An explanatory remark follows the definition.

Definition 10. Let $M = (D, W, R, I)$ and $A = (E, Q, \text{pre}, \text{post})$ be given. The ***product update*** of M and A yields the epistemic model $M \otimes A = (D', W', R', I')$ where

1. $D' = D$,
2. $W' = \{(w, e) \in W \times E : (M, w) \vDash_v \text{pre}(e)\}$,
3. For each $i \in D_{\text{agt}}$, $(w, e)R'_i(w', e')$ iff wR_iw' and $M, w \vDash_{v[x^\star \mapsto i]} Q(e, e')$,
4. $I'(c, (w, e)) = I(c, w)$ for all $c \in \mathbb{C}$, $I'(f, (w, e)) = I(f, w)$ for all $f \in \mathbb{F}$, and $I'(r, (w, e)) = (I(r, w) \cup r^+(w)) \setminus r^-(w)$, where:

$$r^+(w) := \{(\llbracket t_1 \rrbracket^{I,v}_w, \ldots, \llbracket t_k \rrbracket^{I,v}_w) : (M, w) \vDash_v \text{post}(e)(r(t_1, \ldots, t_k))\}$$

$$r^-(w) := \{(\llbracket t_1 \rrbracket^{I,v}_w, \ldots, \llbracket t_k \rrbracket^{I,v}_w) : (M, w) \nvDash_v \text{post}(e)(r(t_1, \ldots, t_k))\}$$

If $(M, w) \vDash_v \text{pre}(e)$, (A, e) is ***applicable*** to (M, w). If (A, e) is applicable to (M, w), the product update of the two yields the pointed epistemic model $(M \otimes A, (w, e))$. Else it is undefined.

Remark 2. The components of the updated model are as follows. The domain of the updated model D' is unchanged, since action models change the state of agents and objects, but do not introduce or remove them. A state (w, e) is in the updated
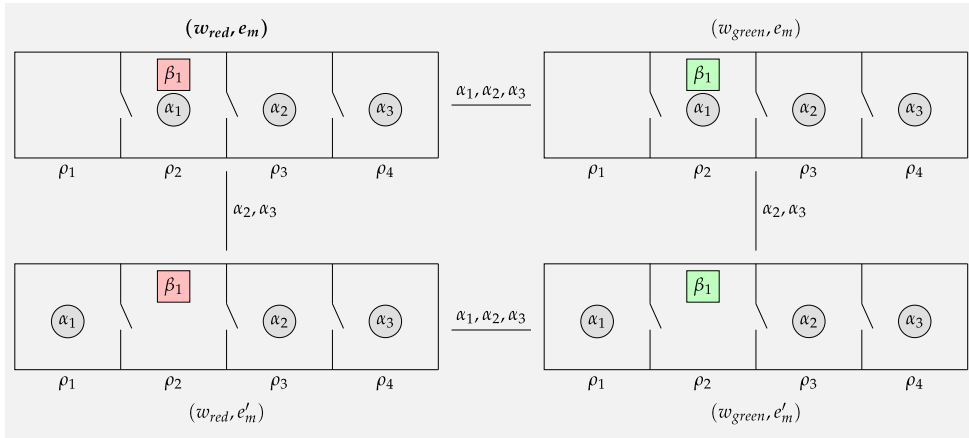
Fig. 6. The pointed model $(M_0 \otimes \text{Move}(a_1, r_1, r_2), (w_{red}, e_m))$, representing the state after α_1 moves into room ρ_2. Edges in the reflexive-transitive closure of the indistinguishability relations are omitted. At this point, α_2 and α_3 are uncertain about the location of α_1. More precisely, they cannot tell whether α_1 stayed in room ρ_1 or moved to ρ_2.

set of states W' if, and only if, e is applicable in w, i.e., if (M, w) satisfies the precondition $\text{pre}(e)$. As $\text{pre}(e)$ has no free variables by construction, the set of worlds W' is independent of the assignment v. The state (w, e) represents the state reached by taking event e in state w. Agent α cannot distinguish (w, e) from (w', e') if (1) α cannot distinguish w from w', which is the case if $wR_\alpha w'$; and (2) α cannot distinguish e from e' given its circumstances in w, which is the case if the edge-condition $Q(e, e')$ is true for agent α at (M, w) when $x^\star$ is mapped to α, i.e., when $M, w \vDash_{v[x^\star \mapsto i]} Q(e, e')$. Since actions do not change the denotation of ground terms, I' agrees with I in this respect. The extension of relations is changed according to event postconditions. If the condition $\text{post}(e)(r(t_1, \ldots, t_k))$ is true at (M, w), then the tuple $(\llbracket t_1 \rrbracket_w^{I, v}, \ldots, \llbracket t_k \rrbracket_w^{I, v})$ is added to the extension of r at (w, e), and it is removed otherwise.

Example 4 (*Product updates for SC(3, 4, 1, 2)*). Starting from the initial epistemic model (M_0, w_{red}) from Example 2, we model the effects of applying the actions in α_1's plan. First, α_1 moves right. This action yields the new pointed model $(M_0 \otimes \text{Move}(a_1, r_1, r_2), (w_{red}, e_m))$, depicted in Fig. 6. The second step is sensing the color of β_1. This action yields the model $(M_0 \otimes \text{Move}(a_1, r_1, r_2) \otimes \text{SenseCol}(a_1, red, b_1, r_2), (w_{red}, e_m, e_s))$, depicted in Fig. 7. Finally, α_1 announces in room ρ_2 that the color of β_1 is red. The result of this action is the model $(M_0 \otimes \text{Move}(a_1, r_1, r_2) \otimes \text{SenseCol}(a_1, red, b_1, r_2) \otimes \text{Announce}(a_1, red, b_1, r_2), (w_{red}, e_m, e_s, e_a))$, depicted in Fig. 8.

5.3. Succinct representation of actions via epistemic action schemas

We introduce epistemic action schemas, which represent sets of actions in a general way, as done in common planning formalisms such as PDDL. Schemas use *variables* to describe actions, rather than *constant symbols*. These variables denote arbitrary agents and objects and are used to describe their roles with respect to a type of action, such as the roles of *speaker* and *listener* in an action of type 'announcement'.

As anticipated in Section 3, a major reason for introducing schemas is that they result in action representations whose size is independent of the number of agents and objects in a domain. For the *SelectiveCommunication* domain $SC(n, m, k, \ell)$, there are $n \cdot m \cdot k \cdot \ell \cdot 2^{n-1}$ possible announcement actions, since each of the n agents could, in each of the m rooms, announce about each of the k boxes, that it is of one out of ℓ colors, with one out of the 2^{n-1} subsets of the other agents hearing the announcement. Representing all actions requires $n \cdot m \cdot k \cdot \ell \cdot 2^{n-1}$ standard DEL action models, i.e., one model per action. Variants of standard DEL models such as edge-conditioned models [13] fare substantially better, since the set of hearers is implicitly represented in such models, but $n \cdot m \cdot k \cdot \ell$ models are still required to represent the set of announcements. Other variants of DEL are also more succinct than standard DEL action models, e.g. the symbolic models of [21,86]. However, all these announcements can be compactly represented with a single *epistemic action schema*, as shown below in Example 5.

Moreover, epistemic schemas open up the possibility of applying well-known techniques such as *least commitment* or *partial order planning* [89] to epistemic problems. These approaches use the notion of a *partially instantiated action*, such as $Move(B, x, C)$, where x is a variable whose substitution has not yet been chosen. If specifying a binding constraint for x is unnecessary at the current point in the planning process, it is often advantageous to delay this commitment until later, i.e., until other necessary parts of the plan are discovered that further constrain what x should be. Other approaches to lifted planning, such as *hierarchical task networks* (HTNs) ([35], ch. 11) similarly exploit partial substitutions to optimize the search for solutions. For an epistemic version of lifted HTN planning, the epistemic schemas defined here could play the role of primitive tasks.

An epistemic action schema $a(x_1, \ldots, x_n)$ is defined using an *action name* a and a *parameter list* $(x_1, \ldots, x_n)$, as done e.g. in PDDL. The parameter list fixes a finite set of agents and objects involved in the execution of the action. Schemas
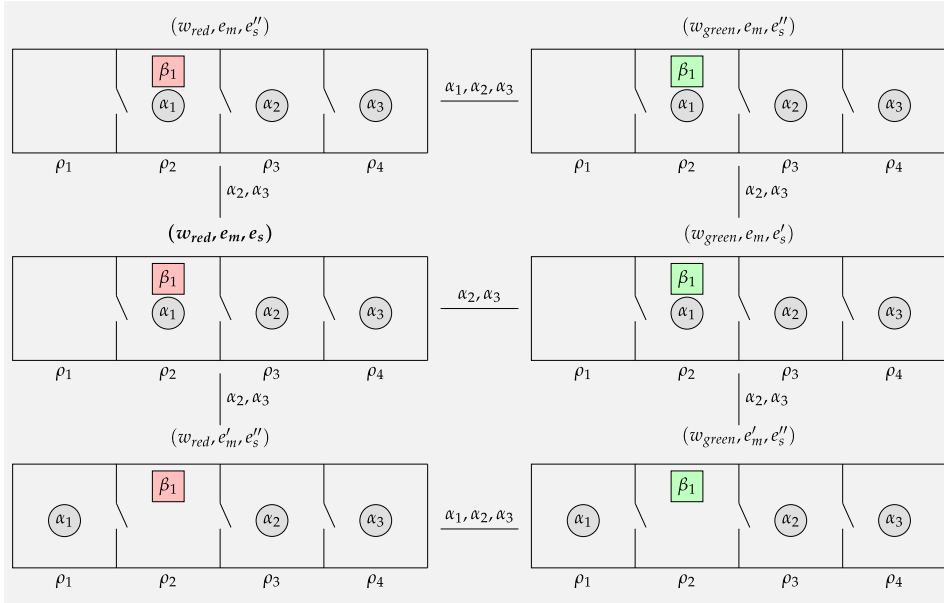
Fig. 7. The pointed model $(M_0 \otimes \text{Move}(a_1, r_1, r_2) \otimes \text{SenseCol}(a_1, red, b_1, r_2), (w_{red}, e_m, e_s''))$, representing the state after α_1 senses that β_1 is red while in room ρ_2. Edges in the reflexive-transitive closure of the indistinguishability relations are omitted. Note that, in the actual world, (w_{red}, e_m, e_s), α_1 does not face any uncertainty. In particular, $K_{a_1}\text{Color}(b_1, red)$ is true at (w_{red}, e_m, e_s). On the other hand, α_2 and α_3 have not observed any of α_1's actions. That is, α_1 may or may not have moved, and it may or may not have sensed whether β_1 is red. As a result, it holds at (w_{red}, e_m, e_s) that $\forall x(x \neq a_1 \rightarrow \neg K_x \text{In}(a_1, r_2) \land \neg K_x \text{Color}(b_1, red))$.
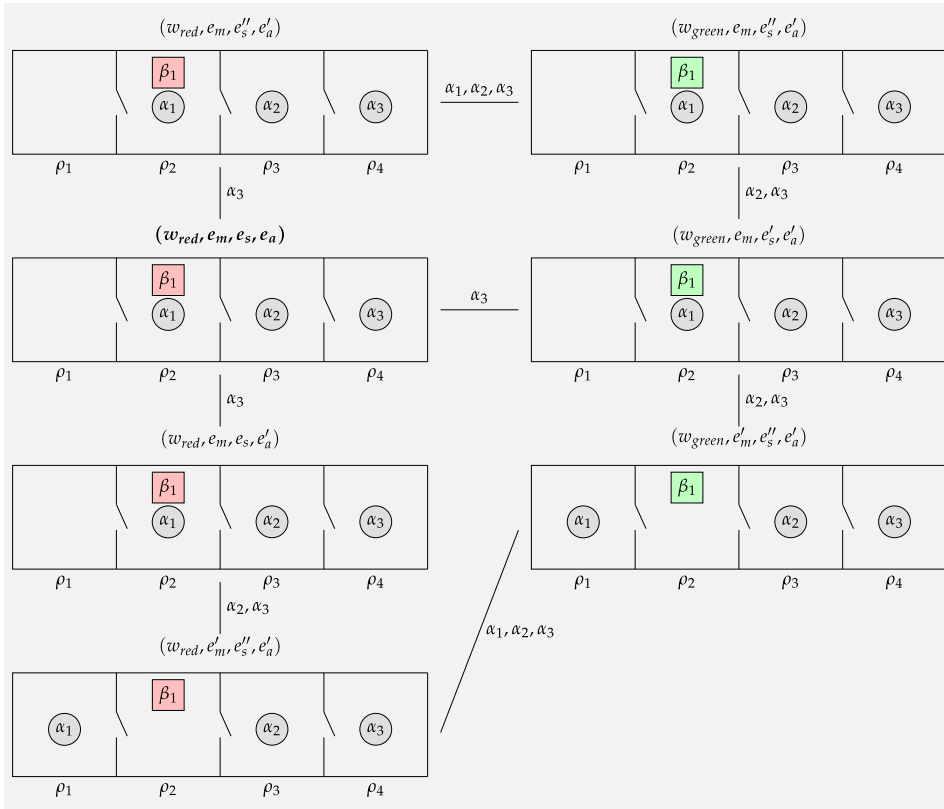

Fig. 8. The pointed model representing the state after α_1 announces that β_1 is red while at room ρ_2, $(M_0 \otimes \text{Move}(a_1, r_1, r_2) \otimes \text{SenseCol}(a_1, red, b_1, r_2) \otimes \text{Announce}(a_1, red, b_1, r_2), (w_{red}, e_m, e_s, e_a))$. Edges in the reflexive-transitive closure of the indistinguishability relations are omitted. Note that, in the actual world, (w_{red}, e_m, e_s, e_a), α_1 and α_2 do not face any uncertainty; there are no outgoing edges from the actual world for these agents. The goal g as stated in Section 2 holds in the actual world: both α_1 and α_2 know the color of β_1, α_1 knows that α_2 knows this, and α_1 knows that α_3 does not know this.
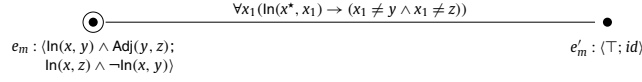
Fig. 9. Move(x, y, z), the action schema for agent x moving from room y to room z.
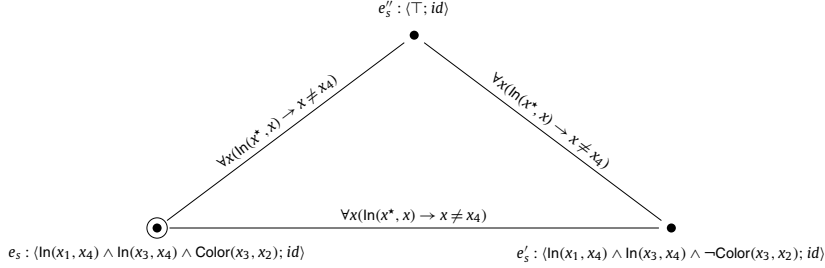

Fig. 10. SenseCol(x_1, x_2, x_3, x_4), the action schema for x_1 sensing in room x_4 whether box x_3 has color x_2.
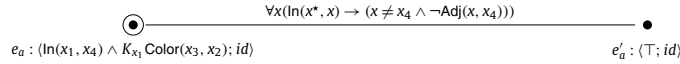

Fig. 11. Announce(x_1, x_2, x_3, x_4), the action schema for x_1 announcing that x_3 has color x_2 while in room x_4.

are required to follow a STRIPS-like scope assumption; all variables referenced in the preconditions or postconditions of an action schema must appear in the action's parameter list. Any agent or object unmentioned in the parameter list is assumed to be unrelated to the action's pre- and postconditions.

Definition 11. An ***epistemic action schema*** is of the form $a(\vec{x}) = (E, Q, \text{pre}, \text{post})$ where

1. a is the *action name* and $\vec{x} \in \mathbb{V}^n$ is a finite *parameter list*.
2. E is a non-empty, finite set of *events*.
3. $Q : (E \times E) \to \mathcal{L}_{AM}$ is an *edge-condition function*, where the formula $Q(e, e')$ has a free variable $x^\star$ of type `agt`, and possibly other free variables all in $\vec{x}$.
4. $\text{pre} : E \to \mathcal{L}_{AM}$ assigns to each event a *precondition formula* with all free variables in $\vec{x}$.
5. $\text{post} : E \to (\text{FreeAtoms}(\mathcal{L}) \to \mathcal{L}_{AM})$ assigns to each event a *partial postcondition function* such that if $y_1, \ldots, y_m$ all occur in $\vec{x}$, then $\text{post}(e)(r(y_1, \ldots, y_m))$ has all free variables from $\vec{x}$; else, $\text{post}(e)(r(y_1, \ldots, y_m))$ is undefined.

Let $\text{dom}(\text{post}(e))$ denote the set of atoms for which $\text{post}(e)(r(t_1, \ldots, t_k)) \neq r(t_1, \ldots, t_k)$.

The postcondition for each event e is defined as a partial function, with all atoms whose arguments are not a subset of those occurring in $\vec{x}$ left unaffected. Since the parameter list is required to be finite, this yields a finite encoding of postconditions.

Note that the parameter list of a schema may include *agent variables*. Just like any other action parameter, these variables can appear in the preconditions or effects of the schema. This is in line with what occurs in multi-agent extensions of PDDL such as MAPL or MA-PDDL [18,48]. Such variables are included to enable the schematization of actions also with respect to agents. In this paper, we adopt the epistemic operators from term-modal logic, *indexed by agent variables*, to achieve schematization with respect to agents when formalizing epistemic planning. In order to be able to express epistemic pre- or postconditions relative to an agent variable x in an action schema, it is necessary to use the variable in the scope of a modal operator. For example, if an action has e.g. a precondition that requires agent x to know $P(y, z)$, we need the term-modal formula $K_x P(y, z)$ to express this constraint in a schematic way.

Example 5 (*Action schemas for SC(n, m, k, ℓ)*). Figs. 9, 10 and 11 depict graphically the action schemas for the movement, sensing and announcement actions described in the example from Section 2. The schemas have the same structure as the action models from Example 3, but the conditions are now expressed in a general way, via free variables for agents, boxes and colors.

As usual in planning, schemas can be instantiated into concrete actions via grounding substitutions. Schema instantiation is defined as follows. Let $a(x_1, \ldots, x_n)$ be a schema and $\sigma : \{x_1, \ldots, x_n\} \to \mathbb{C}$ be a ***grounding substitution***, i.e., a mapping from variables into constants. For a formula φ, let $\varphi\sigma$ be the result of replacing each occurrence of a free variable y in φ by $\sigma(y)$.

Definition 12. Let $a(x_1, \ldots, x_n) = (E, Q, \mathsf{pre}, \mathsf{post})$ be an action schema and let $\sigma : \{x_1, \ldots, x_n\} \to \mathbb{C}$ be a grounding substitution. The **action model induced by** σ is $a(\sigma(x_1), \ldots, \sigma(x_n)) = (E', Q', \mathsf{pre}', \mathsf{post}')$ where

1. $E' = E$.
2. for each $e, e' \in E$, $Q'(e, e') = Q(e, e')\sigma$.
3. for each $e \in E$, $\mathsf{pre}'(e) = \mathsf{pre}(e)\sigma$.
4. for each $e \in E$,

$$\mathsf{post}'(e)(r(t_1, \ldots, t_n)\sigma) = \begin{cases} \mathsf{post}(e)(r(t_1, \ldots, t_n))\sigma & \text{if } \mathsf{post}(e)(r(t_1, \ldots, t_n)) \text{ is defined,} \\ r(t_1, \ldots, t_n)\sigma & \text{otherwise.} \end{cases}$$

Example 6. The announcement model A_1 in Fig. 5 is the ground action of schema S_1 from Fig. 11 induced by the substitution $\sigma = \{x_1 \mapsto a_2, x_2 \mapsto r_3, x_3 \mapsto b_1, x_4 \mapsto green\}$.

6. Problems, plans and solutions

This section defines *first-order epistemic planning tasks*. An epistemic planning task consists of an initial state, a set of actions, and a goal to be achieved. To solve an epistemic planning task, one may take either an *external* or and *internal* perspective [3]. The external perspective is the view of the system designer, who knows the precise initial state and actual effect of every action. The internal perspective is the view of an in-system agent, who has uncertainty about the state of the world and therefore uncertainty about the effects of executed actions. In the DEL planning framework, an external planning task is defined as a special case of a classical planning task (as in, e.g., [4,16]). Following [35], any *classical planning domain* can be described as a state-transition system $T = (\mathcal{S}, \mathcal{A}, \gamma)$ where $\mathcal{S}$ is a finite or recursively enumerable set of finite states, $\mathcal{A}$ is a finite set of actions and $\gamma : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a partial, computable state-transition function. A *classical planning task* is a triple (T, s_0, S_G), where T is a state-transition system, $s_0 \in \mathcal{S}$ is the initial state and $S_G \subseteq \mathcal{S}$ is the set of goal states. A *solution* to a classical planning task (T, s_0, S_G) is a plan consisting of a finite sequence of actions $a_1, a_2, \ldots, a_n$ such that (1) For all $i \leq n$, $\gamma(\gamma(\ldots \gamma(\gamma(s_0, a_1), a_2), \ldots, a_{i-1}), a_i)$ is defined, and (2) $\gamma(\gamma(\ldots \gamma(\gamma(s_0, a_1), a_2), \ldots, a_{n-1}), a_n) \in S_G$. Epistemic planning tasks can be defined as special cases of classical planning tasks, as follows.

Definition 13. Let A be a finite set of action schemas. A *(first-order) epistemic planning task* based on A is a triple $P = (s_0, \mathcal{A}, \varphi_G)$ where the initial state s_0 is a finite epistemic state with a finite domain, $\mathcal{A}$ is the set of all ground instances of the schemas in A, and the goal formula φ_G is a sentence of $\mathcal{L}$. Any epistemic planning task $(s_0, \mathcal{A}, \varphi_G)$ induces a classical planning task $((\mathcal{S}, \mathcal{A}, \gamma), s_0, S_G)$ given by:

- $\mathcal{S} := \{s_0 \otimes a_1 \otimes \cdots \otimes a_n \mid n \in \mathbb{N}, a_i \in \mathcal{A}\}$
- $S_G := \{s \in \mathcal{S} \mid s \vDash \varphi_G\}$
- $\gamma(s, a) := s \otimes a$ if a is applicable in s, else undefined.

A *solution* to an epistemic planning task is a solution to the induced classical planning task.

All the ingredients in this definition of an external planning task can come from the formalism presented here. Note that the planner-modeler in such a task is not one of the agents in the domain D_{agt}. The planner-modeler has access to the actual states s_i, i.e., to pointed models (M, w) where w is the actual world.

Formalisms of internal epistemic planning based on DEL are often defined from the external planning model, either by adding structure to the models or making small modifications. For example, [2] represents internal perspectives using *information cells*, which are defined from the accessibility relations of an epistemic model. An alternative involves using *multi-pointed models* or adding a set of so-called *designated points* to the epistemic model, with each point describing a world that the agent considers as possible from its internal perspective (see e.g. [15]). A third approach uses a belief state representation of the agent's internal view as primitive and then defines an epistemic model from it [45]. The approach in [3] offers two different flavors of internal view, both defined on the basis of a standard epistemic model. These various notions of internal perspective, as well as their associated planning tasks, may be upgraded to our framework without major modifications. We believe that the simplest way to add internal perspectives to the present would be to adopt the approach proposed by Bolander and Andersen [15]. Save for the fact that such models are multi-pointed, the core semantics remains the same. We therefore envision no major difficulty in bringing the internal perspective into our formalism; similarly defined multi-pointed structures should suffice. We do not develop the detail here.

6.1. Decidability of the plan existence problem

Having defined first-order epistemic planning tasks, a natural first question is whether the corresponding plan existence problem is decidable. We follow Aucher and Bolander [4] in defining the plan existence problem:

Definition 14. Let $n \in \mathbb{N}$. PlanEx(n) is the problem: "Given a (first-order) epistemic planning task $P = (s_0, \mathcal{A}, \varphi_G)$ where s_0 is an n-agent epistemic state, does P have a solution?".

For propositional DEL, the corresponding problem is undecidable in general [15]. This entails that the unrestricted first-order problem is undecidable as well, since first-order epistemic planning extends propositional DEL planning. However, decidable and reasonably expressive fragments of propositional DEL planning have been found, such as single-agent planning and multi-agent planning with non-modal preconditions. In [54], we show that *the corresponding first-order fragments are also decidable.* In that paper, bisimulations for the term-modal models presented here are introduced, and shown to have standard model-theoretic properties. Such bisimulations are key in proving the decidability results, as they allow us to show that the state spaces for certain planning fragments are finitely representable, up to bisimulation. We state the main results here and refer the reader to [54] for details.

Theorem 1 *([54]).* PlanEx(1) *(single-agent planning) is decidable.*

Theorem 2 *([54]).* *If all actions have non-modal preconditions, then* PlanEx(k) *is decidable, for $k \geq 1$.*

6.2. An example with a PDDL-like description

With the above, the dynamic term-modal planning framework of the paper has been introduced. This section contains a second example, using its different components in one place. The example also serves to illustrate how uncertainty about names may play a role in epistemic planning, and how a term-modal planning domain and an associated planning problem may be described using a 'PDDL-like syntax'. This description is meant as an indication of how such definitions could be standardized with a PDDL flavor, but no attempt is made at a precise syntax.

6.2.1. The MachineMalfunction (MM) domain

In the MM(n, m, k) domain, there are $n + m$ agents supervising k machines, the agents' tasked to ensure that machines function correctly at all times, where a choice of n, m and k fixes the universe of the domain. The agents have different roles: n of the agents are monitoring the machines for potential malfunctions, while the remaining m agents are system administrators, that from behind a terminal may solve any malfunction by issuing a *reboot* command to the affected machine. To reboot a machine, admins need to know its serial number, which a monitor may be uncertain about. Hence, the optimal sequence of actions to fix a malfunction problem will depend on how the knowledge of serial numbers is distributed amongst agents. Finally, company policy dictates epistemic preconditions for the actions: monitoring agents are only allowed to report machine as malfunctioning once they know that it is malfunctioning. To avoid deadlock, the requirement for admins is weaker: they may reboot any machine once they know some machine is malfunctioning.

The remainder of this section concerns an external epistemic planning task MM_task := $(s_0; \mathbf{A}; \varphi_g)$ in MM(1, 2, 2), i.e., with one monitor, two admins, and two machines.

6.2.2. Initial state and goal

Fig. 12 depicts the pointed epistemic model $s_0 := (M_0, w_0) := ((D, W, R, I), w_0)$, the initial state of MM_task. We aim for a simple example. Hence, the administrator/monitor roles are not formally specified, but could straightforwardly be assigned using predicates. Before stating the available actions, specify the goal of MM_task to be that some agent knows that all machines are not malfunctioning. I.e.,

$$\varphi_g := \exists x K_x \forall y \neg \text{Malfunction}(y)$$

With this goal achieved, the agent knowing that no malfunction occurs can announce this to the remaining agents to achieve that this becomes known to all, but for simplicity, we have omitted this aspect. The initial state s_0 and the goal φ_g may then be described in a 'PDDL-like syntax', as it would appear in a problem file, cf. Fig. 13.

6.2.3. Available actions as action schemas and domain definition

To finalize the specification of the external epistemic planning task MM_task, the available actions of reporting malfunctions rebooting machines must be defined. To this end, we use the action schemas depicted graphically in Fig. 14. Continuing with the 'PDDL-like syntax', Fig. 15 describes the *MachineMalfunction* domain. The figure is suggestive of a possible approach to standardizing domain definitions, but again, no formal specification of the syntax is attempted.

6.2.4. Plan and execution

Given the initial state and actions of the external epistemic planning task MM_task, the goal φ_g may be achieved by the monitor agent reporting that the machine colloquially called *box* is malfunctioning, after which the administrator α_1 may reboot that machine, by knowing its serial number $sn1$. I.e., the plan π below achieves φ_g, cf. Fig. 16:

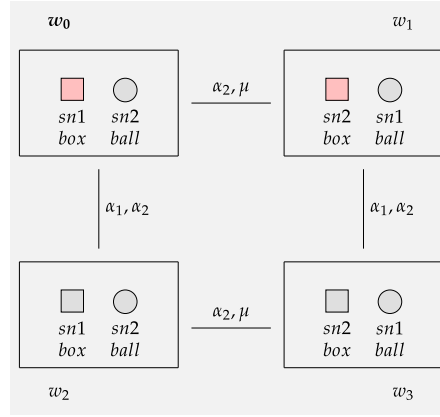$$\pi := \text{Malfunction}(m1, box), \text{Reboot}(a1, sn1)$$

Fig. 12. The pointed epistemic model $s_0 = (M_0, w_0)$, the initial state of MM_task. The agent domain D_{agt} comprises three agents, α_1, α_2 and μ, for simplicity only depicted as edge labels. Agents α_1 and α_2 are admins, while μ is a monitor. There are two machines, depicted as a box and a ball; beneath them are the constants that denote them in that world. In w_0, the box machine is denoted in-system by the serial number $sn1$, but colloquially as box, etc. In all worlds, the constants $a1, a2$ and $m1$ denote respectively α_1, α_2 and μ. The red coloring specifies malfunction. The monitor can observe malfunctions, while the admins cannot. Neither the monitor nor the newly employed admin α_2 know the serial numbers, but α_1 does. They all know the colloquial descriptions. The monitor knows that both admins face uncertainty about the malfunction, and that α_1 knows the serial numbers while α_2 does not.

```
(define (problem machine-malfunction-p1)
  (:domain machine-malfunction)
  (:universe
      Alpha1 Alpha2 - admin_agent
      Mu            - monitoring_agent
      Box Ball      - machine)
  (:constants
      sn1 sn2 box ball - machine_id
      m1               - monitoring_agent_id
      a1 a2            - admin_agent_id)
  (:init
    (:actual_world w0
      :constant_map ((sn1 Box) (sn2 Ball) (box Box)  (ball Ball) (m1 Mu) (a1 Alpha1) (a2 Alpha2))
      :atoms     ((malfunction sn1) (malfunction box)))
    (:world w1
      :constant_map ((sn2 Box) (sn1 Ball) (box Box)  (ball Ball) (m1 Mu) (a1 Alpha1) (a2 Alpha2))
      :atoms     ((malfunction sn2) (malfunction box)))
    (:world w2
      :constant_map ((sn1 Box) (sn2 Ball) (box Box) (ball Ball) (m1 Mu) (a1 Alpha1) (a2 Alpha2))
      :atoms     ())
    (:world w3
      :constant_map ((sn2 Box) (sn1 Ball) (box Box) (ball Ball) (m1 Mu) (a1 Alpha1) (a2 Alpha2))
      :atoms     ())
    (:edges
      :Alpha1 ((w0 -- w2) (w1 -- w3))
      :Alpha2 (all)
      :Mu     ((w0 -- w1) (w2 -- w3)))
  (:goal (exists (?a -agent_id) (knows (?a) (forall (?o - object_id) (not (malfunction ?o))))))))
```

Fig. 13. A 'PDDL-style syntax' description of the initial state s_0 and the goal φ_0. A universe of agents and objects is defined using the keyword `:universe`, with constants denoting this domain declared the keyword `:constants`. The `:init` keyword precedes the description of the initial state, which comprises worlds and edges between worlds. Each world is declared with a keyword `:world` and encompasses a `:constant_map` stating what each constant refers to as a list of pairs (constant, entity), as well as a list of true ground `:atoms` (where the closed-world assumption holds). The actual world is defined with the `:actual_world` keyword. The indistinguishability relation for agents is specified with the `:edges` keyword. For each agent, a set of pairs of worlds in the relation is listed, whose reflexive-transitive closure defines the full relation. Finally, the `:goal` keyword declares the goal.
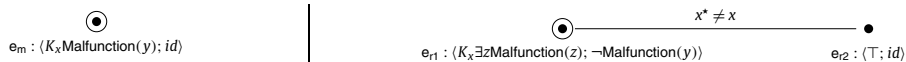

Fig. 14. *Left*: Malfunction(x, y), the action schema for agent x announcing that they know that y is malfunctioning. *Right*: Reboot(x, y), the action schema for agent x rebooting machine y, an action that x is permitted to do only if x knows that some malfunction is occurring, and which is done privately: other agents than x are uncertain about its execution.

```
;; machine-malfunction domain.
(define (domain machine-malfunction)
  (:types admin_agent_id - agent_id
          monitoring_agent_id - agent_id
          serial_number - machine_id
          machine_id
          agent_id)
  (:predicates (malfunction ?o - machine_id))
  (:action MALFUNCTION
    :agent ?s - monitoring_agent_id
    :parameters (?o - machine_id)
    (:actual_event em
      :precondition  (knows (?s) (malfunction ?o))
      :postcondition        (id))
    (:edge-conditions
      :em -- em (= ?x* ?x*)))
  (:action REBOOT
    :agent ?a - admin_agent_id
    :parameters (?n - serial_number)
    (:actual_event er1
      :precondition  (knows (?a) (exists (?x - object) malfunction (?x)))
      :postcondition        ((malfunction ?n if FALSE))
    (:event er2
      :precondition  (TRUE)
      :posttcondition  (id))
    (:edge-conditions
      :er1 -- er1 (= ?x* ?x*)
      :er2 -- er2 (= ?x* ?x*)
      :er1 -- er2 (not (= ?x* ?a)))
```

Fig. 15. A domain definition for the MM domain in a 'PDDL-style syntax'. Each action schema includes an `:agent` executing it as well as the schema's `:parameters` list. The possible events comprised in the action and their corresponding edge-conditions are listed next. The actual event is defined with the `:actual_event` keyword, under which pre- and postconditions are listed. Postconditions are given as a list of statements of the form "`ground_atom if condition`". A similar `:event` keyword is used for non-actual events. The keyword `edge-conditions` lists, for each pair of events, its edge-condition, via a line of the form `event1-event1(edge-condition)`. Both the `MALFUNCTION` and the `REBOOT` schemas have epistemic preconditions, which can be schematized thanks to the variable-indexed epistemic operators K_x.
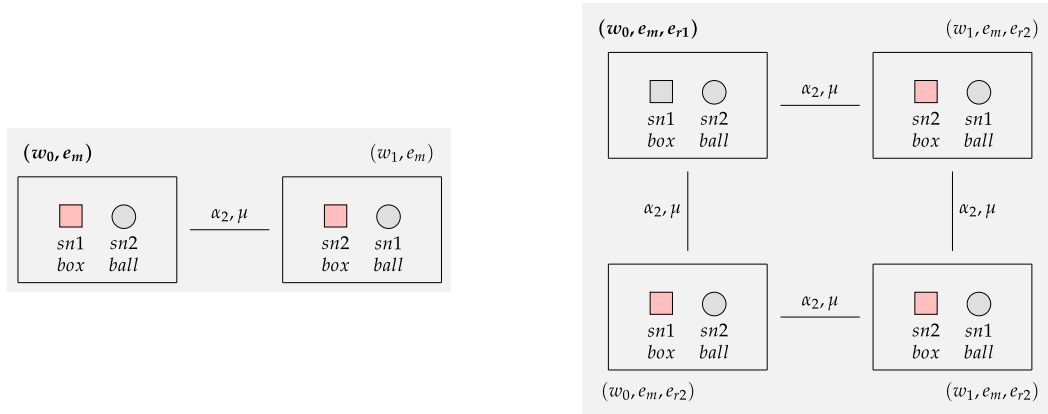

Fig. 16. Executing the plan π in s_0. *Left*: The epistemic state $s_1 := s_0 \otimes \text{Malfunction}(m1, box)$ reached after μ reports that *box* is malfunctioning. All agents learn that a machine is malfunctioning. However, α_2 and μ are still unsure about the serial number of the malfunctioning robot, but α_1 is not. *Right*: The epistemic state $s_2 := s_1 \otimes \text{Reboot}(a1, box)$ reached after admin agent α_1 reboots $sn1$. Since α_1 has rebooted $sn1$ privately, α_2 and μ still do not know that all machines are functioning. However, the goal is achieved: there is *some* agent (α_1) that knows that all machines are functioning.

6.2.5. Knowing who and alternative plans

There are other plans than π that would also solve the external epistemic planning task MM_task, but π achieves the goal in the fewest number of steps. The plan π is kept short by making admin α_1 act and finally witness the existence criteria in the goal φ_g. Admin α_1 is well-suited to this purpose because α_1 does not face uncertainty about the two names *box* and *sn1*. For the both names, α_1 *knows who* the names refer to, which in turn entails that in s_1, α_1 knows what machine to reboot.

Any plan where admin α_2, instead of α_1, acts, will necessarily be longer, because α_2 does not know is uncertain about what serial number belong to which machine, and must therefore reboot *both* machines. More specifically, then given s_0, any successful plan must start with the announcement Malfunction($m1, box$) to the effect that the admins know of a malfunction, required for them to execute reboots. From s_0, this results in s_1 of Fig. 16. From there, both Reboot($a2, sn1$) and Reboot($a2, sn2$) must be performed before α_2 knows that no machine is malfunctioning. Hence, the two shortest successful plans in which only α_2 reboots are

$$\pi' := \text{Malfunction}(m1, box), \text{Reboot}(a2, sn1), \text{Reboot}(a2, sn2)$$

$$\pi'' := \text{Malfunction}(m1, box), \text{Reboot}(a2, sn2), \text{Reboot}(a2, sn1)$$

Hence, the potential uncertainty introduced by the non-rigidity of constants may be consequential for epistemic planning. In this simple example, the non-rigid serial numbers refers to objects, but as also agent terms may be non-rigid, the presented framework allows modeling situations in which e.g. an agent is uncertain about whom a message must be delivered to, or planning in situations involving 'code names', known only to a strict subset of agents.

7. Languages for actions

We define a language for reasoning about actions, denoted $\mathcal{L}_{AM}$. This language extends the basic language $\mathcal{L}$ with *action modalities* with the form $[A, e]$, where A is an action model and e is an event from A. The language $\mathcal{L}_{AM}$ has formulas of the form $[A, e]\varphi$, which are interpreted as: 'after event e of action A occurs, φ is true'. This language extension allows us to include formulas mentioning other actions in the pre- and postconditions of some actions, as well as in goal formulas. It is thus possible to define, e.g., a goal such as: "Achieve a state in which it is impossible to perform an action that will result in φ". With finitely many actions described by the models $\mathcal{A} = \{A_1, \ldots, A_n\}$, such a formula would be $\bigwedge_{A \in \mathcal{A}, e \in E^A} [A, e]\neg\varphi$.

The grammar of $\mathcal{L}_{AM}$ is defined by double recursion, adapting a construction well known in the DEL literature (see, e.g., appendix H in [8] or [27]).

Definition 15. Let $\mathcal{L}_0 = \mathcal{L}$, and let AM_0 be the set of pointed action models whose precondition formulas are all from $\mathcal{L}_0$. Define $\mathcal{L}_{k+1}$ and AM_{k+1} as follows:

$$\varphi ::= r(t_1, ..., t_\ell) \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_t\varphi \mid \forall x\varphi \mid [A, e]\varphi$$

where $(A, e) \in AM_k$, and let $\mathbb{AM}_{k+1}$ be the set of pointed action models whose precondition formulas are all from $\mathcal{L}_{k+1}$. Lastly, define the **language** $\mathcal{L}_{AM}$ and the set of **action models** AM as

$$\mathcal{L}_{AM} := \bigcup_{k \in \mathbb{N}} \mathcal{L}_k, \ AM := \bigcup_{k \in \mathbb{N}} (AM_k)$$

As with the formulas from the static language $\mathcal{L}$, the formulas from $\mathcal{L}_{AM}$ are evaluated over epistemic models.

Definition 16. The satisfaction relation between epistemic models, assignments and formulas of $\mathcal{L}_{AM}$ is the smallest extension of $\vDash$ that satisfies:

$$M, w \vDash_v [A, e]\varphi \text{ iff } M, w \vDash_v \text{pre}(e) \text{ implies } M \otimes A, (w, e) \vDash_v \varphi$$

This extended satisfaction relation makes it possible to model-check conditions concerning actions. Given a pointed model (M, w), we may want to know whether a formula φ would hold after a sequence of pointed action models $(A_1, e_1), \ldots, (A_n, e_n)$ has been executed. This can of course be done by computing a sequence of product updates and checking whether $M \otimes A_1 \otimes \cdots \otimes A_n, (w, e_1, \ldots, e_n) \vDash \varphi$. But, equivalently, we can check whether the corresponding formula holds at (M, w), i.e., whether $M, w \vDash [A_1, e_1] \ldots [A_n, e_n]\varphi$. If φ is a goal formula and $(A_1, e_1), \ldots, (A_n, e_n)$ is a plan, then model-checking such a formula corresponds to *plan verification*. Section 8.2.2 gives so-called reduction axioms for $\mathcal{L}_{AM}$ formulas, showing that any formula containing an action modality can be expressed as a formula in the basic epistemic language $\mathcal{L}$. Consequently, plan verification could be treated as a problem of model-checking formulas of $\mathcal{L}$ in an initial state $s_0 = (M, w)$.

8. Axiomatic systems and metatheory

This section presents axiom systems for both static and dynamic term-modal logic. Metatheoretical results include soundness and completeness, frame characterizations, and decidability results. All proofs may be found in Appendix A.

Table 1
Axiom schemata for the minimal normal term-modal logic K.

First-order principles		Modal and interaction principles	
all propositional tautologies		$K_t(\varphi \to \psi) \to (K_t\varphi \to K_t\psi)$	K
$\forall x\varphi \to \varphi(y/x)$, for y free in φ	UE	$\forall x K_t \varphi \to K_t \forall x\varphi$, for x not occurring in t	BF
$t = t$, for $t \in \mathbb{T}$	Id	$(x \neq y) \to K_t(x \neq y)$	KNI
$(x = y) \to (\varphi(x) \to \varphi(y))$	PS		
$(c = c) \to \exists x(x = c)$	$\exists$Id	**Inference rules**	
$x \neq y$, if $\mathsf{t}(x) \neq \mathsf{t}(y)$	DD		
		From $\varphi, \varphi \to \psi$, infer ψ	MP
		From φ, infer $K_t\varphi$	KG
		From $\varphi \to \psi$, infer $\varphi \to \forall x\psi$, for x not free in φ	UG

8.1. Normal term-modal logic

8.1.1. Axiom system

Table 1 contains the axioms and inference rules for the term-modal logic K. Some are common first-order axioms, like *Universal Elimination* (UE), Reflexivity of Identity (Id), and the *Principle of Substitution* (PS). In a modal logical context, PS also has a modal feature: it is restricted to variables to allow for non-rigid constants. If PS is assumed also for constants, $(a = b) \to (K_t\varphi(a) \to K_t\varphi(b))$ becomes a theorem, valid only for rigid constants. *Existence of Identicals* ($\exists$Id) is included to ensure that all constants obtains an extension in the canonical models of Section A.1.2; *Divided Domain* (DD) is included to enforce type-distinction between variables logically rather than syntactically. The modal and interaction principles *Distribution* (K) and *Knowledge of Non-Identity* (KNI) are formulated as standard while the *Barcan Formula* (BF) has a restriction in the term-modal case; the Barcan Formula ensures constant domains: its validity implies non-growing domains, illustrated in the proof of soundness (Section A.1.1), and its converse implies non-shrinking domains (and is provable in K, cf. e.g. [43, p. 245]). Knowledge of Non-Identity reflects the rigidity of variables. The inference rules *Modus Ponens* (MP), *Knowledge Generalization* (KG) and *Universal Generalization* (UG) contain no surprises.

Notice that nothing in the language or axioms of K specify the number of agents in the system. The number of agents emerges as a definable frame characteristic, cf. Section 8.1.4.

8.1.2. Normality

In Section 8.1.3, we formally state that K is complete with respect to the class of all frames. The axioms and inference rules sufficient for a complete system are close to standard axiomatizations of first-order modal logic, cf. e.g. [17,31,43]. We take the close-to-standard format of the K axioms to indicate the innocence of the term-modal extensions of the syntax and semantics. This is further corroborated by the main result of this section, the *Canonical Class Theorem* on this page. In essence, the theorem shows that any closed extension of K is complete with respect to the class of its canonical models. The result thus justifies the following definition:

Definition 17. A set of formulas $\Lambda \subseteq \mathcal{L}$ is called a ***normal term-modal logic*** if, and only if, Λ contains all axioms of Table 1 and is closed under the Table 1's inference rules. The smallest normal term-modal logic is denoted K.

8.1.3. Canonical class theorem and completeness

In ordinary modal logic, each normal modal logic gives rise to a unique *canonical model*. In a similar manner, each normal term-modal logic Λ gives rise to a *class* of canonical models, one for each Λ-maximal consistent set. Section A contains the details of the construction, as well as the proof of the following main theorem:

Theorem 3 (*Canonical Class Theorem*). *Any normal term-modal logic Λ is strongly complete with respect to its canonical class.*

Mirroring the role of the Canonical Model Theorem of ordinary modal logic (see e.g. [12]), we obtain the following corollary to Theorem 3:

Corollary 1 (*Completeness*). *The logic K is strongly complete with respect to the class of all frames **F**.*

K is also sound with respect to the class of all frames. Section A.1.1 contains the formal statement and a proof sketch, with details given for the axiom K and the Barcan Formula.

8.1.4. Characterizing frame properties

The completeness result of Corollary 1 may be extended to more specific frame classes. Table 2 contains an overview of axiom schemata and the frame conditions they characterize. For illustration, proofs for 4 and N are given in Section A.1.3. From the Canonical Class Theorem and Table 2, completeness results for standard logics like KD45, S4 and S5 follow as corollaries. The principles N and M are special to our term-modal treatment. N and M define domain sizes. Nothing in the

Table 2
Term-modal axiom schemata and the frame conditions they characterize.

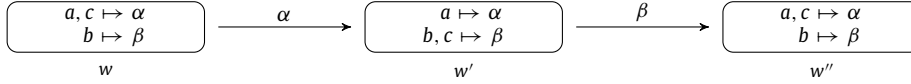| Axiom		Frame condition
$\forall x (K_x \varphi \rightarrow \varphi)$	T	Reflexive
$\forall x (\neg K_x \bot)$	D	Serial
$\forall x (K_x \varphi \rightarrow K_x K_x \varphi)$	4	Transitive
$\forall x (\neg K_x \varphi \rightarrow K_x \neg K_x \varphi)$	5	Euclidean
$\exists x_1, ..., x_n \left(\left(\bigwedge_{i \leq n} K_{x_i} \top \right) \wedge \left(\bigwedge_{i,j \leq n, i \neq j} x_i \neq x_j \right) \wedge \forall y \left(K_y \top \rightarrow \bigvee_{i \leq n} y = x_i \right) \right)$	N	$|D_{\text{agt}}| = n$
$\exists x_1, ..., x_m \left(\left(\bigwedge_{i,j \leq m, i \neq j} x_i \neq x_j \right) \wedge \forall y \left(\bigvee_{i \leq m} y = x_i \right) \right)$	M	$|D| = m$

Fig. 17. A transitive model invalidating $K_c(b=c) \rightarrow K_c K_c(b=c)$ at w. With $D_{\text{agt}} = \{\alpha, \beta\}$, all relations are transitive. The notation $a \mapsto \alpha$ specifies that α is the extension of the constant a in the given world. In w, it holds that $K_c(b=c)$ as $c \mapsto \alpha$ in w and $b, c \mapsto \beta$ in w' (in w, the knowledge that $(b=c)$ is held by agent α, as $c \mapsto \alpha$ in w). World w' satisfies $K_c(b \neq c)$ as $c \mapsto \beta$ in w' and $b \mapsto \beta, c \mapsto \alpha$ in w'' (in w', the knowledge that $(b \neq c)$ is held by β, as $c \mapsto \beta$ in w'). As a consequence, w' also satisfies $\neg K_c(b=c)$. Hence, w does not satisfy $K_c K_c(b=c)$—that the agent named c (i.e., α) knows that the agent named c (i.e., β) knows that $(b=c)$.

language or axioms of K specify the number of agents in the system: as in first-order logic, the domain size is by default left unspecified. In ordinary epistemic logic, it is common to assume a fixed, finite index set of agents. The domain size principles N and M similarly fixes domain sizes: N fixes the agent domain to size n. It uses the $K_x \top$-expressions to ensure that the bound variables are all of type agt. With the quantifications thus ranging only over agents, N specifies specifically the size of D_{agt}. This may be compared to M, which does not put constraints on the type of the bound variables, thereby fixing only the size of the joint agent–object domain D. For details concerning N, see Proposition 6 on 28.

The principles T, D, 4 and 5 deviate from their ordinary forms by being quantified. In standard modal logic, the formula

$$K_c \varphi \rightarrow K_c K_c \varphi \tag{1}$$

characterizes the class of transitive frames. This is not true here, as the constant c may be *non-rigid*[1]: see Fig. 17 for a transitive model invalidating (1). The invalidity arises as the extension of the c is not fixed under scope of operators: in the consequent, the accessibility relation which the inner occurrence of K_c quantifies over need not be the same as the accessibility relation of the outer. This makes the appeal to transitivity void.[2] The formulation in Table 2 avoids the non-rigidity problem, but does impose the criteria for all agents uniformly.

8.1.5. Heterogeneous agents

Though treating all agents uniformly is common in epistemic logic, one may desire heterogeneous agents. With the given setup, we do not believe this can be done at the level of frames. On the level of models, one option to this end is to attribute epistemic criteria to subgroups using predicates; a second is to introduce individual names. In either case, one may desire the defining criterion to be rigid. However, full rigidity is not definable in general as models may be disconnected. *Local rigidity*—invariance of interpretation over connected components—is definable by formulas of the forms

$$\forall x (r(x) \leftrightarrow \forall y K_y r(x)) \tag{2}$$

$$\exists x ((x=a) \wedge \forall y K_y (x=a)) \tag{3}$$

The validity of (2) and (3) characterize features of interpretations: (2) ((3), resp.) is valid in a model $M = (D, W, R, I)$ iff for all $w, w' \in W$, $(w, w') \in R_\alpha$ for some $\alpha \in D_{\text{agt}}$ implies $I(r, w) = I(r, w')$ ($I(a, w) = I(a, w')$, resp.). In conjunction with formulas of the forms

$$\forall x (r(x) \rightarrow (K_x \varphi \rightarrow K_x K_x \varphi)) \tag{4}$$

$$K_a \varphi \rightarrow K_a K_a \varphi \tag{5}$$

one may obtain some individuated control over relation properties.

---

[1] The non-rigidity of constants is reflected in K: *Knowledge of Identity* is provable for variables, but not for constants. I.e., K proves $(x = y) \rightarrow K_t(x = y)$, but not $(a = b) \rightarrow K_t(a = b)$.

[2] In his 1962 [39], Hintikka argues that $K_c \varphi \rightarrow K_c K_c \varphi$ intuitively is valid only if c knows that she is c; i.e., that she knows who c is. Hintikka argues that this is captured by $\exists x K_c(x = c)$, which makes c locally rigid for the agent: $I(c, w') = I(c, w)$ for all w' in $R_{I(c,w)}(w)$. Indeed, $\exists x K_c(x = c) \wedge K_c \varphi \rightarrow K_c K_c \varphi$ is valid on transitive frames.

Table 3
Axiom and rule schemata for the system AM.

Reduction axioms	
$[A, e]r(t_1, ..., t_n) \leftrightarrow (\text{pre}(e) \rightarrow \text{post}(e)(r(t_1, ..., t_n)))$	Action and atom
$[A, e]\neg\varphi \leftrightarrow (\text{pre}(e) \rightarrow \neg[A, e]\varphi)$	Action and negation
$[A, e](\varphi \wedge \psi) \leftrightarrow (([A, e]\varphi) \wedge ([A, e]\psi))$	Action and conjunction
$[A, e]K_t\varphi \leftrightarrow \bigwedge_{e' \in E}(Q(e, e')(x^\star \mapsto t) \rightarrow K_t[A, e']\varphi)$	Action and knowledge
$[A, e]\forall x\varphi \leftrightarrow (\text{pre}(e) \rightarrow \forall x[A, e]\varphi)$	Action and quantification (Dynamic Barcan)
$[A, e][A', e']\varphi \leftrightarrow [(A, e) \circ (A', e')]\varphi$	Action composition

Inference rules	
From φ, infer $[A, e]\varphi$	Action necessitation

8.1.6. Decidability

Let K_n and $\mathsf{K}_{n/m}$ be the smallest normal extensions of K with, respectively, the domain size axiom N, and both domain size axioms N and M under the proviso that $m > n$. K_n and $\mathsf{K}_{n/m}$ are then sound and complete with respect to, respectively, the class of all frames with exactly n agents, and the class of all frames with exactly n agents and exactly $m - n$ objects. These finite domain properties are used in the proof of items 1. and 2. of the below proposition, shown in Section A.1.4. Decidability results from the literature are discussed in Section 9.3.

Proposition 1. *Let $\mathsf{K}_{n/m}$, K_n and K be given in $\mathcal{L}$, based on the signature Σ. Let $\mathcal{L}_{\text{agt}} \subseteq \mathcal{L}$ contain all formulas containing only agent-terms, $t \in \mathsf{t}^{-1}(\text{agt})$.*

1. *For all $\varphi \in \mathcal{L}$, it is decidable whether $\vdash_{\mathsf{K}_{n/m}} \varphi$ or not.*
2. *a) For all $\varphi \in \mathcal{L}_{\text{agt}}$, it is decidable whether $\vdash_{\mathsf{K}_n} \varphi$ or not. b) In general, it is undecidable.*
3. *In general, it is undecidable whether $\vdash_{\mathsf{K}} \varphi$ or not.*

8.2. Dynamic term-modal logic

8.2.1. Axiom system

Table 3 contains the axioms and inference rules for the dynamic term-modal logic AM. In Section 8.2.2, we formally state that $\mathsf{K} + \mathsf{AM}$ is sound and complete with respect to the class of all frames. This completeness result may be extended to more specific frame classes, as was the case with K (see Section 8.1.4). The completeness proof for $\mathsf{K} + \mathsf{AM}$ is by translation, a well-known approach in DEL [7,8,27,71]. The axioms in AM are so-called *reduction axioms*, which enable the translation of formulas with action modalities into provably equivalent ones without any action modalities. Then completeness follows from the known completeness of the static logic K. For a detailed description of the reduction strategy to completeness, see e.g. [27].

The reduction axioms in AM are similar to those used in logics for epistemic actions, introduced by [7]. Naturally, as dynamic term-modal logic is first-order, there are reduction axioms for formulas involving quantifiers. Moreover, the axiom for formulas with the knowledge operator is non-standard. Unlike standard action models, the ones presented here are edge-conditioned and use variable substitutions, which require some modifications. A more detailed comparison of these axioms and standard ones is provided in Section 9.2. The *Action composition* axiom appeals to action models of the form $(A, e) \circ (A', e')$. This notation refers to the *composition* of (A, e) and (A', e'), defined following [29], but adapted to accommodate edge-conditions and first-order atoms:

Definition 18. Let $A_1 = (E_1, Q_1, \text{pre}_1, \text{post}_1)$ and $A = (E_2, Q_2, \text{pre}_2, \text{post}_2)$ be given. The **composition** of A_1 and A_2 is the action model $A_1 \circ A_2 = (E, Q, \text{pre}, \text{post})$ where

1. $E = E_1 \times E_2$.
2. $Q((e_1, f_1), (e_2, f_2)) = Q_1(e_1, f_1) \wedge [A_1, e_1]Q_2(e_2, f_2)$.
3. $\text{pre}(e_1, e_2) = \text{pre}(e_1) \wedge [A_1, e_1]\text{pre}(e_2)$.
4. $\text{dom}(\text{post}((e_1, e_2))) = \text{dom}(\text{post}_1(e_1)) \cup \text{dom}(\text{post}_2(e_2))$ and if $r(t_1, \ldots, t_k) \in \text{dom}(\text{post}((e_1, e_2)))$, then

$$\text{post}((e_1, e_2)(r(t_1, \ldots, t_k)) = \begin{cases} \text{post}_1(e_1)(r(t_1, \ldots, t_k)) & \text{if } r(t_1, \ldots, t_k) \notin \text{dom}(\text{post}_2(e_2)) \\ [A, e]\text{post}_2(e_2)(r(t_1, \ldots, t_k)) & \text{otherwise} \end{cases}$$

8.2.2. Soundness and completeness via reduction axioms

As anticipated in Section 8.2.1, $\mathsf{K} + \mathsf{AM}$ is sound and complete with respect to the class of all models.

Proposition 2 *(Soundness of* K + AM*).* K + AM *is sound with respect to the class of all models.*

The soundness of K + AM (Proposition 2) is established in the standard way, by showing the semantic validity of the reduction axioms and inference rules. The proof is straightforward and therefore omitted.

Completeness follows as a corollary from a number of lemmas that are presented in Section A.2.

Corollary 2 *(Completeness of* K + AM*).* K + AM *is complete with respect to the class of all models. Moreover, any extension of* K + AM *obtained by adding axioms characterizing frame conditions is complete with respect to the corresponding class of models.*

9. Related work

9.1. Epistemic planning

Several articles on multi-agent epistemic planning have appeared recently. The existing work can be organized along the following categories: modeling of epistemic planning, tractability and complexity, and implementation and applications.

On the modeling side, multiple articles have presented formalisms for multi-agent epistemic planning based on DEL [2, 15,57,91]. These models are very expressive, capturing several key aspects of multi-agent epistemic planning. These aspects include: epistemic actions and goals, higher-order knowledge and belief, partial observability, etc. A thorough comparison of the present framework with existing DEL formalisms is found in Section 9.2.

The rich expressivity of DEL comes at a cost, as planning problems specified in DEL are in general computationally difficult to solve (more on this below). This has partly motivated the introduction of simpler formalisms for epistemic planning. Some of these formalisms build on classical planning. The model in [70] extends STRIPS to allow knowledge declarations in preconditions and postconditions. The framework is however restricted to single-agent planning, does not support higher-order reasoning, and allows only a restricted form of quantification. The multi-agent planning frameworks in [45,60] follow a compilation approach, translating restricted fragments of epistemic planning into classical planning languages.

The approaches in [41,44] describe planning domains via a type of state-transition system extended with epistemic information, called a *concurrent epistemic game structure* (CEGS). This representation makes it easy to define multi-agent notions such as 'joint action' or 'multi-agent plan'. However, the representation inherits some of the well-known problems of transition-system models, including the lack of compact descriptions of actions and efficient heuristics that can avoid building the full state-transition system when planning (see [14] for a discussion of these and other limitations).

The non-DEL formalism that most closely resembles the approach of this paper is the epistemic game description language GDL-III [85]. The language is epistemic and first-order. A key feature of this language is that only what agents can see and do has to be defined. This is done via declarations that use the keywords **Sees** and **Does**, which loosely correspond to modalities. GDL-III has a simple syntax and allows compact specifications of actions. For instance, the following GDL-III rules [9] describe schematically communication actions which are similar to the ones from Example 5:

$$\textbf{Sees}(x, i_a) \Leftrightarrow \textbf{Does}(i, announce(z)) \land Obs(i, x)$$

$$\textbf{Sees}(y, \varphi) \Leftrightarrow \textbf{Does}(i, announce(\varphi)) \land Obs(i, y) \land Listen(i, y) \land \varphi$$

These two transition rules are interpreted as follows: if agent i announces z then any agent x observing i will receive the information i_a. Any agent y that observes i and listens to i will learn the content of the announcement, φ. Given the semantics of GDL-III, it follows that agents who only see i_a will know that i made an announcement but will not learn the content of the announcement. Agents who observe i, however, will know that φ must be true. Moreover, if an agent x observes agent i, does not listen to i, but knows that another agent y listens to i, then the semantics entails that x will know that y will know the content of the announcement after it has been made. The model is therefore schematic and context-sensitive, like the epistemic action schemas presented here. The syntax of GDL-III is simpler than that of DEL when it comes to representing actions. However, as noted in [30], specifying nested and higher-order knowledge is more difficult in GDL-III than in DEL, and the formalism requires more involved semantics. The work in [30] provides a detailed comparison of DEL and GDL-III, concluding that GDL-III offers a simpler syntax, while DEL provides simpler semantics. In [30], it is shown that large fragments of GDL-III and DEL are equally expressive by giving compilations between the two.

Concerning decidability and complexity, it was first shown in [15] that the general plan existence problem in propositional DEL planning (i.e., deciding whether a plan exists given a multi-agent planning task) is undecidable. In fact, the problem is undecidable with two agents only, no common knowledge, and no postconditions. In [51] it is shown that public actions are enough for undecidability when the initial state meets certain technical conditions. That paper also identifies an undecidable subclass of small epistemic planning problems comprising two agents, one action, six propositions and a fixed goal. The undecidability results straighforwardly apply also to the present framework.

Although the general problem is undecidable, several papers have identified decidable fragments of epistemic planning that are still reasonably expressive. Single-agent epistemic planning is decidable [15]. The multi-agent problem becomes decidable if actions are only allowed to have propositional preconditions (i.e., no epistemic formulas appear in the preconditions) [91]. The computational complexity of this fragment belongs to $(d + 1)$-ExpTime for a goal formula whose modal

depth is d. If actions are restricted to have propositional preconditions and no postconditions, the plan existence problem becomes PSpace-complete [20]. Stronger restrictions, such as allowing only private and public announcements, bring the complexity down to NP-complete [16]. As mentioned in Section 6, in [54] we show that single-agent epistemic planning and multi-agent planning with non-modal preconditions are also decidable in the term-modal, first-order case described in this paper, echoing the results for propositional planning in [15,91].

As for implementation and applications, a number of techniques and planners have been developed over the last decade. An approach that has gained popularity is the compilation approach. The idea involves choosing a suitably restricted fragment of DEL that can be encoded in a classical planning language. Epistemic problems are then translated into classical ones so that state-of-the-art planners can be used to solve them efficiently. The compilations rely on different restrictions. The system in [45] assumes that actions are public, physical actions are deterministic, and that all agents start with a common initial belief on the set of worlds that are possible. The paper adopts a centralized perspective, with planning done off-line from the viewpoint of a single agent. In [46], the authors extend this framework to cover on-line planning from the perspective of the agents themselves. The planner in Muise *et al.* [60] requires a finite depth of nesting of modalities and no disjunctions. Cooper *et al.* [22] use an encoding based on special variables describing what agents can see. The epistemic problems expressible with this restricted language are then encoded in PDDL and solved using the Fast Downward planner [38]. As mentioned before, the PKS system in [70] encodes epistemic planning using a STRIPS-like language. This language can describe single-agent, epistemic planning problems with conditional effects. The PKS system tries to solve these problems using an efficient but incomplete algorithm.

A small number of epistemic planners do not rely on compilation into classical planning. The system MEPK [42] performs multi-agent epistemic planning from the third-person viewpoint. The system can handle private actions and beliefs, as formalized with the modal logic KD45n. The system does not support arbitrary common knowledge but can deal with a weaker form of common knowledge. Finally, Le *et al.* [50] present two forward planners, called EFP and PG-EFP, for multi-agent epistemic planning. These planners can deal with unlimited nested beliefs, common knowledge, and epistemic goals when the number of worlds in the initial state is not too large.

9.2. Dynamic epistemic logic

There is a vast and excellent literature on both epistemic logic and dynamic epistemic logic to which the reader is referred for both technical and conceptual introductions—see e.g. [8,10,27,31,39,59,52]. The approach to modeling actions taken in this paper is based on the idea of *action models* applied using *product update* as introduced first by Baltag, Moss and Solecki [7]. The reduction axiom approach to proving completeness for logics with actions was first suggested by Plaza [71] for the case of *truthful public announcements*. Our approach is the same, but for general actions models. It is based on [6,8,27].

The format of the action models presented here differs mainly in four aspects from those introduced in [7]: our action models have *postconditions*; are *first-order* rather than propositional; accommodate *term-modal relations*; and have *conditioned edges*. Our approach to postconditions is inspired by [11,29]. From there, it is a straightforward generalization to alter pre- and postconditions to allow updates of first-order Kripke models. A substantial departure from the standard is the accommodation of term-modal relations and the edge-conditioning. The definition avoids two problems for term-modal action models—one pointed out by Kooi [47] and one concerning reduction axioms—by an adjustment of the propositional *edge-conditioned* action models of Bolander [13].[3]

In the standard definition, an action model A for index set of agents $\mathcal{I}$ consists of a finite set of events $E = \{e, ..., e'\}$ and a map $R : \mathcal{I} \to \mathcal{P}(E \times E)$, plus assignments of pre- and postconditions. In the term-modal treatment, the set $\mathcal{I}$ is a proper part of the semantics of state representations. Adding an operator $[A, e]$ to the language thus conflates syntax and semantics, Kooi points out. In considering reduction actions, we found that this problem runs deep. Consider the standard reduction axiom for the modal operator:

$$[A, e]K_i\varphi \leftrightarrow \left(pre(e) \to \bigwedge_{f:(e,f)\in R_i} K_i[A, f]\varphi \right) \tag{6}$$

In (6), the agent index i links the occurrences of the modal operator K_i with the relation R_i used in the quantifying conjunction. This link is broken in the term-modal treatment: the "i" indexing the operators is a syntactic term, while the "i" indexing the relation is an element of a domain of quantification. Without consulting an interpretation (or variable assignment), these two occurrences are unlinked: there is no guarantee that R_i is the relation used in evaluating $K_i\varphi$.

To resolve the conflation problem, Kooi defines action models where accessibility relations over events E are assigned to *groups* of agents on a per-application basis: With Φ a finite set of mutually inconsistent and jointly exhaustive formulas with free variable x, each pointed model (M, w) and variable valuation v defines a partition on the agent domain with cells

---

[3] Both approaches result in context-sensitive actions: the distinguishability of two events depends on model to be updated. See [13,52,76–78] for arguments to the effect that more context-sensitivity than what is present in standard action models is desirable.

$\{d \in D_{agt} : M, w \vDash_{v[x/d]} \varphi(x)\}$ for each $\varphi(x) \in \Phi$; each agent in such a cell (group) is assigned the same accessibility relation using a map $S : \Phi \to \mathcal{P}(E \times E)$. In effect, the action model makes no direct reference to the agent domain, thus avoiding the conflation problem.

Additionally, Kooi's definition yields a solution to the problem of unlinked indices as the relations of the action model may now be referred to using syntactical constructs. With $\Phi = \{P_1(x), ..., P_n(x)\}$, a suggestion for a reduction axiom could be

$$[A, e]K_t\varphi \leftrightarrow \left(pre(e) \to \bigwedge_{k \leq n} \left(P_k(t) \to \bigwedge_{f:(e,f)\in R(P_k)} K_t[A, f]\varphi \right) \right).$$

We obtain a similar solution by adjusting the edge-conditioned action model of Bolander [13]. In an edge-conditioned action model, whether two events are related for some agent $i \in \mathcal{I}$ is conditional on whether a given formula is satisfied in the pointed model on which the action is executed. Formally, each agent-edge pair is assigned a condition by a map $Q : \mathcal{I} \to (E \times E \to \mathcal{L})$.

Inspired by both Bolander and Kooi, we use a map $Q : E \times E \to \mathcal{L}$ where $Q(e, e')(x^\star)$ has exactly one free variable, $x^\star$. When the resulting action model is executed on a pointed model (M, w), an edge is present for an agent $\alpha \in D_{\text{agt}}$ if $M, w \vDash_{v[x^\star \mapsto \alpha]} Q(e, e')(x^\star)$. As the condition $Q(e, e')(x^\star)$ is a formula, this approach allows the formulation of reduction axioms, cf. Section 8.2.1.

Our version of the Q function and Kooi's approach S are equally general. Given an action model $(E, S, \text{pre}, \text{post})$ with $S : \Phi \to \mathcal{P}(E \times E)$, let $Q : E \times E \to \mathcal{L}$ be given by $Q(e, e') = \varphi$ such that $(e, e') \in S(\varphi)$. Then Q emulates S: for all models M, $M \otimes (E, S, \text{pre}, \text{post}) = M \otimes (E, Q, \text{pre}, \text{post})$. *Vice versa*, to emulate a map Q, for each $A \subseteq E \times E$, let

$$\varphi_A := \bigwedge_{\varphi \in Q(A)} \varphi \wedge \bigwedge_{\psi \in \Psi} \psi \tag{7}$$

with Ψ the largest subset of $\{\neg\varphi : \varphi \in Q(E \times E) \setminus Q(A)\}$ such that (7) is consistent. Then $S : \varphi_A \mapsto A$ for each $A \subseteq E \times E$ is a Kooi map that emulates Q. We opt for the edge-conditioned formulation due to its correspondence with the standard precondition maps $\text{pre} : E \to \mathcal{L}$.

Finally, note that both may emulate standard action models over classes of models where each agent α is designated by a rigid constant a_α (as is conceptually implied by identifying agents with indices). The standard map $R : D_{\text{agt}} \to \mathcal{P}(E \times E)$ may be emulated by the map $Q : E \times E \to \mathcal{L}$ with $Q(e, e') = \bigvee_{a_\alpha : (e,e')\in R_\alpha}(x = a_\alpha)$.

9.3. Term-modal logic

The term-modal treatment of epistemic operators as behaving both as modal operators and as first-order predicates was suggested already by von Wright in his 1951 [87], though the direction was not formally explored. Formally, Hintikka allowed the constructions in his 1962 [39], and the term-modal aspects are used in discussions concerning the validity of $K_a\varphi \to K_aK_a\varphi$, where Hintikka notes that the schema is only valid if *a knows who a is*, captured by $\exists x K_a(x = a)$ (see also Section 8.1.4 on frame characterizations). Semantically, Hintikka linked individuals and operators in [40] using world-relative first-order interpretations extended to assign alternatives to individuals in the domain of quantification, D. Work in philosophical logic followed Hintikka's term-modal syntax—even called "standard" by Carlson in 1988 [19]—but the semantic link did not pertain: [83] exemplifies a pseudo-use. Carlson enforced the semantic link, using a partial map $R : D \longrightarrow \mathcal{P}(W \times W)$ to assign accessibility relations to individuals. He further presents a Hintikka-style model set proof theory for a three-valued Kripke-style semantics with non-rigid terms, varying domains and reflexive relations, and shows completeness.

In computer science, a format similar to Carlson's is frequently used when giving the semantics for propositional epistemic logic, with the set of agents D treated as an *index set* instead of a domain of quantification, even in the first-order case: e.g., in Fagin *et al.*'s first-order treatment [31], in a formula like $K_{Alice}Governor(California, Pete)$, both $California$ and $Pete$ are first-order terms, but $Alice$ is not—$Alice$ is an agent. Here, then, agents and their names are *equated*.

The issue of equating agents and their names, and why this is unsatisfactory in many computer science applications, is discussed at length by Grove & Halpern [37] and Grove [36]. They identify the following inadequacies: systems that equate agents with their names cannot represent agent sets of non-fixed size, do not allow for reference to agent groups, for non-rigid names, nor for indexical and relative reasoning (using terms like "me" to express e.g. "the agent to the left of me"). In response, [37] develops a propositional epistemic logic with indexical reference obtained by evaluating formulas at agent-world pairs, on which [36] builds a first-order variant to additionally handle issues of *de dicto/de re*-like reference scope, as well as multi-naming of agents. The latter is in effect a variant of non-rigid constant, varying domain, term-modal logic with formulas evaluated at agent-world pairs. Further, the language contains two sorts, one for *agents* (like our agt terms), and one for *names*.[4] This allows explicit reasoning about *naming*. Adding a third sort to the present framework

---

[4] Such two sorts are also used by Rendsvig in a quantified, but not term-modal, epistemic logic analysis of semantic competence in relation to Frege's puzzle about identity [74,75].

would be unproblematic, but the indexical semantics would require in-depth re-working. Similarly, varying agent domains would require work, unless emulated by an existence predicate, cf. [32]. Beyond this, the present framework tackles the issues raised in [36,37]: agents and names are not equated by the use of (non-rigid) constants of sort `agt`, that additionally allow for multi-naming; agents groups may be denoted by predicates and relative properties by relations; finally, *de dicto/de re* distinctions are expressible using quantification. However, beyond the formal difference and similarities, we would find an in-depth philosophical comparison of the interpretation of the two frameworks interesting. In [53], we illustrate the system presented here with examples that touch on several of the involved issues.

One reason for sticking with ordinary modal operators even in a first-order setting is that term-modal operators adds design choices and possible complications, as discussed by Lomuscio & Colombetti in their early contribution to the term-modal literature [56]. In constructing a term-modal extension of multi-agent KD45 with non-rigid terms, they discuss how to evaluate formulas $B_a\varphi$ when a is not an agent denoting term. Intuitively, $B_a\varphi$ should be false, as only agents can truly hold beliefs, but—they remark—this would imply the invalidity of $B_a(\varphi \vee \neg\varphi)$. They conclude against a two-sorted approach, as a similar problem surfaces for formulas $B_a B_b\varphi$ when agent a believes that the term b denotes a non-agent.[5] Ultimately, Lomuscio & Columbetti opt for a *partial logic* with *truth-value gaps*, letting the truth-value of $B_a\varphi$ be *undefined* when a denotes a non-agent; they take a *valid* formula to be *sometimes satisfied*, but *never false*. Their semantics are constant domain, and each element is, at each world, assigned a set of *doxastic alternatives*; an element is an *agent in world w* if it is assigned a non-empty set. Hence, agenthood is *world-relative*. They present an axiom system—which includes a term-modal Barcan formula $\forall y(B_x\varphi(y)) \rightarrow B_x\forall y(\varphi(y))$ and quantified frame-characterizing formulas like $\forall x(B_x\varphi \rightarrow B_x B_x\varphi)$ like the present paper—and show soundness, citing [55] for details.

Bivalent systems are presented by Thalmann [84] and Fitting, Thalmann & Voronkov [33], with these two works coining the label 'term-modal logic'. In their setting, each world w is associated with an *inner domain* $D(w)$ of objects existing at w, with $D(w)$ a subset of the *outer domain D,* for all w. The inner domains are assumed *increasing*: if $wR_d w'$ for some $d \in D$, then $D(w) \subseteq D(w')$. Further, terms are assumed rigid and with an interpretation defined at every world ($I(c) \in D(w)$ for all $w \in W$). This combination seemingly[6] eliminates the need for truth-value gaps, but the problems raised by non-agents are not discussed. For several classic frame-conditions, [33,84] presents both sequent and tableau proof systems (K, D, T, K4, D4, S4).

Orlandelli & Corsi [62] also investigate sequent calculi for term-modal logics. Their semantics is more general as they omit the increasing domain requirement, and—as they also consider Euclidean frames—they also obtain completeness for more frame classes. The syntax is without constants, so the rigidity/non-rigidity dichotomy is non-applicable. The semantics are bivalent. The combination of varying domains and bivalent semantics is facilitated by the atomic formula satisfaction clause

$$M, w \vDash_v r(x_1, ..., x_n) \text{ iff } (v(x_1), ..., v(x_n)) \in I(r, w),$$

with $I(r, w) \subseteq D^n$ again with D the outer domain. E.g., with $I(=, w) = \{(d, d) \in D^2 : d \in D\}$, the formula $(x = x)$ is satisfied in (M, w) even if $v(x) \notin w$. However, as the quantifiers only range over the inner domain of worlds, the semantics oddly make $p(x) \wedge \forall y \neg p(y)$ satisfiable.

In [47], Kooi introduces a *dynamic* term-modal logic, including a first-ever first-order version of DEL action models. The language of [47] is first-order dynamic logic with wildcard assignment, but where the set of first-order terms is also the set of atomic programs, the models for which are constant agents-only domain with non-rigid terms (and very similar to our general case, but restricted to agents-only). This language is more expressive than ordinary term-modal logic. The first-order dynamic logic aspect implies that the validity problem is Π_1^1 complete, eliminating hope for a finitary proof system. However, the expressivity of the language allows the definition of a *non-rigid common knowledge*. If not for our two-sorted domain, our language and semantics could be seen as a special case of Kooi's. Kooi's action models are discussed in the next section.

Seligman & Wang [88] investigate a fragment Kooi's system. The fragment allows only basic assignment modalities to form a quantifier-free term-modal logic (without function symbols), a fragment rich enough to express *de dicto/de re* distinctions and *knowing who* constructions in a setting where names are not common knowledge. The main result is a complete axiomatization for the class of S5 models. As Barcan-like formulas are not included in the investigated language fragment but are the common characterizers of constant domain semantics, this result is quite non-standard. The authors also discuss decidability: providing no hard results, they conclude "We are not that far from the decidability boundary, if not on the wrong side."

Corsi & Orlandelli [23] introduce a generalization of term-modal syntax to be able to express the difference between *de dicto* and *de re* statements without invoking quantifiers. They introduce complex term-modal operators $|t : {}_x^c|p(x)$ with the

---

[5] This obstacle is avoided in the present paper by syntactically forcing all operator-subscripts to be of the agent-sort.

[6] Seemingly, as we are confused about the satisfaction clause for atomic formulas [33, Def. 7, It. 1], stating that $w, V \Vdash R(t_1, ..., t_n)$ iff $w \Vdash R(V(t_1), ..., V(t_n))$ with $V(t_i) \in D$, but no specification of the conditions for the right-hand condition, nor any specification of how the relation symbol R is assigned extension. However, if this is assumed settled as ordinarily (as in the present paper), the increasing domain assumption seems sufficient to obtain a well-behaved semantics, as is the case in ordinary first-order modal logic. See e.g. [34] for an introduction and [43] for details.

reading that t knows of c that (s)he is $p(x)$. These are interpreted over so-called *epistemic transition structures* with *double-domains*. The resulting *indexed epistemic logics* are further investigated in [25,24]. It would be interesting to know what the relationship is to the also expressive language of Kooi [47].

Where the domain of Kooi [47] consists only of agents, Rendsvig [73] introduces a model with a single-sorted language with non-rigid terms that denote elements in a constant domain containing both agents and objects. As in [56], this requires an *ad hoc* solution to the semantics of formulas $K_a\varphi$ when a denotes a non-agent. The solution used is to then interpret $K_a\varphi$ as a *global* modality. This preserves the bivalence of the systems while making all operators normal. As a result, [73] presents a canonical model theorem, facilitating completeness proofs for classic frame classes.

The semantics of this paper are based on Achen's [1], which in turn is a two-sorted refinement of [73]. What we consider an improvement of [1] over [73] is exactly the two-sorted approach: distinguishing between agent and object terms removes the need to define *ad hoc* semantics for knowledge operators indexed by non-agents. Taking a two-sorted approach eliminates the possibility of modeling agents that are uncertain about whether a given term refers to an agent or an object, but results in a system which we consider well-behaved.

Term-modal like, Naumov & Tao [61] present a propositional term-modal logic, but where operators may be indexed by sets of terms, making $\exists x K_{\{x,a\}}\varphi$ a formula. Such operators are given a *distributed knowledge* semantics in S5 models with constant agents-only domain and rigid terms for which a complete axiom systems is presented.

Sawasaki, Sano and Yamada [80] consider a term-modal syntax where operators are indexed by a sequence of terms making e.g. $\forall x \forall y K_{[x,y]} R(x, y)$ well-formed, with the intended deontic reading that x is obliged by y to ensure $R(x, y)$. They present complete axiom system and sequent calculi.

Sedlar [81] also uses rigid terms, agents-only constant domain semantics to represent an epistemic logic of evidence using a term-modal language as that presented here. Sedlar shows that his term-modal framework is able to emulate monotonic modal logics and epistemic logics with awareness, obtaining a decidability result for the fragment with no constants nor functions, but 0-ary predicates and single unary predicate.

Several other authors have also looked at decidability issues for varieties of term-modal logics. Kooi [47] points out that the monadic fragment of his system is undecidable by a result of Kripke [49]. As Kripke's result concerns first-order modal logic in general (see e.g. [43, p. 271 ff.]), it applies to broadly to term-modal logics, too. For term-modal logics, Padmanabha & Ramanujam [66] even show that the propositional fragment is undecidable. As decidable, they identify the monodic fragment (formulas using only one free variable in the scope of a modality). [64] considers model checking for the fragment over a restricted model class and [63] presents a translation of the monodic fragment (without identity) into FOML.

In [65], Padmanabha & Ramanujam further investigate a variable-free propositional bi-modal logic with implicit quantification, with formulas $[\forall]\varphi$ and $[\exists]\varphi$ asserting that along all (resp. some) accessibility relations φ is necessary. These variable-free formulas thus correspond to the propositional term-modal formulas $\forall x K_x\varphi$ and $\exists x K_x\varphi$. The relevant logic is shown decidable, to be bisimulation-invariant fragment of an appropriate two-sorted first-order logic, related to the 'bundled fragment' of term-modal logic. Model checking for the system is investigated in [68]. In [67] Padmanabha & Ramanujam, turn to the two variable fragment of term-modal logic, which they show decidable. The thesis [63] collects a selection of the mentioned results, and additionally presents a translation of TML without identity into propositional TML.

For their own system, Orlandelli & Corsi [62] show two fragments decidable, the first propositional with quantifiers and operators occurring only in pairs of the forms $\exists x[x]$ or $\forall x\langle x\rangle$. This fragment simulates non-normal monotone epistemic logics. The second fragment allows expressing 1-ary groups' higher-order knowledge about proposition symbols, e.g. with $\forall x(p(x) \to K_x(K_y q))$ an allowed formula. Also Pliuškevičius & Pliuškevičienė [72] treats a fragment of propositional term-modal logic, but with, term-modal operators for belief and mutual belief, allowing only pair-wise quantifier-operator nestings (e.g., for p a propositional atom, $\forall x B_x \exists y B_y p$ is well-formed, while $\forall x \exists y B_x B_y p$ is not). For their agents-only constant domain KD45 semantics, they present a terminating sequent calculus decision procedure. For further decidability results, it may be relevant to consult Shtakser [82], who investigates propositional modal languages includes quantification over modal operators and predicate symbols that take modal operators as arguments.

Beyond its main decidability result, Padmanabha & Ramanujam [67] also discusses translation of term-modal logic into first-order modal logic. In a setting with no constants or function symbols, the authors suggest a translation of TML into FOML with a single modality K and a new unary predicate P, inductively translating $K_x\varphi$ to $K(P(x) \to \varphi)$ and $\hat{K}_x\varphi$ to $\hat{K}(P(x) \land \varphi)$. [67] omits the details, but claims this translation produces FOML formulas equi-satisfiable with their TML originals. This suggests that completeness results for term-modal logics may also be shown indirectly via translation and application of well-known results for FOML (see e.g. [43]), instead of by the direct constructions found in the Appendix.[7] Whether a translation approach would work for the present framework is an open question, but we have reservations concerning the general applicability of the suggested translation.[8]

---

[7] We thank a reviewer for pointing this out.

[8] We hold a reservation as satisfiability is not generally preserved by the translation. In the class of TML models with exactly 2 agents (characterized by axiom N for $n = 2$) wlog called α and β, with constants a and b locally rigid, but non-identical (characterized by $\exists x \exists y((x \neq y) \land (x = a) \land (y = b) \land \forall z K_z((x = a) \land (y = b)))$), and satisfying for $i, j \in \{\alpha, \beta\}, i \neq j, \forall x, y, z \in W$, if $x R_i y$ and $x R_j z$, then $y R_i z$ (characterized by $\forall x \forall y(((x \neq y) \land \hat{K}_x\top \land \hat{K}_y\varphi) \to K_x\hat{K}_y\varphi)$), the formula $\exists x \exists y((x \neq y) \land \hat{K}_y\top \land \hat{K}_x\hat{K}_x\top \land K_x K_x\neg\hat{K}_x\top)$ is satisfiable. However, the translation of the latter is not satisfiable in the class of FOML models characterized by the translation of the three former.

10. Final remarks

We conclude with open questions we see in relation to epistemic planning, and a summary of the main contributions of the paper. The following are some possible avenues for future research:

1. *Decidability and complexity*. As presented in the literature review on epistemic planning with propositional DEL (Section 9.1), results exist concerning the undecidability of several classes of epistemic planning problems, but decidability and complexity results also exist. It is clear that the negative results apply in the richer setting of this paper. In [54], we show that some of the positive decidability results can be established in the decidable finite-agent setting of dynamic term-modal logic (i.e., decidability for single-agent planning and multi-agent planning with non-modal preconditions). It is an open question whether any other decidability results can be extended as well, and the complexity of first-order epistemic planning has not been studied.
2. *Reasoning about schematic actions*. In extension to defining first-order variants of action models, it was natural to define action schemas to obtain succinct action representations. These action schemas are however not described by the dynamic languages and logics introduced. We find it an interesting question how the languages and logics should be altered to obtain a logic of action schemas. Constructing such a logic could possibly draw connections to recent work on *Arbitrary Public Announcement Logic* and its generalizations, cf. e.g. [5,28].
3. *Supporting other planning features*. A possibly fruitful avenue for future research is to devise a first-order *probabilistic* DEL framework for probabilistic epistemic planning. In the standard planning literature, probabilistic PDDL is often used to support probabilistic effects, allowing the specification of Markov decision processes [90]. There is a rich literature on probabilistic propositional DEL on which a first-order setting for probabilistic epistemic planning could be based (for an overview, see [8, Appendix L]). Other well-known planning features, such as numeric fluents, temporal aspects, etc., could also be integrated.

Finally, we briefly recall what we see as the main contributions of the paper:

1. *A first-order dynamic epistemic logic*. The paper develops novel dynamics for a variant of term-modal logic with the addition of first-order action models. It thereby generalizes propositional DEL to a setting allowing full first-order epistemic reasoning about both objects and agents.
2. *A compact epistemic domain definition language*. As the epistemic planning formalism developed builds on first-order logic, it allows for a compact specification of domain dynamics via *epistemic action schemas*. Such schematization is inspired by that used in PDDL, and to the best of our knowledge, it provides the most compact representation of actions available in the DEL framework. The setting conservatively extends propositional DEL, in the sense that it contains it as a special case, inheriting the ingredients of the DEL planning framework.
3. *Expressive, yet decidable axiom systems for reasoning about epistemic actions*. On the reasoning side, the paper develops static and dynamic axiom systems that are well-behaved. Although the logical languages proposed are fairly expressive, it is shown that sound, complete and decidable systems exist for several natural classes of models.

Declaration of competing interest

Acknowledgements

Appendix A. Proof appendix

A.1. Term-modal logic

This section establishes the results stated in Section 8.1. The logic K is well-behaved, with standard techniques for establishing strong completeness carrying over from the propositional and quantified modal logic cases. Therefore, the section presents only proof strategy, with non-standard elements given special attention. Full details may be found in [1].

The involved notions are standard (see e.g. [12,17,43]), but we remark that a formula φ is *valid* over a class of frames $\boldsymbol{X}$ iff for every frame $F = (D, W, R) \in \boldsymbol{X}$, every interpretation I over F, every world $w \in W$ and every valuation v, it is the case that $M, w \vDash_v \varphi$. That φ is a *semantic consequence* of the formula-set Γ over a class $\boldsymbol{X}$ is written $\Gamma \vDash_{\boldsymbol{X}} \varphi$. For φ provable from the assumptions Γ in the logic Λ, write $\Gamma \vdash_\Lambda \varphi$. In both cases, when $\Gamma = \emptyset$, it is omitted.

A.1.1. Soundness

Proposition 3. *The system* K *is sound with respect to the class **F** of all frames: for all $\varphi \in \mathcal{L}$, if $\vdash_K \varphi$, then $\vDash_F \varphi$.*

Proof. The proof is standard: the axioms of K are shown valid over **F** and the rules of inference are shown to preserve validity. To give a feel, arguments follow for the K axiom and the Barcan Formula.

K: Let M be a model based on an arbitrary frame $F \in \mathbf{F}$, let $w \in M$ and let v be a valuation; let $K_t\top, \varphi, \psi \in \mathcal{L}$. To show that $M, w \vDash_v K_t(\varphi \to \psi) \to (K_t\varphi \to K_t\psi)$, assume $M, w \vDash_v K_t(\varphi \to \psi)$. As $K_t\top \in \mathcal{L}$, $[\![t]\!]_w^{I,v} \in D_{\mathrm{agt}}$ by assumption. Hence F contains an accessibility relation $R_{[\![t]\!]_w^{I,v}}$. Having fixed the accessibility relation going through the term t to the agent domain, the argument is standard: By the semantics of K_t, $M, w' \vDash_v \varphi \to \psi$ for every $w' \in M$ with $w' \in R_{[\![t]\!]_w^{I,v}}(w)$. Hence $M, w' \vDash_v \neg\varphi$ or $M, w' \vDash_v \psi$. If all such w' satisfies φ, $M, w \vDash_v K_t\varphi$; but then each w' must also satisfy ψ, so $M, w \vDash_v K_t\psi$, and hence $M, w \vDash_v K_t\varphi \to K_t\psi$. Else, some such w' satisfies $\neg\varphi$; then $M, w \vDash_v \neg K_t\varphi$, so $M, w \vDash_v K_t\varphi \to K_t\psi$.

BF: Let M, w, v, φ and t be as above. Pick a variable $x \neq t$ and assume that $M, w \vDash_v \forall x K_t\varphi$. Then for all x-variants v' of v, $M, w \vDash_v K_t\varphi$ (i.e., intuitively, if x is free in φ so that $K_t\varphi(x)$ defines a predicate, all elements in the $t(x)$-domain of w fall in this predicate's extension). From $M, w \vDash_v K_t\varphi$, it follows that for all $w' \in R_{[\![t]\!]_w^{I,v}}(w)$, $M, w' \vDash_{v'} \varphi$ (intuitively, as v' is an arbitrary x-variant v, all $t(x)$-elements *existing in* w' fall in the extension of $\varphi(x)$. This would not hold if elements could exist in w' that do not exist in w). As v' is an arbitrary x-variant of v, it follows that $M, w' \vDash_v \forall x\varphi$ (again, illegitimate *if* new elements could spring to existence). As w' was arbitrary from $R_{[\![t]\!]_w^{I,v}}(w)$, finally $M, w \vDash_v K_t \forall x\varphi$.   □

A.1.2. Completeness

This section establishes that the system K is *strongly complete* with respect to the class **F** of all frames. I.e.,

for all $\Gamma \subseteq \mathcal{L}$, for all $\varphi \in \mathcal{L}$, if $\Gamma \vDash_F \varphi$, then $\Gamma \vdash_K \varphi$.

This follows as a corollary of the section's main result, the *Canonical Class Theorem* (Theorem 3) which states that any normal term-modal logic is strongly complete with respect to its *canonical class*.

The theorem is established by appeal to the following well-known[9] proposition linking satisfaction and completeness:

Proposition 4. *A logic Λ is strongly complete with respect to a class of structure **S** iff every Λ-consistent set of formulas is satisfiable on some $s \in \mathbf{S}$.*

By this proposition, a completeness proof can be undertaken as an existence proof: For a consistent set of formulas Γ, a satisfying model from the appropriate class must be found. In the propositional case, one model is constructed for all consistent sets simultaneously, giving rise to the propositional *Canonical Model Theorem* (see e.g. [12]): any normal propositional modal logic is strongly complete with respect to its *canonical model*.

The present proof cannot rely on single canonical model. As variables are semantically rigid and any signature Σ includes identity, the same identity statements between variables are true across all worlds of any model-valuation pair. A canonical model defined as usual would not satisfy this: with consistent sets forming the basis of worlds, if two worlds are disconnected by all accessibility relations, then they need not satisfy the same identity statements between variables. Hence, a rigid variable valuation cannot be defined. Further, different K-consistent sets may give rise to different domains. Hence, non-constant domains result, and the construction is thus not of the appropriate class. Therefore, our construction is of a canonical model *per* consistent set, resulting in a *canonical class*.

The construction contains first-order aspects irrelevant in the propositional case and term-modal logical aspects irrelevant to the standard quantified case, but the approach is familiar: worlds are maximally consistent sets that bear witnesses, ensured constructable by Lindenbaum-like lemmas; domains are equivalence classes of variables induced by identity statements; and canonical accessibility relations, interpretation and valuation are defined as expected. That the canonical accessibility relations are well-defined requires an additional lemma, but a familiar Existence Lemma facilitates a familiar Truth Lemma, which in combination with the above Proposition 4 yields the main result.

A.1.2.1. Canonical worlds   Fix a signature $\Sigma = (\mathrm{V}, \mathrm{C}, \mathrm{R}, \mathrm{F}, \mathrm{t})$, its language $\mathcal{L}$ and a normal term-modal logic $\Lambda \subseteq \mathcal{L}$. When a set $\Gamma \subseteq \mathcal{L}$ is *maximal Λ-consistent* (defined as usual [12]), call Γ a Λ-*mcs*.

Maximal consistency does not suffice for a set to be a canonical world in the first-order case. It must also be ensured that whenever a formula of the form $\neg\forall x\varphi$ is included in Γ, then Γ must bear witness to this "falsity" of $\forall x\varphi$[10]:

---

[9] See e.g. [12, p. 194].
[10] Witnesses bearing is called the ∀-*property* in [43, p. 257]; that the set is *saturated* is also used in the literature.

Definition 19. A set $\Gamma \subseteq \mathcal{L}$ bears *witnesses* if for every $\varphi \in \mathcal{L}$, for every variable x, there is some variable y such that $\big(\varphi(y/x) \rightarrow \forall x \varphi\big) \in \Gamma$.

If a set Γ bear witnesses, then so does every super-set of Γ. If Γ is a Λ-mcs that bears witnesses and contains $\neg \forall x \varphi$, then for some $y \in \mathrm{V}$, $\neg \varphi(y/x) \in \Gamma$.

To ensure that every Λ-mcs can be extended to one bearing witnesses, countably infinite sets of both agent and object variables beyond those in V are needed. Define the extended signature Σ^+ as $(\mathrm{V}^+, \mathrm{C}, \mathrm{R}, \mathrm{F}, \mathrm{t}^+)$ where $\mathrm{V} \subseteq \mathrm{V}^+$, $\mathrm{t}^+(x) = \mathrm{t}(x)$ for all $x \in \mathrm{V} \cup \mathrm{C} \cup \mathrm{R} \cup \mathrm{F}$ and both $(\mathrm{t}^+)^{-1}(\mathrm{agt}) \cap \mathrm{V}^+ \backslash \mathrm{V}$ and $(\mathrm{t}^+)^{-1}(\mathrm{obj}) \cap \mathrm{V}^+ \backslash \mathrm{V}$ are countably infinite. Let $\mathcal{L}^+$ be the term-modal language based on Σ^+. Then $\mathcal{L} \subseteq \mathcal{L}^+$. The following two lemmas then ensure that the worlds of the canonical models are constructable:

Lemma 1 (Lindenbaum). *If $\Gamma \subseteq \mathcal{L}$ is Λ-consistent, then there is a Λ-mcs Γ' such that $\Gamma \subseteq \Gamma'$.*

Lemma 2 (Witnessed). *If $\Gamma \subseteq \mathcal{L}$ is Λ-consistent, then there is a set $\Gamma^+ \subseteq \mathcal{L}^+$ such that $\Gamma \subseteq \Gamma^+$ and Γ^+ bears witnesses.*

A.1.2.2. Canonical models To avoid the issue remarked in this section's introduction, a canonical model is defined per Λ-mcs, ensuring that all worlds share its *identity theory*:

Definition 20. The sets $\Gamma, \Gamma' \subseteq \mathcal{L}^+$ have the same **identity theory** if for all $x, y \in \mathrm{V}^+$, $(x = y) \in \Gamma$ iff $(x = y) \in \Gamma'$.

Definition 21. Let $\Lambda \subseteq \mathcal{L}$ be a normal term-modal logic. Let $\Gamma \subseteq \mathcal{L}$ be Λ-consistent and let $\Gamma^* \subseteq \mathcal{L}^+$ be maximal Λ-consistent, witness bearing and such that $\Gamma \subseteq \Gamma^*$ (existing by Lemmas 1 and 2). The **canonical model** for (Λ, Γ^*) is $M_{(\Lambda, \Gamma^*)} = (D, W, R, I)$ such that

1. $D := D_{\mathrm{agt}} \dot{\cup} D_{\mathrm{obj}} := \big\{ [x] : x \in (\mathrm{t}^+)^{-1}(\mathrm{agt}) \cap \mathrm{V}^+ \big\} \dot{\cup} \big\{ [y] : y \in (\mathrm{t}^+)^{-1}(\mathrm{agt}) \cap \mathrm{V}^+ \big\}$ where $[z] := \big\{ z' \in \mathrm{V}^+ : (z = z') \in \Gamma^* \big\}$.
2. W is the set of all maximal Λ-consistent, witness bearing sets of formulas from $\mathcal{L}^+$ that share identity theory with Γ^*.
3. $R : D_{\mathrm{agt}} \rightarrow \mathcal{P}(W \times W)$ such that for all $\alpha \in D_{\mathrm{agt}}$, $(w, w') \in R(\alpha)$ iff for every formula $K_x \varphi \in \mathcal{L}^+$ with $x \in \alpha$, if $K_x \varphi \in w$, then $\varphi \in w'$,
4. and
   (a) $I(r, w) = \Big\{ \big([x_1], ..., [x_n]\big) \in \prod_{i=1}^{len(\mathrm{t}(r))} D_{\mathrm{t}_i(r)} : r(x_1, ..., x_n) \in w \Big\}$, for all $r \in \mathrm{R}$;
   (b) $I(f, w) = \Big\{ \big([x_1], ..., [x_n]\big) \in \prod_{i=1}^{len(\mathrm{t}(f))} D_{\mathrm{t}_i(f)} : (f(x_1, ..., x_{n-1}) = x_n) \in w \Big\}$, for all $f \in \mathrm{F}$;
   (c) $I(c, w) = \big\{ \big([x]\big) \in D_{\mathrm{t}(c)} : (c = x) \in w \big\}$, for all $c \in \mathrm{C}$.

The **canonical valuation** v for (Λ, Γ^*) is given by $v(x) = [x]$ for all $x \in \mathrm{V}^+$.

A.1.2.3. Lemmas: uniformity, existence and truth The canonical model for (Λ, Γ^*) is a model for $\mathcal{L}$. Notably, the domain is well-defined by the identity theory sharing requirement and a two-partition by the inclusion of the DD axiom. Further, $I(c, w)$ is well-defined as for every world w, there exists some $x \in \mathrm{V}^+$ for which $(c = x) \in w$. See [1] for details. Foremost, the map R is well-defined, as is ensured by the following lemma:

Lemma 3 (Uniformity). *Let $K_x \varphi \in w \in W$ with $v(x) = \alpha$. Then for all $y \in \mathrm{V}^+$ for which $v(x) = v(y)$, also $K_y \varphi \in w$.*

Proof. Assume $K_x \varphi \in w \in W$ with $v(x) = \alpha$, and let $v(x) = v(y)$. Then $[x] = [y]$, so by identity theory sharing assumption, $(x = y) \in w'$ for every $w' \in W$; in particular, $(x = y) \in w$. By PS, $(x = y) \rightarrow \big(K_x \varphi \rightarrow K_y \varphi\big) \in w$. By MP, $\big(K_x \varphi \rightarrow K_y \varphi\big) \in w$ and by MP again, $K_y \varphi \in w$. $\square$

As in the propositional case, the proof of the Truth Lemma below relies on the below Existence Lemma. A proof for standard first-order modal logic may be found in [43]; details for term-modal logic may be found in [1].

Lemma 4 (Existence). *If $w \in W$ and $\neg K_x \varphi \in w$, then there exists a $w' \in W$ such that $(w, w') \in R_{[\![x]\!]_w^{l, v}}$ and $\varphi \in w'$.*

Lemma 5 (Truth). *For all $\varphi \in \mathcal{L}^+$, for all $w \in W$, and for the canonical v, $M_{(\Lambda, \Gamma^*)}, w \vDash_v \varphi$ iff $\varphi \in w$.*

Proof. The proof proceeds by induction on the complexity of φ. For the quantified formulas, appeal is made to w bearing witnesses. The negated modal case relies on the Existence Lemma. See [1] for full details. $\square$

A.1.2.4. Canonical class theorem The canonical models defined facilitate the application of Proposition 4 to conclude strong completeness of Λ with respect to its *canonical class*:

Definition 22. The ***canonical class*** of models for the normal term-modal logic Λ is the set $\boldsymbol{C}_\Lambda$ of canonical models $M_{(\Lambda,\Gamma^*)}$ for Λ-consistent $\Gamma \subseteq \mathcal{L}$.

Theorem 3 (*Canonical Class Theorem*). *Any normal term-modal logic Λ is strongly complete with respect to its canonical class.*

Proof. By Proposition 4, it suffices to find for each Λ-consistent set Γ some $s \in \boldsymbol{C}_\Lambda$ that satisfies Γ. One such is $(M_{(\Lambda,\Gamma^*)}, \Gamma^*)$, which exists by the Lindenbaum and Witnessed Lemmas. As $\Gamma \subseteq \Gamma^*$, the Truth Lemma ensures that $(M_{(\Lambda,\Gamma^*)}, \Gamma^*) \vDash_v \Gamma$ for v the canonical valuation. $\square$

Corollary 1 (*Completeness*). *The logic K is strongly complete with respect to the class of all frames $\boldsymbol{F}$.*

Proof. A frame $F \in \boldsymbol{F}$ that satisfies the K-consistent set Γ is the frame of the canonical model $M_{(K,\Gamma^*)}$: Γ is satisfied at Γ^* under the canonical valuation. $\square$

A.1.3. Frame characterization proofs

For illustrative purposes, we show two of the claims made in Table 2, Section 8.1.4.

Proposition 5. *For $\varphi \in \mathcal{L}$, $\forall x(K_x\varphi \to K_xK_x\varphi)$ is valid on the frame $F = (D, W, R)$ if, and only if, $R(\alpha)$ is transitive for every $\alpha \in D_{\mathrm{agt}}$.*

Proof. $\Leftarrow$: Let M be build on the frame F in which R_α is transitive for all $\alpha \in D_{\mathrm{agt}}$. Let v be an arbitrary valuation and assume $M, w \vDash_v K_x\varphi$. Then $M, w' \vDash_v \varphi$ for all $w' \in R_{v(x)}(w)$. For a contradiction, assume $M, w \vDash_v \neg K_xK_x\varphi$. Then there exists a $w^* \in R_{v(x)}(w)$ such that $M, w^* \vDash_v \neg K_x\varphi$, and hence there exists a $w^{**} \in R_{v(x)}(w^*)$ such that $M, w^{**} \vDash_v \neg\varphi$. But $R_{v(x)}$ is transitive, so $w^{**} \in R_{v(x)}(w)$. Hence w^{**} satisfies both φ and $\neg\varphi$. On pain of contradiction, $M, w \vDash_v K_xK_x\varphi$. As v was arbitrary, $M, w \vDash_v \forall x(K_x\varphi \to K_xK_x\varphi)$. $\Rightarrow$: By contraposition. $\square$

Proposition 6. *The formula $\exists x_1, ..., x_n \left(\left(\bigwedge_{i \leq n} K_{x_i}\top \right) \wedge \left(\bigwedge_{i,j \leq n, i \neq j} x_i \neq x_j \right) \wedge \forall y \left(K_y\top \to \bigvee_{i \leq n} y = x_i \right) \right)$ is valid on the frame $F = (W, D, R)$ if, and only if, $|D_{\mathrm{agt}}| = n$.*

Proof. Notice first that the formula, call it N, is only well-formed iff the variables $x_1, ..., x_n, y$ are all of type $\mathtt{agt}$, ensured by them appearing as modal operator subscripts. This ensures that the quantifications range only over D_{agt}.

$\Leftarrow$: Assume given a pointed model (M, w) build on a frame $F = (W, D, R)$ with $|D_{\mathrm{agt}}| = n$. Assume D_{agt} enumerated such that $D_{\mathrm{agt}} = \{\alpha_1, ..., \alpha_n\}$. Let v be an arbitrary valuation. We argue that $M, w \vDash_v N$. Let v' be the valuation identical to v on all points, except for each $i \leq n$, $v'(x_i) = \alpha_i$. Then $M, w \vDash_{v'} \left(\left(\bigwedge_{i \leq n} K_{x_i}\top \right) \wedge \left(\bigwedge_{i,j \leq n, i \neq j} x_i \neq x_j \right) \wedge \forall y \left(K_y\top \to \bigvee_{i \leq n} y = x_i \right) \right)$, as it satisfies each conjunct: First, $M, w \vDash_{v'} \bigwedge_{i \leq n} K_{x_i}\top$, trivially. Second, $M, w \vDash_{v'} \bigwedge_{i,j \leq n, i \neq j} x_i \neq x_j$ as $v'(x_i) \neq v'(x_j)$ for all $i, j \leq n, i \neq j$, by construction of v'. Third and finally, $M, w \vDash_{v'} \forall y \left(K_y\top \to \bigvee_{i \leq n} y = x_i \right)$: as y is of type $\mathtt{agt}$, for any y-variant v'' of v', $v'' \in D_{\mathrm{agt}}$, but then $v''(y) = v''(x_i)$ for some $i \leq n$, by construction of v', satisfying the antecedent.

$\Rightarrow$: Assume given a pointed model (M, w) build on a frame $F = (W, D, R)$ with $|D_{\mathrm{agt}}| \neq n$. Let v be an arbitrary valuation. We argue that not $M, w \vDash_v N$, as (M, w) will falsify either the second or the third conjunct (the first conjunct is satisfied: as each variable x_i is of type $\mathtt{agt}$ for all $i \leq n$, each $K_{x_i}\top$ is satisfied at w under any valuation). If $|D_{\mathrm{agt}}| < n$, then under any valuation v, (M, w) will falsify the second conjunct: as each variable x_i is of type $\mathtt{agt}$, $v(x_i) \in D_{\mathrm{agt}}$ for all $i \in 1, ..., n$. But then $v(x_i) = v(x_j)$ for at least two $i, j \leq n, i \neq j$. But then $M, w \vDash_v x_i = x_j$, contrary to the second conjunct. If $|D_{\mathrm{agt}}| > n$, then under any valuation v, (M, w) will falsify the third conjunct, as there exists a y-variant v' of v such that $v'(y) \neq v(x_i)$ for any $i \leq n$. The existence of this y-variant v' is ensured by $|D_{\mathrm{agt}}| > n$, which implies that $D_{\mathrm{agt}}/v(x_i): i \leq n \neq \emptyset$, so that we can assume $v'y \in D_{\mathrm{agt}}/v(x_i): i \leq n$. $\square$

A.1.4. Decidability

Proposition 7. *Let $K_{n/m}$, K_n and K be given in $\mathcal{L}$, based on the signature Σ. Let $\mathcal{L}_{\mathrm{agt}} \subseteq \mathcal{L}$ contain all formulas containing only agent-terms, $t \in \mathtt{t}^{-1}(\mathtt{agt})$.*

1. *For all $\varphi \in \mathcal{L}$, it is decidable whether $\vdash_{K_{n/m}} \varphi$ or not.*
2. *a) For all $\varphi \in \mathcal{L}_{\mathrm{agt}}$, it is decidable whether $\vdash_{K_n} \varphi$ or not. b) In general, $\vdash_{K_n} \varphi$ is undecidable.*
3. *In general, $\vdash_K \varphi$ is undecidable.*

Proof. 1. $\mathsf{K}_{n/m}$ is sound and complete w.r.t. $\boldsymbol{F}_{n/m}$. To check the validity of any $\varphi \in \mathcal{L}$ over $\boldsymbol{F}_{n/m}$ is a finite procedure: Up to isomorphism, all $F \in \boldsymbol{F}_{n/m}$ share domain $D = D_{\mathrm{agt}} \dot{\cup} D_{\mathrm{obj}}$, $|D_{\mathrm{agt}}| = n$, $|D_{\mathrm{obj}}| = (m-n)$. There are finitely many non-logical symbols in φ; symbols not in φ are irrelevant to its satisfaction. With D fixed, any $w \in F$ will be assigned one of finitely many extensions of φ's non-logical symbols: thus, the maximal set of distinct φ-relevant worlds W_φ is finite. As φ has modal depth k, whether $M, w \vDash_v \varphi$ depends on at most all worlds within k steps from w. Checking whether $M, w \vDash_v \varphi$ is thus a finite procedure for all formulas given the finiteness of D. Finally, up to bisimulation, the set of graphs over W_φ and $\{R(\alpha), \alpha \in D_{\mathrm{agt}}\}$ with maximal path length k is finite: hence, the set of needed to be checked pointed models is finite. **2a.** For any $\varphi \in \mathcal{L}_{\mathrm{agt}}$, φ is a theorem of K_n iff it is a theorem of $\mathsf{K}_{n/m}$, for any $m > n$. For such φ, to determine whether $\vdash_{\mathsf{K}_n} \varphi$, we can thus check whether $\vdash_{\mathsf{K}_{n/n+1}} \varphi$, which is decidable by 1. **2b** and **3.** General undecidability for K_n and K follows as both contain unrestricted first-order logic for the arbitrary object domain. $\square$

A.2. Dynamic term-modal logic: completeness through translation

The completeness proof for the dynamic logic $\mathsf{K} + \mathsf{AM}$ is based on a reduction argument. The argument relies on the existence of so-called reduction axioms for the dynamic language $\mathcal{L}_{AM}$. The axioms used for this specific proof are listed in Table 3 and can be used to translate every formula from the dynamic language $\mathcal{L}_{AM}$ into a provably equivalent $\mathcal{L}$-formula. Given this translation, the completeness of the dynamic logic follows from the known completeness of the static logic K, established in Corollary 1. The building blocks of the specific reduction argument required to prove completeness for $\mathsf{K} + \mathsf{AM}$ are provided below.

First, we provide a translation that by finite iterative application to any formula in the dynamic language $\mathcal{L}_{AM}$ results in a formula from the static language $\mathcal{L}$. The translation is left-to-right: a formula occurring on the left is translated to the formula on the right.

Definition 23. The **translation** $\tau : \mathcal{L}_{AM} \to \mathcal{L}_{AM}$ is defined as follows:

$$\tau((t_1 = t_2)) = (t_1 = t_2)$$
$$\tau(r(t_1, ..., t_n)) = r(t_1, ..., t_n)$$
$$\tau(\neg\varphi) = \neg\tau(\varphi)$$
$$\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$$
$$\tau(K_t\varphi) = K_t\tau(\varphi)$$
$$\tau(\forall x\varphi) = \forall x\tau(\varphi)$$
$$\tau([A, e]r(t_1, ..., t_n)) = \tau(\mathsf{pre}(e) \to \mathsf{post}^A(e)(r(t_1, ..., t_n)))$$
$$\tau([A, e]\neg\varphi) = \tau(\mathsf{pre}(e) \to \neg[A, e]\varphi)$$
$$\tau([A, e](\varphi \wedge \psi)) = \tau([A, e]\varphi \wedge [A, e]\psi)$$
$$\tau([A, e]K_t\varphi) = \tau(\mathsf{pre}(e) \to \bigwedge_{e' \in E^A} (Q(e, e')(x^\star \mapsto t)K_t[A, e']\varphi))$$
$$\tau([A, e]\forall x\varphi) = \tau(\mathsf{pre}(e) \to \forall x[A, e]\varphi)$$
$$\tau([A, e][A', e']\varphi) = \tau([A, e \circ A', e']\varphi)$$

Next, we adapt the formula complexity function introduced by [27].

Definition 24. The **complexity** $c : \mathcal{L}_{AM} \to \mathbb{N}$ is defined as follows, where $\mathtt{GA}(\mathcal{L})$ abbreviates $\mathtt{GroundAtoms}(\mathcal{L})$:

$$c(r(t_1, ..., t_n)) = 1$$
$$c(\neg\varphi) = 1 + c(\varphi)$$
$$c(\varphi \wedge \varphi') = 1 + \max(c(\varphi), c(\varphi'))$$
$$c(K_t\varphi) = 1 + c(\varphi)$$
$$c(\forall x\varphi) = 1 + c(\varphi)$$
$$c([A, e]\varphi) = (4 + c(A)) \cdot c(\varphi)$$
$$c(A) = \max \left(\bigcup_{e, e' \in E, r(t_1, ..., t_n) \in \mathtt{GA}(\mathcal{L})} \{c(\mathsf{pre}^A(e))\} \cup \{c(\mathsf{post}^A(e)(r(t_1, ..., t_n)))\} \cup \{c(Q(e, e'))\} \right)$$

A standard ordering lemma ensures that the right side of a given reduction axiom is indeed less complex than the left side.

Lemma 6. *For all φ, ψ and χ:*

1. $c(\psi) \geq c(\varphi)$ *if* $\varphi \in Sub(\psi)$ *(where* $Sub(\psi)$ *is the set of subformulas of* ψ*)*
2. $c([A, e]r(t_1, ..., t_n)) > c(\mathsf{pre}(e) \rightarrow \mathsf{post}(e)(r(t_1, ..., t_n)))$
3. $c([A, e]\neg\varphi) > c(\mathsf{pre}(e) \rightarrow \neg[A, e]\varphi))$
4. $c([A, e](\varphi \wedge \psi)) > c(([A, e]\varphi \wedge ([A, e]\psi))$
5. $c([A, e]K_t\varphi) > c(\mathsf{pre}(e) \rightarrow \bigwedge_{e' \in E}(Q(e, e')(x^\star \mapsto t)K_t[A, e']\varphi))$
6. $c([A, e]\forall x\varphi) > c(\mathsf{pre}(e) \rightarrow \forall x[A, e]\varphi)$
7. $c([A, e][A', e']\varphi) > c([A, e \circ A', e']\varphi)$

Proof. The proofs are straightforward, along the lines of those provided in [27, Chapter 7]. □

The complexity function c induces an ordering of $\mathcal{L}_{AM}$ formulas which is used to prove the following Lemma, stating that the two sides of a reduction axiom are indeed provably equivalent.

Lemma 7. *For all $\varphi \in \mathcal{L}_{AM}$:* $\vdash_{\mathsf{K+AM}} \varphi \leftrightarrow \tau(\varphi)$.

Proof. The proof is by induction on the complexity $c(\varphi)$. It is similar to the one provided in [27, Chapter 7]. □

The completeness of $\mathsf{K + AM}$ (Corollary 2) follows from the soundness of the dynamic proof system, Lemma 7 and the completeness of the static sub-system (Corollary 1). The argument, which is standard, is as follows.

Proposition 8. $\vDash \varphi$ *implies* $\vdash_{\mathsf{K+AM}} \varphi$, *for all* $\varphi \in \mathcal{L}_{AM}$.

Proof. Suppose $\vDash \varphi$. Since $\vdash_{\mathsf{K+AM}} \varphi \leftrightarrow \tau(\varphi)$ (Lemma 7), we have $\vDash \varphi \leftrightarrow \tau(\varphi)$ by the soundness of the proof system $\mathsf{K + AM}$. Thus $\vDash \tau(\varphi)$. The formula $\tau(\varphi)$ does not contain any action model modalities. Given $\vDash \tau(\varphi)$, by the completeness of K (Corollary 1), it follows that $\vdash_{\mathsf{K}} \tau(\varphi)$. As K is a subsystem of $\mathsf{K + AM}$, we thus have $\vdash_{\mathsf{K+AM}} \tau(\varphi)$. Since $\vdash_{\mathsf{K+AM}} \varphi \leftrightarrow \tau(\varphi)$ and $\vdash_{\mathsf{K+AM}} \tau(\varphi)$, it follows that $\vdash_{\mathsf{K+AM}} \varphi$. □

The completeness result for any system extending $\mathsf{K + AM}$ with frame-characterizing axioms follows from the same type of argument.

References

[1] A. Achen, Putting the Agents Back in the Domain: A Two-Sorted Term-Modal Logic, Bachelor thesis, University of Copenhagen, May 2017.
[2] M.B. Andersen, T. Bolander, M.H. Jensen, Conditional epistemic planning, in: Logics in Artificial Intelligence, Springer, 2012, pp. 94–106.
[3] G. Aucher, An internal version of epistemic logic, Stud. Log. 94 (1) (2010) 1–22.
[4] G. Aucher, T. Bolander, Undecidability in epistemic planning, in: IJCAI-13, 2013, pp. 27–33.
[5] P. Balbiani, A. Baltag, H. v. Ditmarsch, A. Herzig, T. Hoshi, T. de Lima, What can we achieve by arbitrary announcements?: A dynamic take on Fitch's knowability, in: TARK '07: Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge, ACM, 2007, pp. 42–51.
[6] A. Baltag, L.S. Moss, Logics for epistemic programs, Synthese 139 (2) (2004) 165–224.
[7] A. Baltag, L.S. Moss, S.S. Solecki, The logic of public announcements, common knowledge, and private suspicions (extended abstract), in: TARK '98: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge, Morgan Kaufmann Publishers, 1998, pp. 43–56.
[8] A. Baltag, B. Renne, Dynamic epistemic logic, in: E.N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, winter 2016 edition, Metaphysics Research Lab, Stanford University, 2016.
[9] C. Baral, T. Bolander, H. van Ditmarsch, S. McIlrath, Epistemic planning (Dagstuhl seminar 17231), Dagstuhl Rep. 7 (6) (2017) 1–47.
[10] J. van Benthem, Logical Dynamics of Information and Interaction, Cambridge University Press, 2011.
[11] J. van Benthem, J. van Eijck, B. Kooi, Logics of communication and change, Inf. Comput. 204 (11) (2006) 1620–1662.
[12] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, 2001.
[13] T. Bolander, Seeing is believing: formalising false-belief tasks in dynamic epistemic logic, in: A. Herzig, E. Lorini (Eds.), European Conference on Social Intelligence, ECSI 2014, in: CEUR Workshop Proceedings, vol. 1283, 2014, pp. 87–107.
[14] T. Bolander, A gentle introduction to epistemic planning: the DEL approach, in: 9th Workshop on Methods for Modalities, vol. 243, 2017, pp. 1–22.
[15] T. Bolander, M.B. Andersen, Epistemic planning for single- and multi-agent systems, J. Appl. Non-Class. Log. 21 (1) (2011) 9–34.
[16] T. Bolander, M.H. Jensen, F. Schwarzentruber, Complexity results in epistemic planning, in: IJCAI-15, 2015, pp. 2791–2797.
[17] T. Brauner, S. Ghilardi, First-order modal logic, in: F.W.P. Blackburn, J. van Bentham (Eds.), Handbook of Modal Logic, Elsevier Science, 2007.
[18] M. Brenner, A multiagent planning language, in: Proc. of the Workshop on PDDL, ICAPS, vol. 3, 2003, pp. 33–38.
[19] L. Carlson, Quantified Hintikka-style epistemic logic, Synthese 74 (1988) 223–262.
[20] T. Charrier, B. Maubert, F. Schwarzentruber, On the impact of modal depth in epistemic planning, in: IJCAI-16, 2016, pp. 1030–1036.
[21] T. Charrier, F. Schwarzentruber, A succinct language for dynamic epistemic logic, in: Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 123–131.
[22] M.C. Cooper, A. Herzig, F. Maffre, F. Maris, P. Régnier, A simple account of multi-agent epistemic planning, in: ECAI, 2016, pp. 193–201.
[23] G. Corsi, E. Orlandelli, Free quantified epistemic logics, Stud. Log. 101 (6) (Dec 2013) 1159–1183.

[24] G. Corsi, E. Orlandelli, Sequent calculi for indexed epistemic logics, in: ARQNL 2016: Automated Reasoning in Quantified Non-Classical Logics, in: CEUR-WS, vol. 1770, 2016, pp. 21–35.
[25] G. Corsi, G. Tassi, A new approach to epistemic logic, in: E. Weber, D. Wouters, J. Meheus (Eds.), Logic, Reasoning, and Rationality, Springer Netherlands, Dordrecht, 2014, pp. 27–44.
[26] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic epistemic logic with assignment, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, 2005, pp. 141–148.
[27] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic Epistemic Logic, Synthese Library, vol. 337, Springer, 2007.
[28] H. van Ditmarsch, W. van der Hoek, L.B. Kuijer, Fully arbitrary public announcements, in: L. Beklemishev, S. Demri, A. Máté (Eds.), Advances in Modal Logic, vol. 11, 2018, pp. 252–267.
[29] H. van Ditmarsch, B. Kooi, Semantic results for ontic and epistemic change, in: G. Bonanno, W. van der Hoek, M. Wooldridge (Eds.), Logic and the Foundations of Game and Decision Theory, LOFT 7, in: Texts in Logic and Games, vol. 3, Amsterdam University Press, 2008, pp. 87–117.
[30] T. Engesser, R. Mattmüller, B. Nebel, M. Thielscher, Game description language and dynamic epistemic logic compared, in: International Joint Conferences on Artificial Intelligence Organization, IJCAI-18, 2018.
[31] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Reasoning About Knowledge, The MIT Press, 1995.
[32] M. Fitting, R.L. Mendelsohn, First-Order Modal Logic, Springer, 1999.
[33] M. Fitting, L. Thalmann, A. Voronkov, Term-modal logics, Stud. Log. 69 (2001) 133–169.
[34] L.T.F. Gamut, Logic, Language and Meaning. Volume 2: Intensional Logic and Logical Grammar, The University of Chicago Press, 1996.
[35] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, Elsevier, 2004.
[36] A.J. Grove, Naming and identity in epistemic logic Part II: a first-order logic for naming, Artif. Intell. 74 (2) (1995) 311–350.
[37] A.J. Grove, J.Y. Halpern, Naming and identity in epistemic logics Part I: the propositional case, J. Log. Comput. 3 (4) (1993) 345–378.
[38] M. Helmert, The fast downward planning system, J. Artif. Intell. Res. 26 (2006) 191–246.
[39] J. Hintikka, Knowledge and Belief: An Introduction to the Logic of the Two Notions, 2nd edition, College Publications, 2005.
[40] J. Hintikka, Semantics for propositional attitudes, in: J.W. Davis, D.J. Hockney, W.K. Wilson (Eds.), Philsophical Logic, in: Synthese Library, Springer, 1969, pp. 21–45.
[41] W. van der Hoek, M. Wooldridge, Tractable multiagent planning for epistemic goals, in: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, ACM, 2002, pp. 1167–1174.
[42] X. Huang, B. Fang, H. Wan, Y. Liu, A general multi-agent epistemic planner based on higher-order belief change, in: IJCAI-17, 2017, pp. 1093–1101.
[43] G. Hughes, M. Cresswell, A New Introduction to Modal Logic, Routledge, 1996.
[44] W. Jamroga, T. Ågotnes, Constructive knowledge: what agents can achieve under imperfect information, J. Appl. Non-Class. Log. 17 (4) (2007) 423–475.
[45] F. Kominis, H. Geffner, Beliefs in multiagent planning: from one agent to many, in: ICAPS, 2015, pp. 147–155.
[46] F. Kominis, H. Geffner, Multiagent online planning with nested beliefs and dialogue, in: Proc. ICAPS, 2017.
[47] B. Kooi, Dynamic term-modal logic, in: J. van Benthem, S. Ju, F. Veltman (Eds.), A Meeting of the Minds, Proceedings of the Workshop on Logic, Rationality and Interaction, Beijing, in: Texts in Computer Science, vol. 8, College Publications, London, 2007, pp. 173–185.
[48] D.L. Kovács, A multi-agent extension of PDDL3.1, in: Proceedings of the 3rd Workshop on the International Planning Competition, 2012, pp. 19–27.
[49] S.A. Kripke, The undecidability of monadic modal quantification theory, Z. Math. Log. Grundl. Math. 8 (1962) 113–116.
[50] T. Le, F. Fabiano, T.C. Son, E. Pontelli, EFP and PG-EFP: epistemic forward search planners in multi-agent domains, in: ICAPS, 2018, pp. 161–170.
[51] S. Lê Cong, S. Pinchinat, F. Schwarzentruber, Small undecidable problems in epistemic planning, in: IJCAI-18, 2018, pp. 4780–4786.
[52] H.S. van Lee, R.K. Rendsvig, S. van Wijk, Intensional protocols for dynamic epistemic logic, J. Philos. Log. 48 (6) (Dec 2019) 1077–1118.
[53] A. Occhipinti Liberman, R.K. Rendsvig, Dynamic term-modal logic for epistemic social network dynamics, in: P. Blackburn, E. Lorini, M. Guo (Eds.), Logic, Rationality, and Interaction, LORI 2019, in: Lecture Notes in Computer Science, vol. 11813, Springer, 2019, pp. 168–182.
[54] A. Occhipinti Liberman, R.K. Rendsvig, Decidability results in first-order epistemic planning, in: IJCAI-20, 2020, in press.
[55] A. Lomuscio, QLB: una logica predicativa delle credenze. Sintassi, semantica e una procedura di prova automatica dei teoremi, Master's thesis, Politecnico di Milano, December 1995.
[56] A. Lomuscio, M. Colombetti, QLB: a quantified logic for belief, in: J. Müller, M.J. Wooldridge, N.R. Jennings (Eds.), Intelligent Agents III. Agent Theories, Architectures, and Languages, in: LNAI, vol. 1193, Springer, 1997, pp. 71–85.
[57] B. Löwe, E. Pacuit, A. Witzel, DEL planning and some tractable cases, in: International Workshop on Logic, Rationality and Interaction, Springer, 2011, pp. 179–192.
[58] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, D. Wilkins, PDDL-the Planning Domain Definition Language, 1998.
[59] L.S. Moss, Dynamic epistemic logic, in: H. van Ditmarsch, J.Y. Halpern, W. van der Hoek, B. Kooi (Eds.), Handbook of Epistemic Logic, College Publications, 2015.
[60] C.J. Muise, V. Belle, P. Felli, S.A. McIlraith, T. Miller, A.R. Pearce, L. Sonenberg, Planning over multi-agent epistemic states: a classical planning approach, in: AAAI, 2015, pp. 3327–3334.
[61] P. Naumov, J. Tao, Everyone knows that someone knows: quantifiers over epistemic agents, Rev. Symb. Log. 12 (2) (2019) 255–270.
[62] E. Orlandelli, G. Corsi, Decidable term-modal logics, in: F. Belardinelli, E. Argente (Eds.), Multi-Agent Systems and Agreement Technologies, Springer International Publishing, Cham, 2018, pp. 147–162.
[63] A. Padmanabha, Propositional Term Modal Logic, PhD thesis, Homi Bhabha National Institute, 2019.
[64] A. Padmanabha, R. Ramanujam, Model checking a logic over systems with regular sets of processes, in: Joint Proceedings of the 3rd Modelling Symposium (ModSym), Developmental Aspects of Intelligent Adaptive Systems (DIAS), and Educational Data Mining Practices in Indian Academia (EDUDM) Co-Located with 10th Innovations in Software Engineering, ISEC 2017, in: CEUR Workshop Proceedings, vol. 1819, CEUR-WS.org, 2017.
[65] A. Padmanabha, R. Ramanujam, Propositional modal logic with implicit modal quantification, in: M.A. Khan, A. Manuel (Eds.), Logic and Its Applications, ICLA 2019, in: Lecture Notes in Computer Science, vol. 11600, Springer, Berlin, Heidelberg, 2019, pp. 6–17.
[66] A. Padmanabha, R. Ramanujam, The monodic fragment of propositional term modal logic, Stud. Log. 107 (3) (2019) 533–557.
[67] A. Padmanabha, R. Ramanujam, Two variable fragment of term modal logic, in: P. Rossmanith, P. Heggernes, J.-P. Katoen (Eds.), 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, Dagstuhl, Germany, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 138, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 30:1–30:14.
[68] A. Padmanabha, R. Ramanujam, Verifying implicitly quantified modal logic over dynamic networks of processes, in: D.V. Hung, M. D'Souza (Eds.), Distributed Computing and Internet Technology, ICDCIT 2020, in: Lecture Notes in Computer Science, vol. 11969, Springer, 2020, pp. 165–176.
[69] E.P. Pednault, Formulating multiagent, dynamic-world problems in the classical planning framework, in: Reasoning About Actions & Plans, Elsevier, 1987, pp. 47–82.
[70] R.P. Petrick, F. Bacchus, Extending the knowledge-based approach to planning with incomplete information and sensing, in: ICAPS, 2004, pp. 2–11.
[71] J.A. Plaza, Logics of public communications, in: M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, Z.W. Ras (Eds.), Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems, 1989, pp. 201–216.
[72] R. Pliuškevičius, A. Pliuškevičienė, Decision procedure for a fragment of mutual belief logic with quantified agent variable, in: F. Toni, P. Torrini (Eds.), CLIMA VI, in: LNAI, vol. 3900, Springer-Verlag, 2006, pp. 112–128.
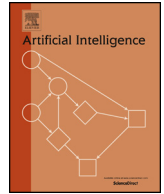
[73] R.K. Rendsvig, Epistemic term-modal logic, in: M. Slavkovik (Ed.), Proceedings of the 15th Student Session of the European Summer School in Logic, Language and Information, 2010, pp. 37–46.

[74] R.K. Rendsvig, Towards a Theory of Semantic Competence, Master's thesis, Dept. of Philosophy and Science Studies and Dept. of Mathematics, Roskilde University, 2011.

[75] R.K. Rendsvig, Modeling semantic competence: a critical review of Frege's puzzle about identity, in: D. Lassiter, M. Slavkovik (Eds.), New Directions in Logic, Language and Computation, in: Lecture Notes in Computer Science, vol. 7415, Springer, 2012, pp. 140–157.

[76] R.K. Rendsvig, Pluralistic ignorance in the bystander effect: informational dynamics of unresponsive witnesses in situations calling for intervention, Synthese 191 (2014) 2471–2498.

[77] R.K. Rendsvig, Model transformers for dynamical systems of dynamic epistemic logic, in: W. van der Hoek, W.H. Holliday, W.F. Wang (Eds.), Logic, Rationality, and Interaction, LORI 2015, Taipei, in: LNCS, Springer, 2015, pp. 316–327.

[78] R.K. Rendsvig, Logical Dynamics and Dynamical Systems, PhD thesis, Lund University, 2018.

[79] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education Limited, 2016.

[80] T. Sawasaki, K. Sano, T. Yamada, Term-sequence-modal logics, in: P. Blackburn, E. Lorini, M. Guo (Eds.), Logic, Rationality, and Interaction, LORI 2019, in: Lecture Notes in Computer Science, vol. 11813, Springer, Berlin, Heidelberg, 2019, pp. 244–258.

[81] I. Sedlar, Term-modal logics of evidence, in: Epistemic Logic for Individual, Social, and Interactive Epistemology, ESSLLI 2014, Tübingen, Germany, 2014.

[82] G. Shtakser, Propositional epistemic logics with quantification over agents of knowledge, Stud. Log. 106 (2) (Apr 2018) 311–344.

[83] R.C. Sleigh, Restricted range in epistemic logic, J. Philos. 69 (1972) 67–77.

[84] L. Thalmann, Term-Modal Logic and Quantifier-Free Dynamic Assignment Logic, PhD thesis, Uppsala University, 2000.

[85] M. Thielscher, GDL-III: a description language for epistemic general game playing, in: IJCAI-17, 2017, pp. 1276–1282.

[86] J. van Benthem, J. van Eijck, M. Gattinger, K. Su, Symbolic model checking for dynamic epistemic logic - S5 and beyond, J. Log. Comput. 28 (2) (2018) 367–402.

[87] G.H. von Wright, An Essay in Modal Logic, North-Holland Publishing Company, 1951.

[88] Y. Wang, J. Seligman, When names are not commonly known: epistemic logic with assignments, in: Advances in Modal Logic 12, Proceedings of the 12th Conference on "Advances in Modal Logic," Held in Bern, Switzerland, August 27-31, 2018, 2018, pp. 611–628.

[89] D.S. Weld, An introduction to least commitment planning, AI Mag. 15 (4) (1994) 27.

[90] H.L. Younes, M.L. Littman, PPDDL1.0: An Extension to PDDL for Expressing Planning Domains with Probabilistic Effects, Techn. Rep. CMU-CS-04-162, 2004.

[91] Q. Yu, X. Wen, Y. Liu, Multi-agent epistemic explanatory diagnosis via reasoning about actions, in: IJCAI-13, 2013, pp. 1183–1190.

Update

Artificial Intelligence

Erratum

Erratum to "Dynamic term-modal logics for first-order epistemic planning" [Artif. Intell. 286 (2020) 103305] ☆

Andrés Occhipinti Liberman [a], Andreas Achen [b], Rasmus Kræmmer Rendsvig [c,*]

[a] *Pompeu Fabra University, Spain*
[b] *The London School of Economics and Political Science, United Kingdom*
[c] *University of Copenhagen, Denmark*

This is an erratum to the paper *Dynamic term-modal logics for first-order epistemic planning*, [1].

Following publication, Prof. Andreas Herzig noticed two issues, one concerning identity and the definition of action models, and another concerning the introduced product update and the associated reduction axiom for atomic formulas. In this erratum, we describe and fix these issues.

1. Preserving the meaning of the identity symbol

Herzig notes that a central definition—that of an *action model*—has an issue that may be exploited to break the semantics by dis-aligning the relation symbol = from its intended use as *identity*.

The issue concerns Def. 9, specifically the definition of the postcondition function post. Postconditions are defined there as follows:

post $: E \to (\texttt{GroundAtoms}(\mathcal{L}) \to \mathcal{L}_{AM})$ is a map that assigns to each event $e \in E$ a postcondition for each ground atom. It is required that $\text{post}(e)(=(t,t)) = \top$ for each event e, to preserve the meaning of equality.

The requirement that $\text{post}(e)(=(t,t)) = \top$ is not sufficiently strong to ensure the meaning of equality/identity. The postcondition function still allows changing the truth value of atomic propositions of the form $t_1 = t_2$, e.g. by changing $t_1 = t_2$ from false to true using an event $e \in E$ with $\text{post}(e)(t_1 = t_2) = \top$.

This was unintentional, and may be exploited: turning $t_1 = t_2$ from false to true may be used to add a pair (d_1, d_2) to the extension of the identity predicate =, despite d_1 and d_2 being distinct elements. If the event e is applied to a world w which satisfies $\neg(t_1 = t_2)$, the extension of = in their update (w, e) will be incorrect: the pair (d_1, d_2) will be in

☆ This article belongs to Special Issue: Epistemic Planning.
* Corresponding author.
E-mail address: rkr@rends.dk (R.K. Rendsvig).

$I'(=, (w, e))$, which should contain only pairs (d, d') with $d = d'$, cf. Def. 5. Hence, $=$ may be made to no-longer satisfy its intended use, and be made to invalidate the axioms of DTML that concern identity.

The issue is fixed by ensuring that the extension of $=$ is unchanged by updates. This may be ensured by changing the last sentence of Def. 9 to:

It is required that $\mathsf{post}(e)((t = t')) = (t = t')$ for all events $e \in E$ and all $t, t' \in \mathsf{T}$, to preserve the meaning of identity.

This change ensures that identity is correctly preserved under updates:

Lemma 1. *For any terms $t, t' \in \mathsf{T}$, any valuation v, any pointed model $(M, w) = (D, W, R, I, w)$ and any pointed action model $(A, e) = (E, Q, \mathsf{pre}, \mathsf{post}, e)$, $M, w \vDash_v t = t'$ iff $M \otimes A, (w, e) \vDash_v t = t'$.*

Proof. Let $M \otimes A = (D', W', R', I')$. Recall from Def. 10 that:

$$I'(r, (w, e)) = (I(r, w) \cup r^+(w)) \setminus r^-(w)$$

When r is the identity symbol, we get:

1. $I(=, w) = \{(d, d) : d \in D\}$ (Def. 5).
2. $=^+(w) = \{(\llbracket t_1 \rrbracket_w^{I,v}, \llbracket t_2 \rrbracket_w^{I,v}) : (M, w) \vDash_v \mathsf{post}(e)((t_1 = t_2))\} = \{(\llbracket t_1 \rrbracket_w^{I,v}, \llbracket t_2 \rrbracket_w^{I,v}) : (M, w) \vDash_v (t_1 = t_2)\}$, since $\mathsf{post}(e)((t = t')) = (t = t')$. Thus, $=^+(w) \subseteq I(=, w)$.
3. $=^-(w) = \{(\llbracket t_1 \rrbracket_w^{I,v}, \llbracket t_2 \rrbracket_w^{I,v}) : (M, w) \nvDash_v \mathsf{post}(e)((t_1 = t_2))\} = \{(\llbracket t_1 \rrbracket_w^{I,v}, \llbracket t_2 \rrbracket_w^{I,v}) : (M, w) \nvDash_v (t_1 = t_2)\}$, again from $\mathsf{post}(e)((t = t')) = (t = t')$. So, $=^-(w) \subseteq \{(d, d') : d, d' \in D, d \neq d'\}$, which means that $=^-(w) \cap I(=, w) = \emptyset$.

From this, letting still r stand for the identity symbol, we get that $I(r, w) \cup r^+(w) = I(r, w)$, which with $r^-(w) \cap I(r, w) = \emptyset$ entails the following:

$$I'(r, (w, e)) = (I(r, w) \cup r^+(w)) \setminus r^-(w)$$
$$= I(r, w) \setminus r^-(w)$$
$$= I(r, w)$$

That is, $I(=, w) = I'(=, (w, e))$. Thus:

$$M, w \vDash_v (t = t') \Leftrightarrow (\llbracket t \rrbracket_w^{I,v}, \llbracket t' \rrbracket_w^{I,v}) \in I(=, w) \quad \text{(satisfaction of formulas, Def. 8)}$$
$$\Leftrightarrow \llbracket t \rrbracket_w^{I,v} = \llbracket t' \rrbracket_w^{I,v} \quad \text{(interpretation of equality symbol, Def. 5)}$$
$$\Leftrightarrow \llbracket t \rrbracket_{(w,e)}^{I',v} = \llbracket t' \rrbracket_{(w,e)}^{I',v} \quad \text{(from the reasoning above, showing that } I(=, w) = I'(=, (w, e)))$$
$$\Leftrightarrow M \otimes A, (w, e) \vDash_v (t = t'). \quad \square$$

2. Reduction axiom for atoms

The second issue noticed by Herzig concerns the reduction axioms used to axiomatize the dynamic logic presented in the paper. The issue concerns the axiom called "Action and atom". The axiom is currently stated as follows:

$$[A, e] r(t_1, ..., t_n) \leftrightarrow (\mathsf{pre}(e) \rightarrow \mathsf{post}(e)(r(t_1, ..., t_n)))$$

The problem with this axiom is that, in the term-modal setting, postconditions for atoms may "interact" through their ground terms, if these refer to the same elements in the domain. E.g., the postcondition for one atom $r(t_1, ..., t_n)$ can affect the truth value of another atom $r(t_1', ..., t_n')$ after the update, if t_i and t_i' refer to the same elements, for each $i = 1, ..., n$.[1] However, interaction between atoms is not accounted for by the original axiom.

Let us illustrate how postconditions can interact. Suppose we have a model with interpretation function I and let w be a world in this model. Consider two atoms $r(t_1, ..., t_n)$ and $r(t_1', ..., t_n')$ such that $M, w \vDash_v \bigwedge_{i=1}^n (t_i = t_i')$. Note that this implies that $M, w \vDash_v r(t_1, ..., t_n) \leftrightarrow r(t_1', ..., t_n')$. Suppose that these two atoms are false in w. Consider an action model A with a single event e, with precondition $\mathsf{pre}(e) = \top$, the postcondition $\mathsf{post}(e)(r(t_1, ..., t_n)) = \top$, and all other postconditions are given as $\mathsf{post}(e)(r'(t_1', ..., t_n')) = r'(t_1', ..., t_n')$. As $\mathsf{post}(e)(r(t_1, ..., t_n)) = \top$, $(\llbracket t_1 \rrbracket_w^{I,v}, ..., \llbracket t_n \rrbracket_w^{I,v}) \in r^+(w)$.

---

[1] Interestingly, this interdependence of atoms is specific to dynamics with first-order logical atoms; in a propositional modal setting, a postcondition for propositional atom p can only change the truth value of p, it has no way of altering the truth value of e.g. q.

Moreover, given (i) $post(e)(r'(t'_1, \ldots, t'_n)) = r'(t'_1, \ldots, t'_n)$, (ii) $M, w \vDash_v \neg r'(t'_1, \ldots, t'_n)$ and (iii) $M, w \vDash_v \bigwedge_{i=1}^n (t_i = t'_i)$, we have $(\llbracket t_1 \rrbracket_w^{I,v}, \ldots, \llbracket t_n \rrbracket_w^{I,v}) \in r^-(w)$. Recall the definition of product update:

$$I'(r, (w, e)) = (I(r, w) \cup r^+(w)) \setminus r^-(w)$$

I.e., whenever a tuple $(\llbracket t_1 \rrbracket_w^{I,v}, \ldots, \llbracket t_n \rrbracket_w^{I,v}) \in (r^+(w) \cap r^-(w))$, "priority" is given to r^-, since the set difference operation is applied last. This means that although the action model has the postcondition $post(e)(r(t_1, \ldots, t_n)) = \top$, the atom $r(t_1, \ldots, t_n)$ will not be true after the update, since the postcondition requiring it to be false, $post(e)(r'(t'_1, \ldots, t'_n)) = r'(t'_1, \ldots, t'_n)$, takes priority over it. Hence the reduction atom

$$[A, e]r(t_1, \ldots, t_n) \leftrightarrow (pre(e) \rightarrow post(e)(r(t_1, \ldots, t_n)))$$

is not true in (M, w), as in this case the axiom boils down to $M, w \vDash_v [A, e]r(t_1, \ldots, t_n) \leftrightarrow \top$.

To address this issue and produce a correct axiomatization, we suggest the following two revisions:

1. Require that each event of an action model only changes a finite number of ground atoms, ensured by adding as last sentence to Def. 9:

    It is required that the set $G(e) := \{\varphi \in \texttt{GroundAtoms}(\mathcal{L}) : post(e)(\varphi) \neq \varphi\}$ is finite, for each event e.

2. Revise the *Action and atom* axiom from Table 3 to

$$[A, e]r(t_1, \ldots, t_n) \leftrightarrow \left(pre(e) \rightarrow \bigwedge_{(t'_1, \ldots, t'_n) \in \texttt{GT}(e)^n} \left(\left(\bigwedge_{i=1}^n (t'_i = t_i) \right) \rightarrow post(e)(r(t'_1, \ldots, t'_n)) \right) \right) \tag{1}$$

    where $\texttt{GT}(e)$ is the set of ground terms that occur in any formula of $G(e)$.

Neither of these revisions affect other results in the paper. The revision to postcondition functions ensures that they may be finitely represented. Thus the revision *(voids)* the paragraph immediately following **Notation 2**, which in the published version states that we do not make such a restriction.[2]

The revised version of the axiom takes the possible interaction of atoms into account by ensuring that the postconditions for $r(t_1, \ldots, t_n)$ and $r(t'_1, \ldots, t'_n)$ are dependent when $t_i = t'_i$ for each $i = 1, \ldots, n$.

Lemma 2. *For any language $\mathcal{L}$ based on a signature $\Sigma = (\texttt{V}, \texttt{C}, \texttt{R}, \texttt{F}, \texttt{t})$, for any action model (A, e) based on the extended language $\mathcal{L}_{AM}$, the formula (1) is valid over the class of all frames **F**.*

Proof. Let (M, w), (A, e) and v be arbitrary. There are two cases: where (A, e) is not applicable to (M, w) and where it is.

Assume (A, e) is not applicable to (M, w), i.e. that $(M, w) \nvDash_v pre(e)$. By Definition 16, for any $\varphi \in \mathcal{L}_{AM}$, $(M, w) \vDash_v [A, e]\varphi$ iff $(M, w) \vDash_v pre(e)$ implies $(M \otimes A, (w, e)) \vDash_v \varphi$. The inapplicability of (A, e) to (M, w) implies that (M, w) satisfies both sides of (1) under any valuation v.

Assume (A, e) is applicable to (M, w), i.e. that $(M, w) \vDash_v pre(e)$.

For the left-to-right implication of the reduction axiom, assume that $(M, w) \vDash_v [A, e]r(t_1, \ldots, t_n)$. The following holds:

$$(M, w) \vDash_v [A, e]r(t_1, \ldots, t_n) \qquad \text{(assumption)}$$

$$\Leftrightarrow (M \otimes A, (w, e)) \vDash_v r(t_1, \ldots, t_n) \qquad \text{(satisfaction of dynamic formulas, Def. 16)}$$

$$\Leftrightarrow (\llbracket t_1 \rrbracket_{(w,e)}^{I',v}, \ldots, \llbracket t_n \rrbracket_{(w,e)}^{I',v}) \in I'(r, (w, e)) \qquad \text{(satisfaction of formulas, Def. 8)}$$

$$\Leftrightarrow (\llbracket t_1 \rrbracket_{(w,e)}^{I',v}, \ldots, \llbracket t_n \rrbracket_{(w,e)}^{I',v}) \in (I(r, w) \cup r^+(w)) \setminus r^-(w) \qquad \text{(product update, Def. 10)}$$

$$\Leftrightarrow (\llbracket t_1 \rrbracket_w^{I,v}, \ldots, \llbracket t_n \rrbracket_w^{I,v}) \in (I(r, w) \cup r^+(w)) \setminus r^-(w) \qquad \text{(as terms never change extension}$$

$$\text{under product updates, Def. 10)}$$

The latter implies that $(M, w) \vDash_v post(e)(r(t_1, \ldots, t_n))$: If $(\llbracket t_1 \rrbracket_w^{I,v}, \ldots, \llbracket t_n \rrbracket_w^{I,v}) \in I(r, w) \setminus r^-(w)$, then, by definition of $r^-(w)$, it is not the case that $(M, w) \nvDash_v post(e)(r(t_1, \ldots, t_n))$, so $(M, w) \vDash_v post(e)(r(t_1, \ldots, t_n))$. Else, $(\llbracket t_1 \rrbracket_w^{I,v}, \ldots, \llbracket t_n \rrbracket_w^{I,v}) \in$

---

[2] Our first instinct was to restrict the language to signatures that give rise to a *finite* set of ground terms $\texttt{G}$ and conjugate over $\texttt{G}^n$ in the revised axiom. To ensure a finite set of ground terms, one must use only a finite set of constants $\texttt{C}$, in combination with e.g. an empty set of function symbols $\texttt{F}$, or a finite set of function symbols and the restriction that a term may only feature each function symbol f applied *finitely* many times (up to some fixed number k). We thank Herzig for suggesting the weaker and more finely targeted restriction above.

$r^+(w) \backslash r^-(w)$, so, by definition of $r^+(w)$, $(M, w) \vDash_v post(e)(r(t_1, ..., t_n))$. As $(M, w) \vDash_v post(e)(r(t_1, ..., t_n))$, it follows that, for any $t'_1, ..., t'_n$ for which $[\![t'_i]\!]_w^{I,v} = [\![t_i]\!]_w^{I,v}$ for $i = 1, ..., n$, $(M, w) \vDash_v post(e)(r(t'_1, ..., t'_n))$. In particular, this is the case for all the finitely many grounded terms $\text{GT}(e)$. Hence the conjunction

$$(M, w) \vDash_v \bigwedge_{(t'_1, ..., t'_n) \in \text{GT}(e)^n} \left(\bigwedge_{i=1}^n (t'_i = t_i) \rightarrow post(e)(r(t'_1, ..., t'_n)) \right).$$

For the right-to-left implication, assume

$$(M, w) \vDash_v \bigwedge_{(t'_1, ..., t'_n) \in \text{GT}(e)^n} \left(\bigwedge_{i=1}^n (t'_i = t_i) \rightarrow post(e)(r(t'_1, ..., t'_n)) \right).$$

This implies specifically that

$$(M, w) \vDash_v \bigwedge_{i=1}^n (t_i = t_i) \rightarrow post(e)(r(t_1, ..., t_n))$$

so $([\![t_1]\!]_{(w,e)}^{I,v}, ..., [\![t_n]\!]_{(w,e)}^{I,v}) \in r^+(w)$. Hence $([\![t_1]\!]_w^{I',v}, ..., [\![t_n]\!]_w^{I',v}) \in I'(r, w)$, as $[\![t_i]\!]_{(w,e)}^{I',v} = [\![t_i]\!]_w^{I,v}$ for each $i = 1, ..., n$. Therefore, $(M \otimes A, (w, e)) \vDash_v r(t_1, ..., t_n)$, and so $(M, w) \vDash_v [A, e]r(t_1, ..., t_n)$. $\square$

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A. Occhipinti Liberman, A. Achen, R.K. Rendsvig, Dynamic term-modal logics for first-order epistemic planning, Artif. Intell. 286 (2020) 103305.