Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

A Dynamic Epistemic Logic with Finite Iteration and Parallel Composition

Andreas Herzig[1] , **Frédéric Maris**[2] , **Elise Perrotin**[2]
[1]IRIT, CNRS, Toulouse
[2]IRIT, Univ. Toulouse III
{Andreas.Herzig, Frederic.Maris, Elise.Perrotin}@irit.fr

Abstract

Existing dynamic epistemic logics combine standard epistemic logic with a restricted version of dynamic logic. Instead, we here combine a restricted epistemic logic with a rich version of dynamic logic. The epistemic logic is based on 'knowing-whether' operators and basically disallows disjunctions and conjunctions in their scope; it moreover captures 'knowing-what'. The dynamic logic has not only all the standard program operators of Propositional Dynamic Logic, but also parallel composition as well as an operator of inclusive nondeterministic composition; its atomic programs are assignments of propositional variables. We show that the resulting dynamic epistemic logic is powerful enough to capture several kinds of sequential and parallel planning, both in the unbounded and in the finite horizon versions.

1 Introduction

Dynamic epistemic logics (DELs) combine epistemic logic and dynamic logic. Most approaches in the literature combine full-fledged epistemic logic with restricted versions of dynamic logic. The latter typically lack the program operators of propositional dynamic logic (PDL), in particular the operator of finite iteration ('Kleene star'). The reason is that the latter causes undecidability of satisfiability (Miller and Moss 2005) and even of model checking. The latter is directly related to the general undecidability of DEL-based planning (Bolander and Andersen 2011; Aucher and Bolander 2013; Bolander et al. 2020). In previous work we have proposed a different route whose starting point is a restricted version of standard epistemic logic: boolean combinations of 'knowing-whether' operators followed by a propositional variable (Cooper et al. 2016; Cooper et al. 2020a). In that lightweight epistemic logic, EL-O, the existence of sequential and parallel epistemic plans can be decided in PSPACE (Cooper et al. 2020b). A dynamic extension, DEL-PAO, was proposed in (Herzig, Lorini, and Maffre 2015), and a modelling of planning tasks with no common knowledge and no parallel plans was given in (Cooper et al. 2016). We here give a fuller and more succinct dynamic logic, adding in particular an operator of parallel composition as well as an operator of inclusive nondeterministic composition. Both are imported from Dynamic Logic of Parallel Propositional Assignments DL-PPA (Herzig, Maris, and Vianey 2019), which is conceptually and mathematically simpler than several other proposals for adding parallel composition to dynamic logic such as (Benevides and Schechter 2014; Balbiani and Boudou 2018; Boudou, Herzig, and Troquard 2021). In the resulting DEL the solvability of various planning tasks can be captured: the existence of sequential and parallel plans, both in the unbounded and in the finite horizon versions. The operator of inclusive nondeterministic composition turns out to be instrumental for the succinct modelling of parallel planning.

We generalize the epistemic 'knowing-whether' operator to a 'knowing-what' (or 'knowing the value') operator, introduced and argued for in (Plaza 2007; Wang and Fan 2013; Wang 2018) as a useful and natural addition to logics of knowledge, particularly in the field of AI: agents may not only need to know whether or not a given proposition is true, but also what another agent's telephone number is or the code to open a door. We show that this notion of 'knowing what' can be added to EL-O without much difficulty, leading to an enlarged field of possibilities regarding the planning tasks that can be modelled and worked with. We illustrate our logic by means of the gossip problem where each agent $i \in Agt$ knows a secret s_i that none of the other agents knows and where the goal is to find a sequence of phone calls after which every agent knows every secret (Landau 1954; Knödel 1975; van Ditmarsch, van der Hoek, and Kuijer 2020). Several interesting variants of that problem can be designed, in particular where the goal is to obtain higher-order knowledge (Cooper et al. 2019; Cooper et al. 2020a; van Ditmarsch, Gattinger, and Ramezanian 2020).

We present the static epistemic logic EL-OC in Section 2 and extend it to the dynamic logic DEL-PPAOC in Section 3. In Section 4 we apply DEL-PPAOC to planning. Proofs of the results can be found in (Perrotin forthcoming, Ch. 6).

2 The Static Logic: EL-OC

We introduce the Epistemic Logic of Observation with Constants, abbreviated to EL-OC.

2.1 Atoms, Introspective Atoms, and Atomic Consequence

Let *Prop* be a countable set of propositional variables, let *Cst* be a countable set of constants, and let *Agt* be a finite set of agents. The set of *observability operators* is

$$OBS = \{S_i \,:\, i \in Agt\} \cup \{JS\},$$

where S_i stands for individual observability of agent i and JS stands for joint observability of all agents. The set of all sequences of observability operators is noted OBS^* and the set of all non-empty sequences is noted OBS^+. We use σ, σ', etc. to denote elements of OBS^*.

Observability atoms, or atoms for short, are finite sequences of observability operators followed by a propositional variable. The set of all atoms is

$$ATM = \{\sigma\, p \,:\, \sigma \in OBS^*, p \in Prop\} \cup \{\sigma\, c \,:\, \sigma \in OBS^+, c \in Cst\}.$$

We use $\alpha, \alpha', \beta, \ldots$ to denote atoms, unless specified as members of $ATM \cup Cst$. Here are some examples: $S_1\, p$ reads "1 sees the truth value of p", i.e., 1 knows whether p is true or false. $S_1\, c$ reads "1 sees the value of c". $JS\, S_2\, c$ reads "all agents jointly see whether agent 2 sees the value of c", i.e., there is joint attention in the group of all agents concerning 2's observation of c: agent 2 may or may not see the value of c, and in both cases this is jointly observed. An atom with an empty sequence of observability operators is nothing but a propositional variable; constants are always preceded by at least one observability operator as they have no truth value.

We follow the principles of EL-O and call an atom *introspective* if it contains two consecutive S_i, or a JS that is preceded by a non-empty sequence of observability operators. The set of all introspective atoms is

$$I\text{-}ATM = \{\sigma\, S_i\, S_i\, \alpha \,:\, \sigma \in OBS^* \text{ and } \alpha \in ATM \cup Cst\} \cup \{\sigma\, JS\, \alpha \,:\, \sigma \in OBS^+ \text{ and } \alpha \in ATM \cup Cst\}.$$

We finally define a relation of *atomic consequence* between observability atoms as follows:

$$\alpha \Rightarrow \beta \;\text{ iff }\; \alpha = \beta \text{ or there are } \alpha' \in ATM \cup Cst, \sigma \in OBS^+ \text{ such that } \alpha = JS\,\alpha' \text{ and } \beta = \sigma\,\alpha'.$$

When $\alpha \Rightarrow \beta$ we say that α is a *cause* of β and that β is a *consequence* of α. We denote by $\alpha^{\Leftarrow}$ the set of causes of α, and by $\alpha^{\Rightarrow}$ the set of its consequences. We generalize this to sets of atoms $V \subseteq ATM$: $V^{\Rightarrow} = \bigcup_{\alpha \in V} \alpha^{\Rightarrow}$.

2.2 Language and Semantics

The language of EL-OC is defined by the grammar

$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \wedge \varphi)$$

where α ranges over ATM. The boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined in the standard way.

A *state* is a subset of the set of atoms ATM. We denote states by V, U, W, etc. The set of all states is 2^{ATM}.

As in EL-O semantics, we interpret formulas such that introspection is simulated. The only non-standard case is:

$$\|\alpha\| = \{V \,:\, \alpha \in V^{\Rightarrow} \cup I\text{-}ATM\}.$$

Hence α is true in state V if and only if α is introspective or $\beta \Rightarrow \alpha$ for some $\beta \in V$.

$S_i\, S_i\, \alpha$	(Vis_1)	$JS\,\alpha \rightarrow S_i\,\alpha$	(Vis_4)
$JS\, JS\, \alpha$	(Vis_2)	$JS\,\alpha \rightarrow JS\, S_i\,\alpha$	(Vis_5)
$JS\, S_i\, S_i\, \alpha$	(Vis_3)		

Table 1: Axioms for introspection, where $\alpha \in ATM \cup Cst$

Example 1. In the initial state V_0^G of the gossip problem every agent only knows her own secret. Therefore $V_0^G = \{S_i\, s_i \,:\, i \in Agt\}$ where all s_i are constants. (This differs from modellings of the gossip problem without 'knowing-what' where secrets are assumed to be either true or false.) Then $V_0^G \in \|S_i\, s_i \wedge \bigwedge_{j \neq i} \neg S_i\, s_j\|$ for every $i \in Agt$.

A formula φ is EL-OC *valid* if $V \in \|\varphi\|$ for every $V \subseteq ATM$; it is *satisfiable* if $V \in \|\varphi\|$ for some $V \subseteq ATM$. Clearly, atom α is valid if and only if it is introspective. It is easily shown that all properties of EL-O given in (Cooper et al. 2020a), in particular NP-completeness of the satisfiability problem, still hold in EL-OC. The EL-OC validities are axiomatised in Table 1; the completeness proof closely follows that of (Cooper et al. 2020a).

3 Adding Dynamic Logic

We now describe the full dynamic logic DEL-PPAOC.

3.1 Language of DEL-PPAOC

The language of DEL-PPAOC extends that of EL-OC with the dynamic operator $\langle\pi\rangle$, where π is a program. Programs π and formulas φ are defined by the grammar

$$\begin{array}{lcl} \varphi & ::= & \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\pi\rangle\varphi, \\ \pi & ::= & \alpha{\leftarrow}\varphi \mid \varphi? \mid \pi;\pi \mid \pi \cup \pi \mid \pi \sqcup \pi \mid \pi \sqcap \pi \mid \pi^*, \end{array}$$

where α ranges over the set of atomic formulas ATM. The formula $\langle\pi\rangle\varphi$ reads "there is a possible execution of π such that φ is true afterwards". The program $\alpha{\leftarrow}\varphi$ assigns the truth value of φ to α. $\varphi?$ tests whether φ is true (and fails when φ is false). $\pi_1;\pi_2$ executes π_1 and π_2 in sequence. $\pi_1 \cup \pi_2$ nondeterministically chooses between executing either π_1 or π_2; and $\pi_1 \sqcup \pi_2$ nondeterministically chooses between executing either π_1, or π_2, or both. $\pi_1 \sqcap \pi_2$ is the parallel composition of π_1 or π_2. The set of all formulas is $Fml_{\text{DEL-PPAOC}}$.

The formula $[\pi]\varphi$ abbreviates $\neg\langle\pi\rangle\neg\varphi$ and therefore has to be read "φ is true after every possible execution of π". We define n-times iteration of π by induction on n: $\pi^0 = \top?$ and $\pi^{n+1} = \pi;\pi^n$. Finally, we define $\pi^{\leq n}$ as $\bigcup_{0 \leq i \leq n} \pi^i$. π^* is the unbounded iteration of π.

3.2 Semantics of DEL-PPAOC

The interpretation of a formula φ is a set of valuations $\|\varphi\| \subseteq 2^{ATM}$, just as in EL-OC. The interpretation of a program π is a ternary relation on the set of valuations: $\|\pi\| \subseteq 2^{ATM} \times 2^{ATM} \times 2^{ATM}$. When $\langle V, U, W\rangle \in \|\pi\|$ then there is an execution of π from state V to state U assigning the variables at W. The interpretation function is defined by mutual recursion. The main clauses are given in Table 2; the others are standard.

The interpretation of the assignment $\alpha{\leftarrow}\varphi$ is that, either (1) φ is true, α gets the value true and the set of assigned

$$
\begin{aligned}
\|\alpha\| &= \{V \ : \ \alpha \in V^{\Rightarrow} \cup \mathit{I\text{-}ATM}\},\\
\|\langle\pi\rangle\varphi\| &= \{V \ : \ \text{there are } U, W \text{ such that } \langle V, U, W\rangle \in \|\pi\| \text{ and } U \in \|\varphi\|\},\\
\|\alpha{\leftarrow}\varphi\| &= \{\langle V, V{\cup}\{\alpha\}, \{\alpha\}\rangle \ : \ V \in \|\varphi\|\} \cup \{\langle V, V\backslash\alpha^{\Leftarrow}, \alpha^{\Leftarrow}\rangle \ : \ V \notin \|\varphi\|\},\\
\|\varphi?\| &= \{\langle V, V, \emptyset\rangle \ : \ V \in \|\varphi\|\},\\
\|\pi_1;\pi_2\| &= \{\langle V, U, W\rangle \ : \ \text{there are } U_1, W_1, W_2 \text{ such that } \langle V, U_1, W_1\rangle \in \|\pi_1\|, \langle U_1, U, W_2\rangle \in \|\pi_2\|, \text{ and } W = W_1 \cup W_2\},\\
\|\pi_1 \cup \pi_2\| &= \|\pi_1\| \cup \|\pi_2\|,\\
\|\pi_1 \sqcup \pi_2\| &= \|\pi_1\| \cup \|\pi_2\| \cup \|\pi_1 \sqcap \pi_2\|,\\
\|\pi_1 \sqcap \pi_2\| &= \left\{\langle V, U, W\rangle \ : \ \begin{array}{l}\text{there are } U_1, W_1, U_2, W_2 \text{ such that } \langle V, U_1, W_1\rangle \in \|\pi_1\|, \langle V, U_2, W_2\rangle \in \|\pi_2\|,\\ W_1 \cap W_2 \cap U_1 = W_1 \cap W_2 \cap U_2, U = (V\backslash W) \cup (U_1{\cap}W_1) \cup (U_2{\cap}W_2), \text{ and } W = W_1 \cup W_2\end{array}\right\},\\
\|\pi^*\| &= \bigcup_{k\in\mathbb{N}_0} \|\pi^k\|.
\end{aligned}
$$

Table 2: Interpretation of DEL-PPAOC dynamic operator and programs

variables is the singleton $\{\alpha\}$, or (2) φ is false, all causes of α get the value false and the set of assigned variables is the set $\alpha^{\Leftarrow}$ of all causes of α.

The interpretation of parallel composition $\pi_1 \sqcap \pi_2$ is that each subprogram π_k is executed locally; then it is checked that the modifications (in terms of assigned variables) are compatible, in the sense that variables that are assigned by both subprograms (i.e., those at $W_1 \cap W_2$) get the same truth value. If this is not the case then the parallel composition fails; otherwise the resulting valuation U is computed by putting together (1) the unchanged part of V (i.e., $V \setminus W$), (2) the updates of π_1 (i.e., $U_1 \cap W_1$), (3) the updates of π_2 (i.e., $U_2 \cap W_2$). Moreover, the set of variables W assigned by a parallel composition is the union of the sets of variables assigned by the subprograms.

The interpretation of inclusive nondeterministic composition $\pi_1 \sqcup \pi_2$ is the exclusive nondeterministic composition of the three programs π_1, π_2 and $\pi_1 \sqcap \pi_2$.

4 Epistemic Planning

In this section we show how DEL-PPAO captures simple epistemic planning tasks where actions change not only the world, but also the agents' knowledge. We assume deterministic actions with conditional effects described by add- and delete-lists. Such conditional effects are crucial: when an agent performs an action then the effects on another agent's epistemic state typically depend on whether that agent sees the variables that are modified by the action (Andersen, Bolander, and Jensen 2012; Cooper et al. 2020a).

4.1 Action Descriptions

An *action description* is a pair $\mathsf{a} = \langle pre(\mathsf{a}), \mathit{eff}(\mathsf{a})\rangle$ where $pre(\mathsf{a}) \in Fml_{\text{DEL-PPAOC}}$ (the *precondition* of a) and

$$\mathit{eff}(\mathsf{a}) \subseteq Fml_{\text{DEL-PPAOC}} \times 2^{ATM} \times 2^{ATM}$$

are the *conditional effects* of a. For a triple

$$ce = \langle cnd(ce), \mathit{ceff}^+(ce), \mathit{ceff}^-(ce)\rangle$$

in $\mathit{eff}(\mathsf{a})$, $cnd(ce)$ is the condition of ce, $\mathit{ceff}^+(ce)$ are the added atoms, and $\mathit{ceff}^-(ce)$ are the deleted atoms. We require consistency of effects: if $ce_1, ce_2 \in \mathit{eff}(\mathsf{a})$ and $\|pre(\mathsf{a}) \wedge cnd(ce_1) \wedge cnd(ce_2)\| \neq \emptyset$ then $\mathit{ceff}^+(ce_1) \cap (\mathit{ceff}^-(ce_2))^{\Leftarrow} = \emptyset$.

Example 2. In the gossip problem a call has two conditional effects per secret: if i knows s_k then s_k becomes known to j, and vice versa. Hence $pre(\mathsf{call}^i_j) = \top$ and

$$\mathit{eff}(\mathsf{call}^i_j) = \{\langle S_i\, s_k, \{S_j\, s_k\}, \emptyset\rangle \ : \ 1 \leq k \leq n\} \cup \{\langle S_j\, s_k, \{S_i\, s_1\}, \emptyset\rangle \ : \ 1 \leq k \leq n\}.$$

To every action a we associate a partial function τ_{a} on states as follows: $\tau_{\mathsf{a}}(V)$ is defined if $V \in \|pre(\mathsf{a})\|$; and when $\tau_{\mathsf{a}}(V)$ is defined then for every conditional effect $ce \in \mathit{eff}(\mathsf{a})$, if its triggering condition $cnd(ce)$ is satisfied ('ce fires') then the negative effects of ce are removed, as well as their causes, and the positive effects of ce are added:

$$\tau_{\mathsf{a}}(V) = \Bigg(V \setminus \Big(\bigcup_{\substack{ce\in\mathit{eff}(\mathsf{a}),\\ V\in\|cnd(ce)\|}} (\mathit{ceff}^-(ce))^{\Leftarrow}\Big)\Bigg) \cup \Big(\bigcup_{\substack{ce\in\mathit{eff}(\mathsf{a}),\\ V\in\|cnd(ce)\|}} (\mathit{ceff}^+(ce))\Big).$$

We capture τ_{a} by a program testing the precondition and processing in parallel the conditional effects:

$$\mathsf{exeAct}(\mathsf{a}) = pre(\mathsf{a})?; \bigsqcap_{ce\in\mathit{eff}(\mathsf{a})} \left(\begin{array}{l}\neg cnd(ce)? \cup \\ \left(\begin{array}{l}cnd(ce)? \sqcap \\ (\bigsqcap_{\alpha\in\mathit{ceff}^+(ce)} \alpha{\leftarrow}\top) \sqcap \\ (\bigsqcap_{\alpha\in\mathit{ceff}^-(ce)} \alpha{\leftarrow}\bot)\end{array}\right)\end{array}\right).$$

Note that thanks to the constraint of consistency of action effects the big parallel composition is always executable.

4.2 Consistency of a Set of Actions at a State

We give two consistency constraints for parallel execution.

First, a_1 and a_2 *have no contradictory effects at* V if for every $ce_1 \in \mathit{eff}(\mathsf{a}_1)$ and $ce_2 \in \mathit{eff}(\mathsf{a}_2)$, if $V \in \|cnd(ce_1) \wedge cnd(ce_2)\|$ then $\mathit{ceff}^+(ce_1) \cap \mathit{ceff}^-(ce_2)^{\Leftarrow} = \emptyset$. Hence no effect of one action is destroyed by the effect of another action executed in parallel.[1] For example, two actions of moving the same object to two different spots have contradictory effects.

Second, two different actions a_1 and a_2 *have no cross-interaction at* V if

1. V and $\tau_{\mathsf{a}_1}(V)$ agree on $pre(\mathsf{a}_2)$ and on the condition $cnd(ce_2)$ of every conditional effect $ce_2 \in \mathit{eff}(\mathsf{a}_2)$,

[1]The case $\mathsf{a}_1 = \mathsf{a}_2$ is already excluded by our requirement on action descriptions in Section 4.1.

2. V and $\tau_{a_2}(V)$ agree on $pre(a_1)$ and on the condition $cnd(ce_1)$ of every conditional effect $ce_1 \in eff(a_1)$,

where "V and V' agree on φ" means that either both $V \in ||\varphi||$ and $V' \in ||\varphi||$, or both $V \notin ||\varphi||$ and $V' \notin ||\varphi||$. Hence neither action preconditions nor effect conditions are modified by the effect of another action executed in parallel, guaranteeing that they can be interleaved. For example, the actions of two different agents picking up the same object cross-interact.

A set of actions A is *consistent at* V if for every $\mathsf{a}, \mathsf{a}' \in \mathsf{A}$ such that $\mathsf{a} \neq \mathsf{a}'$, (1) a and a' have no contradictory effects at V; (2) a and a' have no cross interaction at V.

Example 3. A gossipping conference call $\{\mathsf{call}^i_j, \mathsf{call}^i_k\}$ is consistent. A way to exclude this is to replace call^i_j by Tcall^i_j, with $pre(\mathsf{Tcall}^i_j) = \top$ and

$$eff(\mathsf{Tcall}^i_j) = eff(\mathsf{call}^i_j) \cup \{\langle tg_i, \emptyset, \{tg_i\}\rangle\} \cup \{\langle \neg tg_i, \{tg_i\}, \emptyset\rangle\} \cup \{\langle tg_j, \emptyset, \{tg_j\}\rangle\} \cup \{\langle \neg tg_j, \{tg_j\}, \emptyset\rangle\}.$$

Then two different calls involving i each toggle the value of tg_i, ensuring that they cross-interact in any state.

4.3 Semantics of a Set of Actions

A set of actions A determines a partial function τ_{A} on valuations. It is defined at V if $V \in ||\bigwedge_{\mathsf{a}\in\mathsf{A}} pre(\mathsf{a})||$ and A is consistent at V. When it is defined at V then:

$$\tau_{\mathsf{A}}(V) = \Bigg(V \setminus \Big(\bigcup_{\substack{\mathsf{a}\in\mathsf{A}, ce\in eff(\mathsf{a}),\\ V\in||cnd(ce)||}} (ceff^-(ce))^{\Leftarrow}\Big)\Bigg) \cup \Bigg(\bigcup_{\substack{\mathsf{a}\in\mathsf{A}, ce\in eff(\mathsf{a}),\\ V\in||cnd(ce)||}} (ceff^+(ce))\Bigg).$$

Thanks to consistency of A the order in which negative and positive effects are processed does not matter.

Let us capture this in DEL-PPAO. First, the formula

$$\mathsf{Insensitive}(\mathsf{a}, \mathsf{a}') = \langle \mathsf{exeAct}(\mathsf{a}')\rangle pre(\mathsf{a}) \wedge \bigwedge_{ce\in eff(\mathsf{a})} (cnd(ce) \leftrightarrow \langle \mathsf{exeAct}(\mathsf{a}')\rangle cnd(ce))$$

expresses that neither executability nor effects of a are sensitive to the execution of a'. Then to every A we associate the DEL-PPAOC program

$$\mathsf{exeAct}(\mathsf{A}) = \bigwedge_{\mathsf{a},\mathsf{a}'\in\mathsf{A},\mathsf{a}\neq\mathsf{a}'} \mathsf{Insensitive}(\mathsf{a}, \mathsf{a}')?; \bigsqcap_{\mathsf{a}\in\mathsf{A}} \mathsf{exeAct}(\mathsf{a}).$$

Proposition 1. *For every finite set of actions* A*:*

1. τ_{A} *is defined at* V *if and only if* $\langle V, U, W\rangle \in ||\mathsf{exeAct}(\mathsf{A})||$ *for some* U, W.
2. *If* τ_{A} *is defined at* V *then* $\tau_{\mathsf{A}}(V) = U$ *iff* $\langle V, U, W\rangle \in ||\mathsf{exeAct}(\mathsf{A})||$ *for some* W.

4.4 Simple Epistemic Planning Tasks

A *simple epistemic planning task* is a triple $\langle Act, V_0, Goal\rangle$ where Act is a finite set of actions, $V_0 \in 2^{ATM}$ is a finite state (the initial state), and $Goal \in Fml_{\mathsf{DEL\text{-}PPAOC}}$ is a DEL-PPAOC formula.

Example 4. The gossip problem can be viewed as the planning task $G = \langle Act^G, V_0^G, Goal^G\rangle$ with

$$Act^G = \{\mathsf{Tcall}^i_j : i, j \in Agt \text{ and } i \neq j\} \text{ (cf. Example 3)},$$
$$V_0^G = \{S_i\, s_i : i \in Agt\} \text{ (cf. Example 1)},$$
$$Goal^G = \bigwedge_{i\in Agt} \bigwedge_{j\in Agt} S_j\, s_i.$$

State V is *reachable by a parallel plan* from $V_0 \in 2^{ATM}$ via a set of actions Act if there is an $m \geq 0$ and sequences $\langle V_0, \dots, V_m\rangle$ and $\langle \mathsf{A}_1, \dots, \mathsf{A}_m\rangle$ such that $V_m = V$ and for $1 \leq k \leq m$, $V_k \in 2^{ATM}$, $\mathsf{A}_k \subseteq Act$, and $\tau_{\mathsf{A}_k}(V_{k-1}) = V_k$.

A simple epistemic planning task $\langle Act, V_0, Goal\rangle$ is *solvable by a parallel plan* if there is a state V that is reachable by a parallel plan from V_0 via Act such that $V \in ||Goal||$; otherwise it is unsolvable by a parallel plan.

Theorem 1. *A planning task* $\langle Act, V_0, Goal\rangle$ *is solvable by a parallel plan with no more than k steps if and only if:*

$$V_0 \in \Big|\Big|\Big\langle\Big(\bigsqcap_{\mathsf{a}\in Act} x_{\mathsf{a}}{\leftarrow}\bot; \bigsqcup_{\mathsf{a}\in Act} x_{\mathsf{a}}{\leftarrow}\top; \pi_x\Big)^{\leq k}\Big\rangle Goal\Big|\Big|,$$

where the x_{a} *are fresh variables and where*

$$\pi_x = \bigwedge_{\mathsf{a},\mathsf{a}'\in Act,\mathsf{a}'\neq\mathsf{a}} ((x_{\mathsf{a}} \wedge x_{\mathsf{a}'}) \to \mathsf{Insensitive}(\mathsf{a}, \mathsf{a}'))?\,; \bigsqcap_{\mathsf{a}\in Act} (\neg x_{\mathsf{a}}? \cup (x_{\mathsf{a}}?; \mathsf{exeAct}(\mathsf{a}))).$$

Example 5. Task G can be solved in $\lceil \log_2 n\rceil$ steps of parallel calls if the number of agents n is even, and in $\lceil \log_2 n\rceil + 1$ steps if n is odd (Bavelas 1950; Landau 1954; Knödel 1975; Cooper et al. 2019). For instance, for $n = 4$ the parallel plan $\langle\{\mathsf{Tcall}^1_2, \mathsf{Tcall}^3_4\}, \{\mathsf{Tcall}^1_3, \mathsf{Tcall}^2_4\}\rangle$ solves G in 2 steps.

Solvability by a sequential plan is the special case where the parallel plan is a sequence of singletons.

Theorem 2. *A planning task* $\langle Act, V_0, Goal\rangle$ *is solvable by a sequential plan with no more than k actions if and only if:*

$$V_0 \in \Big|\Big|\Big\langle\Big(\bigcup_{\mathsf{a}\in Act} \mathsf{exeAct}(\mathsf{a})\Big)^{\leq k}\Big\rangle Goal\Big|\Big|.$$

5 Conclusion

Our logic DEL-PPAOC extends the lightweight epistemic logic EL-O in two directions. First, we generalise the 'knowing-whether' operator to a 'knowing-what' (or 'knowing the value') operator. Second, we add dynamic operators: all standard program operators of PDL, plus operators of parallel composition and of inclusive nondeterministic composition. We show how to model solvability of simple epistemic planning tasks by bounded parallel or sequential plans.

We conjecture that DEL-PPAOC satisfiability and model checking are both PSPACE-complete. One way of proving this is to combine the reduction of DL-PPA to DL-PA of (Herzig, Maris, and Vianey 2019) and that of DEL-PAO to DL-PA of (Herzig, Lorini, and Maffre 2015). From that, PSPACE membership of the problem of solvability of planning tasks can be obtained by polynomially eliminating finite iteration $\pi^{\leq k}$ by means of counters as done in (Herzig, Maris, and Vianey 2019). We believe that thanks to its decidability and relatively low complexity, DEL-PPAOC provides an interesting alternative to DEL-based epistemic planning.

Acknowledgements

We would like to thank the KR 2021 reviewers for their comments. Our work is partially supported by the EU ICT-48 2020 project TAILOR (No. 952215) and the ANR project CoPains (No. ANR-18-CE33-0012).

References

Andersen, M. B.; Bolander, T.; and Jensen, M. H. 2012. Conditional epistemic planning. In Fariñas del Cerro, L.; Herzig, A.; and Mengin, J., eds., *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings*, volume 7519 of *Lecture Notes in Computer Science*, 94–106. Springer.

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 27–33. AAAI Press.

Balbiani, P., and Boudou, J. 2018. Iteration-free PDL with storing, recovering and parallel composition: a complete axiomatization. *J. Log. Comput.* 28(4):705–731.

Bavelas, A. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America* 22(6):725–730.

Benevides, M. R. F., and Schechter, L. M. 2014. Propositional dynamic logics for communicating concurrent programs with CCS's parallel operator. *J. Log. Comput.* 24(4):919–951.

Bolander, T., and Andersen, M. B. 2011. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics* 21(1):9–34.

Bolander, T.; Charrier, T.; Pinchinat, S.; and Schwarzentruber, F. 2020. DEL-based epistemic planning: Decidability and complexity. *Artif. Intell.* 287:103304.

Boudou, J.; Herzig, A.; and Troquard, N. 2021. Resource separation in dynamic logic of propositional assignments. *J. Log. Algebraic Methods Program.* 121:100683.

Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; and Régnier, P. 2016. A simple account of multi-agent epistemic planning. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, 193–201.

Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; and Régnier, P. 2019. The epistemic gossip problem. *Discret. Math.* 342(3):654–663.

Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; Perrotin, E.; and Régnier, P. 2020a. A lightweight epistemic logic and its application to planning. *Artificial Intelligence* 103437.

Cooper, M. C.; Herzig, A.; Maris, F.; Perrotin, E.; and Vianey, J. 2020b. Lightweight parallel multi-agent epistemic planning. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, 274–283.

Herzig, A.; Lorini, E.; and Maffre, F. 2015. A poor man's epistemic logic based on propositional assignment and higher-order observation. In van der Hoek, W.; Holliday, W. H.; and Wang, W., eds., *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, volume 9394 of *Lecture Notes in Computer Science*, 156–168. Springer.

Herzig, A.; Maris, F.; and Vianey, J. 2019. Dynamic logic of parallel propositional assignments and its applications to planning. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 5576–5582. ijcai.org.

Knödel, W. 1975. New gossips and telephones. *Discrete Mathematics* 13(1):95.

Landau, H. G. 1954. The distribution of completion times for random communication in a task-oriented group. *The Bulletin of Mathematical Biophysics* 16(3):187–201.

Miller, J. S., and Moss, L. S. 2005. The undecidability of iterated modal relativization. *Stud Logica* 79(3):373–407.

Perrotin, E. forthcoming. *Lightweight approaches to reasoning about knowledge and belief*. Ph.D. Dissertation, Univ. Toulouse III, Toulouse.

Plaza, J. 2007. Logics of public communications. *Synthese* 158(2):165–179.

van Ditmarsch, H.; Gattinger, M.; and Ramezanian, R. 2020. Everyone knows that everyone knows: Gossip protocols for super experts. *CoRR* abs/2011.13203.

van Ditmarsch, H.; van der Hoek, W.; and Kuijer, L. B. 2020. The logic of gossiping. *Artif. Intell.* 286:103306.

Wang, Y., and Fan, J. 2013. Knowing that, knowing what, and public communication: Public announcement logic with kv operators. In *IJCAI*, volume 13, 1147–1154.

Wang, Y. 2018. Beyond knowing that: A new generation of epistemic logics. In *Jaakko Hintikka on Knowledge and Game-Theoretical Semantics*. Springer. 499–533. arXiv preprint arXiv:1605.01995, 2016.