

Concurrent Games in Dynamic Epistemic Logic

Bastien Maubert¹, Sophie Pinchinat², François Schwarzentruber² and Silvia Stranieri¹

¹Università degli Studi di Napoli “Federico II”, Italy

²Université de Rennes, IRISA, CNRS, France

Abstract

Action models of Dynamic Epistemic Logic (DEL) represent precisely how actions are perceived by agents. DEL has recently been used to define infinite multi-player games, and it was shown that they can be solved in some cases. However, the dynamics being defined by the classic DEL update product for individual actions, only turn-based games have been considered so far. In this work we define a concurrent DEL product, propose a mechanism to resolve conflicts between actions, and define concurrent DEL games. As in the turn-based case, the obtained concurrent infinite game arenas can be finitely represented when all actions are public, or all are propositional. Thus we identify cases where the strategic epistemic logic ATL_K^* can be model checked on such games.

1 Introduction

The discipline of *distributed synthesis* seeks solutions for automating the design of artificial agents that interact and make decisions in order to achieve some service. The main challenge is to synthesize strategies for each agent of a team collaborating towards a common goal, against opponents, usually in an imperfect information setting.

As advocated in the Dynamic Epistemic Logic (DEL) approach to *epistemic planning*, we consider multi-agent systems described symbolically by *DEL presentations*, which consist in an initial epistemic state (actual world and worlds considered possible by some agents), and epistemic actions that agents may perform. An epistemic action, besides the standard precondition and effects of the action, also describes the agents’ ability to perceive its execution. Synthesising strategies was studied in the single-agent case [Bolander and Andersen, 2011; Aucher and Bolander, 2013; Bolander *et al.*, 2015; Bolander, 2017], and in a multi-player turn-based game setting [Maubert *et al.*, 2019; Maubert *et al.*, 2020].

However, concurrent execution of actions has not been considered so far in epistemic planning. Some works consider concurrent execution of abstract actions (as in *concurrent game structures* [de Alfaro and Henzinger, 2000]), or concurrent execution of purely epistemic actions without effects on the world [van Eijck *et al.*, 2011; Ågones and van

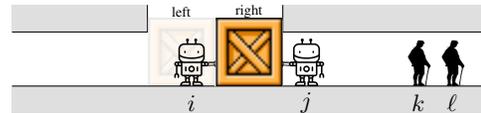


Figure 1: The two-robot coordination example.

Ditmarsch, 2011; Benevides and Lima, 2019] or only public [Lima, 2014]. But to our knowledge, concurrent execution of arbitrary epistemic actions with explicit effects has never been studied. And yet they are essential for modelling realistic situations.

Consider the two-robot coordination example of Figure 1. Some container can be placed either left or right. Two robots (or agents), i and j , can either wait or push the container, which incidentally is energy-consuming. Both agents waiting, the container stays in place. Otherwise, if i pushes while j waits, the container moves right (if not already there), and, symmetrically, if j pushes while i waits, it moves left (if not already there). Agents pushing concurrently results in a *conflict*. To determine the effect of such conflicting actions, one solution is to consider that they are *blocking*, as done for instance in [Fox and Long, 2003] for in temporal planning, or facts involved in conflicts are just maintained [Lima, 2014]. Another option consists in selecting a maximal subset of non-conflicting actions, as done for instance in [Eshuis and Wieringa, 2001] in UML work-flow modelling.

In this paper, we adopt the latter solution, which is more challenging. A major additional difficulty with respect to this latter UML setting arises from epistemic features of actions of the DEL framework. In our robot example, while robots i and j perfectly perceive pushing and waiting actions, the two other agents k and l cannot distinguish them. Note that this issue is not tackled in [Lima, 2014] since actions are public.

Our first contribution is to define a *DEL concurrent update product* that provides the dynamics of concurrent epistemic actions. This new product yields a non-deterministic dynamics controlled by a *scheduler*, whose role is to resolve conflicts. On the basis of this product we show how finite DEL presentations can generate infinite concurrent game arenas.

Our second contribution is a proof that distributed synthesis can be solved for such games for two cases: the case where all actions are public (perfectly observed by everyone), and the case where all actions have propositional pre- and post-

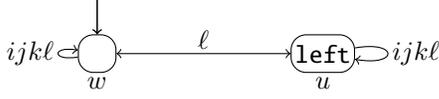


Figure 2: The epistemic state of Figure 1.

conditions and information among the agents is hierarchical (Theorems 4 and 7). To prove this we extend to the concurrent setting some results from [Maubert *et al.*, 2019; Maubert *et al.*, 2020], namely, that the infinite concurrent game arenas arising from DEL game presentations can be finitely represented when actions are all public or all propositional. We then transfer existing results on the model-checking problem for the epistemic strategic logic ATL_K^* [Van der Hoek and Wooldridge, 2003] to obtain, in particular, decidability of distributed synthesis for rich temporal epistemic objectives.

Remark that our work is unrelated to the *concurrent dynamic logic* from [van Benthem *et al.*, 2007], where several games are played concurrently. Our work also differs from *concurrent dynamic epistemic logic* [van Ditmarsch *et al.*, 2003], where concurrency represents some form of alternation [Chandra and Stockmeyer, 1976].

Outline: In Section 2, we provide all the necessary material on DEL presentations. In Section 3 we present the concurrent update product, and we use it in Section 4 to define DEL games and establish our main results.

2 Background on DEL

We fix a finite set of *agents* $Ag = \{1, \dots, N\}$ and a countable set of *atomic propositions* AP . The language of epistemic logic, noted \mathcal{L}_K , is defined by the grammar $\varphi ::= true \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi$ where $p \in AP$ and $i \in Ag$. Formula $K_i\varphi$ reads as “agent i knows that φ ”. Ruling out modality K_i yields *propositional logic*.

2.1 Epistemic Models and States

Epistemic models are relational structures that represent agents’ knowledge.

Definition 1 (Epistemic models). An *epistemic model* is a structure $\mathcal{M} = (W, (\approx_i)_{i \in Ag}, \lambda)$ where W is a finite non-empty set of *worlds*, for each agent $i \in Ag$, the equivalence relation $\approx_i \subseteq W \times W$ is an *epistemic relation* that reflects agents i ’s information (or knowledge), and $\lambda : W \rightarrow 2^{AP}$ is a *valuation* function. An *epistemic state* is a pair (\mathcal{M}, w) where \mathcal{M} is an epistemic model and w is a world of \mathcal{M} representing the *actual world*.

An epistemic state (\mathcal{M}, w) will often be written w when the model \mathcal{M} is clear from the context.

Example 1. The epistemic state of Figure 2 represents the situation of Figure 1. Proposition `left` holds when the container is on the left. In the actual world w , `left` is false since the container is on the right and we assume that agents i, j and k know this fact, but that ℓ does not: he does not distinguish between world w and the world u where the container is on the left.

The semantics of \mathcal{L}_K is based on epistemic states.

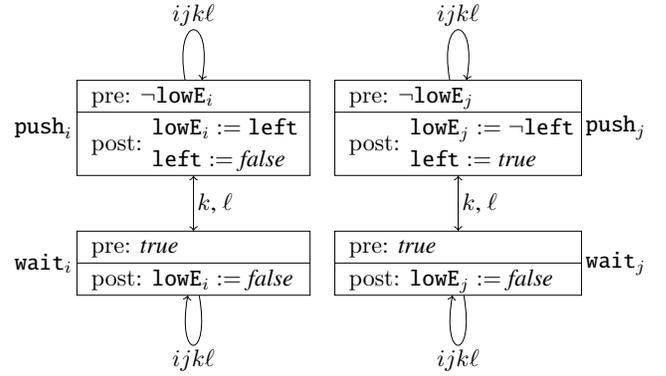


Figure 3: The action model of the two-robot example.

Definition 2 (Semantics of \mathcal{L}_K). Let (\mathcal{M}, w) be an epistemic state. We define the sentence “Formula φ holds in (\mathcal{M}, w) ”, written $(\mathcal{M}, w) \models \varphi$ or $w \models \varphi$, by induction on φ : $w \models true$; $w \models p$ whenever $p \in \lambda(w)$; $w \models \neg\varphi$ whenever $w \not\models \varphi$; $w \models \varphi_1 \wedge \varphi_2$ whenever $w \models \varphi_1$ and $w \models \varphi_2$; and $w \models K_i\psi$ whenever $(\mathcal{M}, u) \models \psi$, for all worlds u such that $w \approx_i u$.

2.2 Action Models and Epistemic Actions

Action models ([Baltag *et al.*, 1998], [Van Ditmarsch *et al.*, 2007]) gather several “concrete” actions (with preconditions and effects [van Ditmarsch and Kooi, 2006]) with epistemic relations describing how the concrete actions are perceived.

Definition 3 (Action models). An *action model* is a structure $\mathcal{A} = (A, (\approx_i)_{i \in Ag}, pre, post)$ where A is a nonempty finite set of *actions*, $\approx_i \subseteq A \times A$ is the *epistemic relation* of agent $i \in Ag$, the *precondition* function $pre : A \rightarrow \mathcal{L}_K$ specifies the guard for an action to be executable, and the *postcondition* is a partial function $post : A \times AP \rightarrow \mathcal{L}_K$ that describes the effect of actions over atomic propositions; when $post(a, p)$ is undefined, p is not affected by a . An *epistemic action* is a pair (\mathcal{A}, a) where $a \in A$ denotes the action actually executed.

Example 2. Figure 3 describes the action model for the two-robot coordination example. Action `pushi` represents agent i pushing the container to the right. Agent i can execute `pushi` whenever its energy is high, i.e., when proposition `lowEi` is false (`lowEi` and `lowEj` are initially false); this action consumes energy if it is effective (i.e., the container was not already on the right): in that case the postcondition makes proposition `lowEi` true. Action `pushj` for agent j is defined similarly. Finally, the waiting action of robots, represented by `waiti` and `waitj` respectively, have the *true* precondition and the effect of refilling energy (i.e. turning `lowEi` to false). To keep our example simple, we suppose that all agents but ℓ distinguish between `pushi` and `waiti` even if the box is on the right, and similarly for `pushj` and `waitj`.

We exhibit two particular kinds of epistemic actions.

Definition 4 (Propositional, public action). An action model \mathcal{A} is *propositional* if for every concrete action a of \mathcal{A} and each proposition p , formulas $pre(a)$ and $post(a)(p)$ are propositional. An epistemic action (\mathcal{A}, a) is *public* if for all agents $i, a \approx_i a'$ if and only if $a' = a$.

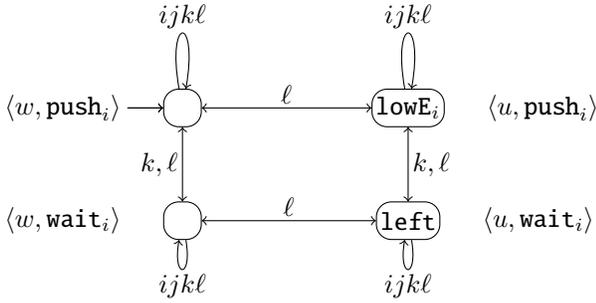


Figure 4: Epistemic state after agent i pushed the container.

2.3 Update Product

Execution of epistemic actions may change the world but also agents' information. It is formalised by the *update product* between epistemic models and action models.

Definition 5 (Update product). The *update product* of epistemic model $\mathcal{M} = (W, (\approx_i)_{i \in Ag}, \lambda)$ with action model $\mathcal{A} = (A, (\approx_i)_{i \in Ag}, \text{pre}, \text{post})$ is the epistemic model $\mathcal{M} \otimes \mathcal{A} := (W^\otimes, (\approx_i^\otimes)_{i \in Ag}, \lambda^\otimes)$, where:

- $W^\otimes = \{(w, a) \in W \times A \mid (\mathcal{M}, w) \models \text{pre}(a)\}$;
- $(w, a) \approx_i^\otimes (u, b)$ whenever $w \approx_i u$ and $a \approx_i b$;
- $p \in \lambda^\otimes(w, a)$ whenever $(\mathcal{M}, w) \models \text{post}(a, p)$.

We lift the product to epistemic states and epistemic actions: whenever $\mathcal{M}, w \models \text{pre}(a)$, we let $(\mathcal{M}, w) \otimes (\mathcal{A}, a)$ be the epistemic state $(\mathcal{M} \otimes \mathcal{A}, (w, a))$.

Example 3. Figure 4 shows $(\mathcal{M} \otimes \mathcal{A}, (w, \text{push}_i))$, that describes the epistemic state after execution of action push_i from the epistemic state of Figure 1 (strictly speaking, it only displays worlds connected to the actual one). The physical situation has not changed: the container is still on the right. However agent ℓ , who initially considered it possible that the container is on the left, and cannot distinguish between push_i and wait_i , still considers it possible that the container is on the left. Some information has changed though: initially, agent ℓ knew that robots had energy, but now he considers it possible that robot i 's energy has become low.

3 Concurrent Actions

Fix an epistemic state (\mathcal{M}, w) and an action model \mathcal{A} . From now on, atomic propositions are partitioned into *shared propositions* (AP^s) that all agents can modify, and *private ones*: AP_i^p is the set of private propositions of agent i . As a result, $AP = AP^s \uplus \bigsqcup_{i \in Ag} AP_i^p$, where \uplus is the disjoint union. An agent can play any action that does not modify private propositions of others. We gather in set A_i those actions, namely those whose postconditions are undefined on $\cup_{j \neq i} AP_j^p$. A *joint action* is a tuple $\vec{a} = \langle a_1, \dots, a_N \rangle \in \prod_{i \in Ag} A_i$, and we let \vec{a}_j denote action a_j . Finally, a joint action \vec{a} is *available* in w when every *individual* action \vec{a}_j can be executed in w : $(\mathcal{M}, w) \models \text{pre}(\vec{a}_1) \wedge \dots \wedge \text{pre}(\vec{a}_n)$.

3.1 Conflicts

We define a formula $\text{noconflict}(\vec{a})(p)$ expressing that all individual actions of a joint action \vec{a} agree on their effect (if any)

on proposition p . We first introduce the set $Ag(\vec{a}, p)$ of agents whose individual action in \vec{a} has an effect on p . Formally, $Ag(\vec{a}, p)$ is the set $\{i = 1..N \mid \text{post}(\vec{a}_i)(p) \text{ is defined}\}$. We then let $\text{noconflict}(\vec{a})(p)$ be the formula

$$\bigwedge_{i \in Ag(\vec{a}, p)} \text{post}(\vec{a}_i)(p) \vee \bigwedge_{i \in Ag(\vec{a}, p)} \neg \text{post}(\vec{a}_i)(p).$$

Now, in a situation where all individual actions of the joint action \vec{a} agree on their effect on each proposition, the effect of executing them concurrently can be determined unambiguously. Such a situation is expressed by the formula $\text{noconflict}(\vec{a}) := \bigwedge_{p \in AP} \text{noconflict}(\vec{a})(p)$.

Definition 6. A joint action \vec{a} is *non-conflicting* in w if $\mathcal{M}, w \models \text{noconflict}(\vec{a})$. Otherwise \vec{a} is *conflicting* in w .

Example 4. In our running example, $\langle \text{push}_i, \text{push}_j \rangle$ is conflicting in both worlds w and u .

To resolve conflicts, we resort to a *scheduler*.

3.2 The Role of the Scheduler

Given a conflicting available joint action, the scheduler selects a maximal subset of consistent individual actions, and *inhibits* the remaining actions by “cancelling” their effects on shared variables. An inhibited action of agent i may still have effects on AP_i^p , which may differ from the original ones. The inhibiting mechanism is implemented by a *ghost mapping* $\circ : A \rightarrow A$ with the following properties: for every agent $i \in Ag$ and every action $a \in A_i$, the *ghost* $\circ(a)$ (written a_\circ) must be in A_i and $\text{post}(a_\circ)$ is undefined on AP^s . This assumption ensures that no conflict arises when inhibiting actions. We also assume that $\circ(a)$ is executable as long as a is, i.e., that $\text{pre}(a_\circ)$ is a logical consequence of $\text{pre}(a)$.

Example 5. For the robots, $AP^s = \{\text{left}\}$, $AP_i^p = \{\text{lowE}_i\}$, $AP_j^p = \{\text{lowE}_j\}$, and $AP_k^p = AP_\ell^p = \emptyset$. Facing the conflicting joint action $\langle \text{push}_i, \text{push}_j \rangle$, the scheduler can choose between $\langle \text{push}_{i_\circ}, \text{push}_j \rangle$ and $\langle \text{push}_i, \text{push}_{j_\circ} \rangle$, and we set $\text{push}_{i_\circ} := \text{wait}_i$ and $\text{push}_{j_\circ} := \text{wait}_j$.

The epistemic relations from a ghost can be set in many different ways, and the choice is a design matter. For example, assume that ghost actions are distinct from those that can be played by agents, and that ghosts are not epistemically related to those playable actions. Then the scheduler's selection is public: indeed, when for joint action \vec{a} scheduler inhibits the action of agent i , all agents are implicitly informed that agent i is discarded.

Definition 7. If $A = A_\bullet \uplus A_\circ$ is split between *playable* actions A_\bullet and *ghosts* ones A_\circ , and A_\bullet and A_\circ are not related by any \approx_i , we say that the scheduler is *public*.

In the sequel, we consider that action models \mathcal{A} are implicitly equipped with a ghost mapping.

3.3 Concurrent Update Product

To define the concurrent execution of possibly conflicting actions, we introduce a *concurrent update product* \boxplus that generalises the classic update product \otimes (Definition 5).

Before defining this product, we formalise the possible choices for the scheduler. We say that joint action \vec{b} is a *sub-action* of joint action \vec{a} , written $\vec{b} \preceq \vec{a}$, whenever \vec{b} is obtained

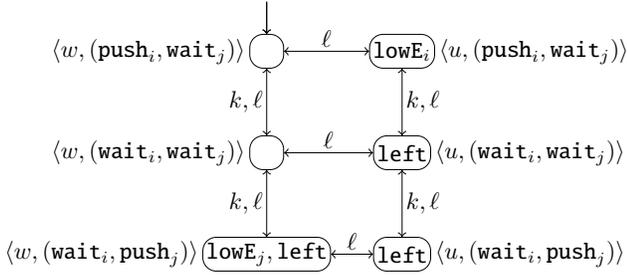


Figure 5: The model $(\mathcal{M} \boxplus \mathcal{A}, \langle w, (\text{push}_i, \text{wait}_j) \rangle)$ (for readability, all reflexive edges are missing).

from \vec{a} by replacing some individual actions by their ghosts. Formally, $\vec{b} \preceq \vec{a}$ if $\vec{b}_i \in \{\vec{a}_i, (\vec{a}_i)_o\}$, for every $1 \leq i \leq N$. Remark that, by definition of the ghost mapping, if \vec{a} is available in w , so are its sub-actions.

Given a joint action \vec{a} , we let $\text{Max}(\vec{a}, w)$ be the set of joint actions composed of the \preceq -maximal sub-actions of \vec{a} non-conflicting in w . Elements of $\text{Max}(\vec{a}, w)$ are therefore joint actions non-conflicting in w where a minimal number of actions are inhibited. Notice that $\text{Max}(\vec{a}, w) = \{\vec{a}\}$ as soon as \vec{a} is non-conflicting in w .

Example 6. The set $\text{Max}(\langle \text{push}_i, \text{push}_j \rangle, w)$ is $\{\langle \text{wait}_i, \text{push}_j \rangle, \langle \text{push}_i, \text{wait}_j \rangle\}$.

We now define the concurrent update product, that implements the execution of joint actions.

Definition 8 (Concurrent update product). The *concurrent update product* of \mathcal{M} and \mathcal{A} is the Kripke model $\mathcal{M} \boxplus \mathcal{A} = (\mathcal{W}^\boxplus, (\approx_i^\boxplus)_{i \in \text{Ag}}, \lambda^\boxplus)$, where:

- $\mathcal{W}^\boxplus = \{(w, \vec{b}) \in \mathcal{W} \times \mathcal{A}^N \mid \vec{b} \in \text{Max}(\vec{a}, w)\}$
for some joint action \vec{a} available in w ;
- $(w, \vec{b}) \approx_i^\boxplus (u, \vec{c})$ if $w \approx_i u$ and $\vec{b}_j \approx_i \vec{c}_j$ for all j ;
- $p \in \lambda^\boxplus(u, \vec{b})$ if $(\mathcal{M}, u) \models \bigwedge_{i \in \text{Ag}} (\vec{b}_i, p)$.

We can now define the set of epistemic states that may result from executing a joint action \vec{a} in (\mathcal{M}, w) :

$$(\mathcal{M}, w) \boxplus \vec{a} := \{(\mathcal{M} \boxplus \mathcal{A}, (w, \vec{b})) \mid \vec{b} \in \text{Max}(\vec{a}, w)\}.$$

This set is a singleton when \vec{a} is non-conflicting in w . Remark that the component-wise definition of concurrent epistemic relations \approx_i^\boxplus in Definition 8 makes the *identity* of who performs an action common knowledge.

Example 7. The pointed concurrent update product $(\mathcal{M} \boxplus \mathcal{A}, \langle w, (\text{push}_i, \text{wait}_j) \rangle)$ of Figure 5 belongs to $(\mathcal{M}, w) \boxplus \langle \text{push}_i, \text{push}_j \rangle$. Action push_j was inhibited, but agent ℓ considers it possible that lowE_j is now true.

4 Concurrent DEL Games

In this section, we show how DEL can represent infinite concurrent game arenas, in which players act simultaneously. We then show how, as for turn-based games [Maubert *et al.*, 2019; Maubert *et al.*, 2020] these infinite game arenas can in some cases be folded back into finite ones, when actions are all public, or when they are all propositional. Finally, we show that

in these cases, one can model check the epistemic strategic logic ATL_K^* on DEL-represented concurrent games.

4.1 Concurrent Game Arenas

Here we recall (nondeterministic) concurrent game arenas with imperfect information and related notions.

Definition 9. A concurrent game arena is a tuple $G = (\text{Act}, V, v_{\text{ini}}, (\text{Act}_i)_{i \in \text{Ag}}, \Delta, (\approx_i)_{i \in \text{Ag}}, \lambda)$, where

- Act is a non-empty set of actions,
- V is a non-empty set of positions,
- $v_{\text{ini}} \in V$ is an initial position,
- $\text{Act}_i : V \rightarrow 2^{\text{Act}} \setminus \{\emptyset\}$ is a repertoire function for i ,
- $\Delta \subseteq V \times \text{Act}^N \times V$ is a transition relation,
- $\approx_i \subseteq V \times V$ is an indistinguishability equivalence relation for agent i , and
- $\lambda : V \rightarrow 2^{AP}$ is a valuation function.

In a position v each agent i chooses an available action $a_i \in \text{Act}_i(v)$, yielding a joint action $\vec{a} = \langle a_1, \dots, a_N \rangle$. The set of joint actions available in v is written $\text{JAct}(v)$. The game then moves nondeterministically to some position $v' \in \Delta(v, \vec{a})$. If, for every position v and joint action \vec{a} , $\{v' \mid (v, \vec{a}, v') \in \Delta\}$ is a singleton, then G is *deterministic* and we may represent the transition relation as a function $\delta : V \times \text{Act}^N \rightarrow V$.

A play $\pi = v_0 v_1 v_2 \dots$ is an infinite sequence of positions such that for all $k \in \mathbb{N}$, there exists $\vec{a} \in \text{Act}^N$ such that $v_{k+1} \in \Delta(v_k, \vec{a})$. We let $\pi_k = v_k$ and $\pi_{\leq k} = v_0 v_1 \dots v_k$. A history $h = v_0 v_1 \dots v_n$ is a finite non-empty prefix of a play, $\text{last}(h) = v_n$ is the last position in h , $|h| = n + 1$ is its length, and Hist^G is the set of histories in G . We write $\lambda(h)$ for $\lambda(\text{last}(h))$.

As usual for agents with synchronous perfect recall (see for instance [Fagin *et al.*, 1995]), the indistinguishability relation of each agent i is lifted to histories as follows: $h \approx_i h'$ if $|h| = |h'|$ and $h_k \approx_i h'_k$ for every $k < |h|$.

A (*uniform*) *strategy* σ for an agent i is a function σ_i that associates an action to any history and such that, for every pair of indistinguishable histories $h \approx_i h'$, $\sigma(h) = \sigma(h')$. Classically, we assume that $\text{Act}_i(v) = \text{Act}_i(v')$ whenever $v \approx_i v'$. A *strategy profile* for a coalition of agents $C \subseteq \text{Ag}$ is a tuple $\sigma_C = \langle \sigma_i \rangle_{i \in C}$, and Σ_C is the set of strategy profiles for coalition C . Finally, an *outcome* of a strategy profile σ_C from a position v_0 is a play starting in position v_0 and in which agents in C follow the strategies in Σ_C : formally, a play $\pi = v_0 v_1 \dots$ is an outcome of $\langle \sigma_i \rangle_{i \in C}$ from v_0 if, for every $k \in \mathbb{N}$, there is a joint action \vec{a} such that $v_{k+1} \in \Delta(v_k, \vec{a})$ and $(\vec{a})_i = \sigma_i(v_0 \dots v_k)$, for all $i \in C$. We let $\text{out}(h, \sigma_C)$ be the set of outcomes of σ_C from position v .

Definition 10. Given a concurrent game arena $G = (\text{Act}, V, v_{\text{ini}}, (\text{Act}_i)_{i \in \text{Ag}}, \Delta, (\approx_i)_{i \in \text{Ag}}, \lambda)$, we define the *unfolding* of G as the game arena $G^{\text{unf}} = (\text{Act}, V', v'_{\text{ini}}, (\text{Act}'_i)_{i \in \text{Ag}}, \Delta', (\approx'_i)_{i \in \text{Ag}}, \lambda')$ where $V' = \text{Hist}^G$, $v'_{\text{ini}} = v_{\text{ini}}$, for every $h \in \text{Hist}^G$, $\text{Act}'_i(h) = \text{Act}_i(\text{last}(h))$, $\lambda'(h) = \lambda(\text{last}(h))$, \approx'_i is the synchronous perfect-recall lifting of \approx_i to histories, and $\Delta' = \{(h, \vec{a}, h \cdot v) \mid (\text{last}(h), \vec{a}, v) \in \Delta\}$.

We say that two game structures \mathcal{G} and \mathcal{G}' are *equivalent* whenever their unfoldings are isomorphic.

4.2 DEL Presentations of Game Arenas

We use DEL models to define infinite concurrent arenas:

Definition 11. A *DEL game presentation*, or *DEL game*, is a tuple $\mathfrak{G} = \langle \mathcal{M}, w, \mathcal{A} \rangle$ where (\mathcal{M}, w) is an initial epistemic state and \mathcal{A} is an action model.

The game starts in the initial epistemic state (\mathcal{M}, w) . In each round, each agent i chooses an action $a \in A_i$ available in the current state (\mathcal{M}', w') , resulting in a joint action \vec{a} (we assume that each agent always has at least one available action; if needed we may add a dummy action in the model). The next epistemic state is nondeterministically chosen by the scheduler among $(\mathcal{M}', w') \boxplus \vec{a}$ (this choice is nontrivial if \vec{a} is conflicting), and the game goes on.

After n rounds in which the players chose joint actions $\vec{a}_1, \dots, \vec{a}_n$, the epistemic state is of the form $\mathcal{M}\mathcal{A}^n, (w, \vec{b}_1, \dots, \vec{b}_n)$, where $\mathcal{M}\mathcal{A}^n$ is defined by letting $\mathcal{M}\mathcal{A}^0 = \mathcal{M}$ and $\mathcal{M}\mathcal{A}^{n+1} = \mathcal{M}\mathcal{A}^n \boxplus \mathcal{A}$, and each \vec{b}_{k+1} is a maximal sub-action of \vec{a}_{k+1} consistent in $\mathcal{M}\mathcal{A}^k, (w, \vec{b}_1, \dots, \vec{b}_k)$. In the following we may write $w \vec{b}_1 \dots \vec{b}_n$ instead of $(w, \vec{b}_1, \dots, \vec{b}_n)$, and call it a history. The *length* of a history in $\mathcal{M}\mathcal{A}^n$ is defined as $|w \vec{b}_1 \dots \vec{b}_n| = n$. Since the length of a history determines the epistemic model to which it belongs, we may omit it and write $w \vec{b}_1 \dots \vec{b}_n \models \varphi$ instead of $\mathcal{M}\mathcal{A}^n, w \vec{b}_1 \dots \vec{b}_n \models \varphi$. We require that players know their available actions: if $w \vec{b}_1 \dots \vec{b}_n \approx_i u \vec{c}_1 \dots \vec{c}_n$, then the same actions are available to i in both worlds.

A DEL presentation denotes a concurrent game arena, where positions are epistemic states attainable from the initial one, and moves are obtained by applying the concurrent update product of Definition 8. Because this product yields several epistemic states, this concurrent game arena is non-deterministic.

Definition 12. Given a DEL game presentation $\mathfrak{G} = \langle \mathcal{M}, w, \mathcal{A} \rangle$ we define the concurrent game arena $G_{\mathfrak{G}} = (\mathbf{A}, V, v_{\text{ini}}, (\text{Act}_i)_{i \in Ag}, \Delta, (\approx_i)_{i \in Ag}, \lambda)$ where, writing $\mathcal{M}\mathcal{A}^n = (W^n, (\approx_i^n)_{i \in Ag}, \lambda^n)$ for every n :

- \mathbf{A} is the set of actions in \mathcal{A} ,
- the set of positions is $V = \bigcup_{n \in \mathbb{N}} W^n$,
- the initial position is $v_{\text{ini}} = w$,
- $\text{Act}_i(v) = \{a \in A_i \mid v \models \text{pre}(a)\}$,
- $\Delta = \{(v, \vec{a}, v') \mid v' \in v \boxplus \vec{a}\}$,
- $v \approx_i v'$ if $|v| = |v'| = n$ and $v \approx_i^n v'$, and
- $\lambda(v) = \lambda^{|v|}(v)$.

We want to reason about strategic abilities and knowledge on infinite concurrent games given as DEL game presentations. To do so we consider the logic ATL_K^* .

Definition 13. The syntax of ATL_K^* is given by the following grammar:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid K_i\varphi \mid \langle C \rangle\psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \bigcirc\psi \mid \psi_1 \mathcal{U} \psi_2 \end{aligned}$$

with $p \in AP$, $C \subseteq Ag$ and $i \in Ag$.

Formula $\langle C \rangle\psi$ reads as ‘‘coalition C has a strategy profile to ensure ψ ’’, and the meaning of other operators is as usual. We let $\approx_C = \bigcup_{i \in C} \approx_i$, and define the semantics of ATL_K^* .

Definition 14. Let G be a concurrent game arena, v a position and π a play. The semantics of ATL_K^* is defined as follows, where h is a history in G , π is a play and $i \in \mathbb{N}$ is a point in time (we omit boolean cases):

$$\begin{aligned} G, h &\models p && \text{if } p \in \lambda(h) \\ G, h &\models K_i\varphi && \text{if } \forall h' \in \text{Hist}^G \text{ s.t. } h \approx_i h', G, h' \models \varphi \\ G, h &\models \langle C \rangle\psi && \text{if } \exists \sigma_C \in \Sigma_C \text{ s.t. } \forall h' \approx_C h, \\ &&& \forall \pi \in \text{out}(h', \sigma_C), G, \pi \models \psi \\ G, \pi &\models \varphi && \text{if } \pi[0] \models \varphi \\ G, \pi &\models \bigcirc\psi && \text{if } G, \pi_{\geq 1} \models \psi \\ G, \pi &\models \psi_1 \mathcal{U} \psi_2 && \text{if } \exists i \geq 0 \text{ s.t. } G, \pi_{\geq i} \models \psi_2 \text{ and} \\ &&& \forall 0 \leq j < i, G, \pi_{\geq j} \models \psi_1 \end{aligned}$$

Remark 1. We use the *uninformed semantics* for the knowledge operators, which does not depend on the strategies used by the agents, and is the usual one used in epistemic planning (see [Puchala, 2010; Maubert, 2014; Maubert *et al.*, 2020] for more details). Also, we use the *subjective semantics* for strategic operators, in contrast with the *objective* one that only asks strategies to be winning from the actual current history (see [Bulling and Jamroga, 2014]).

Given a game arena G with initial position v_{ini} , we write $G \models \varphi$ if $G, v_{\text{ini}} \models \varphi$. We study the following model-checking problem: given a DEL game presentation \mathfrak{G} and a formula $\varphi \in \text{ATL}_K^*$, do we have $G_{\mathfrak{G}} \models \varphi$?

4.3 Decidability for Public Actions

A *deterministic* game arena with transition function δ is said to have *only public actions* if, for all positions v, v' and joint actions \vec{a}, \vec{a}' such that $\vec{a} \neq \vec{a}'$, we have $\delta(v, \vec{a}) \not\approx_i \delta(v', \vec{a}')$. The following is known for ATL_K^* on such arenas:

Theorem 1 ([Belardinelli *et al.*, 2017]). *Model checking ATL_K^* on finite deterministic concurrent game structures with only public actions is 2-EXPTIME-complete.*

First we prove that DEL games with public actions can be finitely represented. Our proof yields nondeterministic game arenas, that we will then transform into deterministic ones that have only public actions in the sense of [Belardinelli *et al.*, 2017]. We will need the following result on the concurrent update product with public action models and a public scheduler (see Definition 7):

Lemma 2. *Let \mathcal{M} be an epistemic model and \mathcal{A} an event model. If the scheduler is public, \mathcal{A} has only public actions and its ghost mapping $\circ : \mathbf{A}_{\bullet} \rightarrow \mathbf{A}_{\circ}$ is injective, then for all worlds $w, w' \in \mathcal{M}$, all joint actions \vec{a}, \vec{a}' , if $\vec{a} \neq \vec{a}'$ then for all $(w, \vec{b}) \in (\mathcal{M}, w) \boxplus \vec{a}$ and $(w', \vec{b}') \in (\mathcal{M}, w') \boxplus \vec{a}'$, for all agents i , $(w, \vec{b}) \not\approx_i (w', \vec{b}')$.*

The injectivity assumption means that two different played actions cannot become the same after being inhibited, in which case the actions of DEL games could not be public in the sense of [Belardinelli *et al.*, 2017].

The proof that the infinite game arena $G_{\mathfrak{G}}$ induced by a DEL game \mathfrak{G} with public actions can be finitely represented is very similar to that in [Maubert *et al.*, 2020]. It relies on the fact that updating an epistemic state with a joint action made of public actions can only decrease the size of the obtained epistemic states (when removing their disconnected components) as long as their ghosts are public too. It follows that only finitely many different epistemic models can be generated from a given initial state, up to isomorphism.

Proposition 3. *Given a DEL game $\mathfrak{G} = \langle \mathcal{M}, w, \mathcal{A} \rangle$ with only public actions, one can build a finite game arena G equivalent to $G_{\mathfrak{G}}$ and of size exponential in $|\mathfrak{G}|$.*

Because of the scheduler, the resulting arena is nondeterministic. To obtain Theorem 4 below it essentially remains to show that we can transform these nondeterministic arenas into deterministic ones that have only public actions in the sense of [Belardinelli *et al.*, 2017].

Theorem 4. *Model checking ATL_K^* on DEL concurrent games with public actions, public scheduler and injective ghost mapping is 2-EXPTIME-complete.*

Proofsketch. The lower bound comes from LTL synthesis [Pnueli and Rosner, 1989]. For the upper bounds, let \mathfrak{G} be a DEL presentation and Φ an ATL_K^* formula. First, by Proposition 3 we build in exponential time a finite nondeterministic game arena $G = (\text{Act}, V, v_{\text{ini}}, (\text{Act}_i)_{i \in Ag}, \Delta, (\approx_i)_{i \in Ag}, \lambda)$ equivalent to $G_{\mathfrak{G}}$. We build a *deterministic* game arena G^d by adding a special agent, the scheduler s , who has perfect information and whose possible actions Act_s allow him to resolve the non-determinism. The agents choose their actions, and then the scheduler resolves nondeterminism. To do so we introduce intermediary positions of the form (v, \vec{a}) : in each round, in a position v , the agents in Ag choose a joint action \vec{a} (the scheduler can only play some dummy action), and the game moves deterministically to (v, \vec{a}) . Now all “normal” agents can only choose the dummy move, and the scheduler can choose a maximal consistent sub-action \vec{b} of \vec{a} , which determines the new position v' . Indistinguishability relations for all normal agents are as in G for positions v , and we let $(v, \vec{a}) \approx_i (v', \vec{a}')$ iff $v \approx_i v'$ and $\vec{a} = \vec{a}'$. This definition is correct because, by Lemma 2, when the agents choose different joint actions, they distinguish the resulting epistemic states no matter the choices of the scheduler. One can check that G^d has only public actions in the sense of [Belardinelli *et al.*, 2017]. We then duplicate all next operators in Φ to skip artificial positions of the form (v, \vec{a}) . To check Φ on G^d , the procedure from [Belardinelli *et al.*, 2017] takes time doubly exponential in the size of Φ and exponential in the size of G^d , which is itself of size exponential in the size of the original DEL game presentation \mathfrak{G} . \square

4.4 Decidability for Propositional Actions

A DEL game presentation presents *hierarchical information* if the set of agents Ag can be totally ordered ($i_1 < \dots < i_N$) so that $\approx_{i_i} \subseteq \approx_{i_{i+1}}$ for each $1 \leq i < N$, and similarly for concurrent game arenas. We prove that when, in a DEL game presentation, all actions are propositional actions and information is hierarchical, model checking ATL_K^* is decidable.

We first generalise to the concurrent setting a result from [Maubert *et al.*, 2019] which states that infinite turn-based DEL games induced by propositional models can be finitely represented. As for public actions, the game arena that we obtain is nondeterministic.

Proposition 5. *Let $\mathfrak{G} = \langle \mathcal{M}, w, \mathcal{A} \rangle$ be a DEL game where \mathcal{A} is propositional. One can build a finite nondeterministic game arena G equivalent to $G_{\mathfrak{G}}$, of size exponential in $|\mathfrak{G}|$.*

We will invoke the following result, that was stated for deterministic game arenas:

Theorem 6 ([Berthon *et al.*, 2017]). *Model checking ATL_K^* with uninformed semantics is decidable on deterministic game arenas with hierarchical information.*

The result in [Berthon *et al.*, 2017] is for ATL^* without knowledge operators, and for the objective semantics of strategic operators. But their bottom-up algorithm can be easily extended to deal with knowledge operators and subjective semantics: one first performs a powerset construction on the game arena to include sufficient information to evaluate knowledge operators positionally (for instance using *k-trees* [van der Meyden, 1998]), and then one evaluates knowledge and strategic modalities in a bottom-up fashion.

Thanks to Proposition 5 and a reduction to deterministic arenas similar to the one presented in the proof of Theorem 4, we can use Theorem 6 and obtain that:

Theorem 7. *Model checking ATL_K^* on DEL game presentations with propositional actions and hierarchical information is decidable.*

However the complexity is nonelementary, as it is already the case for multiplayer reachability games with hierarchical information [Peterson *et al.*, 2001]. The number of exponentials will be the maximum between the number of agents with different observation power and the alternation of knowledge operators in the formula.

5 Conclusion

We proposed a setting that permits the succinct specification of concurrent game arenas with epistemic actions. Conflicts are resolved by a scheduler that nondeterministically selects consistent subsets of actions, and inhibits the discarded ones. We established that on such games one can perform strategic reasoning, or more precisely one can model check the strategic epistemic logic ATL_K^* , in two cases: when actions are public (Theorem 4) and when actions are propositional and information is hierarchical among the agents (Theorem 7).

Note that we could tune the scheduler differently: in our robot example, a third choice for $\langle \text{push}_i, \text{push}_j \rangle$ could be $\langle \text{wait}_i, \text{wait}_j \rangle$, which would capture *blocking* conflicts. Technically, we could relax the selection to other sub-actions than \preceq -maximal ones, and the complexities established in our main theorems would still hold for such different schedulers.

One could also consider fairness requirements to ensure that the scheduler does not favor some agents in resolving conflicts, for instance requiring that if an agent infinitely often enters a conflict, then infinitely often his action is selected by the scheduler. Fairness constraints can often be expressed directly in ATL_K^* , in which case our main results still hold.

References

- [Ågotnes and van Ditmarsch, 2011] Thomas Ågotnes and Hans van Ditmarsch. What will they say? - public announcement games. *Synthese*, 179(Supplement-1):57–85, 2011.
- [Aucher and Bolander, 2013] G. Aucher and T. Bolander. Undecidability in epistemic planning. In *IJCAI'13*, 2013.
- [Baltag *et al.*, 1998] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements and common knowledge and private suspicions. In Itzhak Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98)*, Evanston, IL, USA, July 22-24, 1998, pages 43–56. Morgan Kaufmann, 1998.
- [Belardinelli *et al.*, 2017] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Verification of multi-agent systems with imperfect information and public actions. In *AAMAS'17*, pages 1268–1276, 2017.
- [Benevides and Lima, 2019] Mario Roberto Folhadela Benevides and Isaque Macalam Saab Lima. Dynamic epistemic logic with communication actions. *arXiv preprint arXiv:1902.01164*, 2019.
- [Berthon *et al.*, 2017] Raphaël Berthon, Bastien Maubert, and Aniello Murano. Decidability results for atl^* with imperfect information and perfect recall. In *AAMAS'17*, pages 1250–1258, 2017.
- [Bolander and Andersen, 2011] T. Bolander and M.B. Andersen. Epistemic planning for single and multi-agent systems. *J. Appl. Non-Class. Log.*, 21(1):9–34, 2011.
- [Bolander *et al.*, 2015] T. Bolander, M.H. Jensen, and F. Schwarzenruber. Complexity results in epistemic planning. In *IJCAI*, pages 2791–2797, 2015.
- [Bolander, 2017] Thomas Bolander. A gentle introduction to epistemic planning: The DEL approach. In *M4M@ICLA 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017.*, pages 1–22, 2017.
- [Bulling and Jamroga, 2014] Nils Bulling and Wojciech Jamroga. Comparing variants of strategic ability: how uncertainty and memory influence general properties of games. *Autonomous agents and multi-agent systems*, 28(3):474–518, 2014.
- [Chandra and Stockmeyer, 1976] Ashok K Chandra and Larry J Stockmeyer. Alternation. In *17th Annual Symposium on Foundations of Computer Science (sfcs 1976)*, pages 98–108. IEEE, 1976.
- [de Alfaro and Henzinger, 2000] Luca de Alfaro and Thomas A Henzinger. Concurrent omega-regular games. In *LICS'00*, pages 141–154. IEEE, 2000.
- [Eshuis and Wieringa, 2001] Rik Eshuis and Roel J. Wieringa. A real-time execution semantics for UML activity diagrams. In *FASE'01*, pages 76–90, 2001.
- [Fagin *et al.*, 1995] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about knowledge*, volume 4. MIT press Cambridge, 1995.
- [Fox and Long, 2003] Maria Fox and Derek Long. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.*, 20:61–124, 2003.
- [Lima, 2014] Tiago De Lima. Alternating-time temporal dynamic epistemic logic. *J. Log. Comput.*, 24(6):1145–1178, 2014.
- [Maubert *et al.*, 2019] Bastien Maubert, Sophie Pinchinat, and François Schwarzenruber. Reachability games in dynamic epistemic logic. In *IJCAI'19*, pages 499–505, 2019.
- [Maubert *et al.*, 2020] Bastien Maubert, Aniello Murano, Sophie Pinchinat, François Schwarzenruber, and Silvia Stranieri. Dynamic epistemic logic games with epistemic temporal goals. In *arXiv*, 2020.
- [Maubert, 2014] B. Maubert. *Logical foundations of games with imperfect information : uniform strategies*. PhD thesis, University of Rennes 1, France, 2014.
- [Peterson *et al.*, 2001] Gary Peterson, John Reif, and Salman Azhar. Lower bounds for multiplayer noncooperative games of incomplete information. *CAMWA*, 41(7):957–992, 2001.
- [Pnueli and Rosner, 1989] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190, 1989.
- [Puchala, 2010] Bernd Puchala. Asynchronous omega-regular games with partial information. In *MFCS'10*, pages 592–603, 2010.
- [van Benthem *et al.*, 2007] Johan van Benthem, Sujata Ghosh, Fenrong Liu, et al. *Modelling simultaneous games with concurrent dynamic logic*. Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 2007.
- [Van der Hoek and Wooldridge, 2003] Wiebe Van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia logica*, 75(1):125–157, 2003.
- [van der Meyden, 1998] Ron van der Meyden. Common knowledge and update in finite environments. *Inf. Comput.*, 140(2):115–157, 1998.
- [van Ditmarsch and Kooi, 2006] Hans P. van Ditmarsch and Barteld P. Kooi. Semantic results for ontic and epistemic change. *CoRR*, abs/cs/0610093, 2006.
- [van Ditmarsch *et al.*, 2003] Hans P. van Ditmarsch, Wiebe van der Hoek, and Barteld P. Kooi. Concurrent dynamic epistemic logic for MAS. In *AAMAS'03*, pages 201–208, 2003.
- [Van Ditmarsch *et al.*, 2007] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.
- [van Eijck *et al.*, 2011] Jan van Eijck, Floor Sietsma, and Yanjing Wang. Composing models. *Journal of Applied Non-Classical Logics*, 21(3-4):397–425, 2011.