# Reasoning about reachability and concurrency in DEL games

Silvia Stranieri[1]

[1]University of Naples Federico II, Italy

Abstract
Dynamic Epistemic Logic allows modelling high order knowledge and the evolution of what an agent knows over time. This work shows decidability results about reachability goals and concurrent execution of Dynamic Epistemic Logic games.

**Keywords**
Dynamic Epistemic Logic, Multi-agent systems, DEL games

## 1. Introduction

Dynamic Epistemic Logic (DEL) [1] is used to describe how actions affect the world, how agents perceive them, and how their knowledge changes during the execution of the game. In this work, DEL is investigated on reachability and concurrency aspects. We do this in two different directions. First, a setting in which agents are not active, but they simply observe a controller and an environment playing in turn and modifying their knowledge. In this case the controller synthesis problem is addressed. In the second setting, agents become players of an imperfect information game playing in coalitions against each other, addressing the distributed synthesis problem. Precisely, reachability goals are expressed through LTLK formulas, involving both temporal and knowledge operators, and decidability results over public actions and public announcements are provided [2]. We recall that an action is public when it is visible to all agents [3, 4, 5, 6], while a public announcement is a special case of public action with no effect besides epistemic ones.

DEL game concurrent executions are also considered in this work, by providing an opportune concurrent update product to define how actions played concurrently affect the epistemic model, relying on a scheduler to solve possible conflicts. The distributed synthesis is proved to be decidable when actions are public. This is obtained by reducing the problem to the model checking of $\mathsf{ATL}^*_{\mathsf{K}}$ [7].

## 2. DEL Models

In this section, the key aspects of epistemic logic are provided [8].

*OVERLAY 2022: 4th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, November 28, 2022, Udine, Italy*
silvia.stranieri@unina.it (S. Stranieri)

**Epistemic Model**

**Definition 2.1.** *Let $AP$ and $Agt$ be the set of atomic propositions and the set of agents respectively, an* epistemic model $\mathcal{M} = (W, (\preccurlyeq_a)_{a \in Agt}, \lambda)$ *is a tuple where*

- $W$ *is a set of* worlds *(or situations),*
- $\preccurlyeq_a \subseteq W \times W$ *is an* accessibility relation *for agent $a$, and*
- $\lambda : W \to 2^{AP}$ *is a* valuation function.

One can write $w \preccurlyeq_a u$ instead of $(w, u) \in \preccurlyeq_a$; the intended meaning of $w \preccurlyeq_a u$ is that when the actual world is $w$, agent $a$ considers that $u$ may be the actual world. The valuation function $\lambda$ provides the subset of atomic propositions that hold in a world. A pair $(\mathcal{M}, w)$ where $w$ is a world in $\mathcal{M}$ is called a *pointed epistemic model*, or *epistemic state*, while a pair $(\mathcal{M}, W')$, where $W' \subseteq W$ is a subset of worlds, is called a *multipointed epistemic model*.

An epistemic model is *finite* if its set of worlds $W$ is finite and for each world $w \in W$, $\lambda(w)$ is finite. In that case, we let $|\mathcal{M}|$ be the *size* of $\mathcal{M}$, defined as $|W| + \sum_{a \in Agt} |\preccurlyeq_a| + \sum_{w \in W} |\lambda(w)|$. From now on, all epistemic models are assumed to be finite.

**Event models** Dynamic Epistemic Logic also relies on *event models*. These models specify actions, the preconditions for their execution, their effects on the world, and how agents perceive their occurrence.

**Definition 2.2.** *An* event model $\mathcal{A} = (A, (\preccurlyeq_a^{\mathcal{A}})_{a \in Agt}, pre, post)$ *is a tuple where:*

- $A$ *is a set of possible* actions*,*
- $\preccurlyeq_a^{\mathcal{A}} \subseteq A \times A$ *is the* accessibility relation *for agent $a$,*
- $pre : A \to \mathbf{EL}$ *is a* precondition *function (where $\mathbf{EL}$ stands for epistemic logic), and*
- $post : A \times AP \to$ *Prop is a* postcondition *function (where Prop stands for set of propositions).*

An action $\alpha$ is *executable* in a world $w$ of an epistemic model $\mathcal{M}$ if its precondition $pre(\alpha)$ holds in $w$, i.e., $\mathcal{M}, w \models pre(\alpha)$. A set of actions $A' \subseteq A$ is *non-blocking* if $\bigvee_{\alpha \in A'} pre(\alpha) \equiv \top$, i.e., there is always at least one action in $A'$ that is executable. After executing an executable action $\alpha$ in a world $w$, proposition $p$ holds if its postcondition was satisfied before executing the action; thus, let us define $\lambda(w, \alpha) := \{p \in AP \mid \mathcal{M}, w \models post(\alpha, p)\}$ as the set of propositions holding after executing $\alpha$ in $w$. Since postconditions are propositional, one can define similarly $\lambda(\nu, \alpha)$ where $\nu \subseteq 2^{AP}$ is a valuation. A *pointed action model* is a pair $(\mathcal{A}, \alpha)$ where $\alpha$ represents the actual action.

Only finite action models will be considered, i.e., such that the set of actions $A$ is finite, and for every action $\alpha \in A$ there are only finitely many atomic propositions $p \in AP$ whose postcondition is not trivially false, i.e., such that $post(\alpha, p) \neq \bot$. We let $|\mathcal{A}|$ be the *size* of $\mathcal{A}$, defined as follows:

$$|\mathcal{A}| := |A| + \sum_{a \in Agt} |\preccurlyeq_a^{\mathcal{A}}| + \sum_{\alpha \in A} |pre(\alpha)| + \sum_{\alpha \in A, p \in AP} |post(\alpha, p)|$$

When working with variables $x$ over finite domains, one may write $x := d$ for the effect of setting $x$ to value $d$. This can again be encoded with atomic propositions $x_d$ and postconditions as defined above.

**Update product** After occurrence of an action $\alpha$ in a world $w$, agent $a$ considers it possible that action $\alpha'$ occurred in world $w'$, if in $w$ he considers $w'$ possible and $\alpha'$ is executable in $w'$. Hence, he considers action $\alpha'$ possible when action $\alpha$ is executed. This leads to the following definition of the product that models how to update an epistemic model when an action is executed [9].

**Definition 2.3 (Product [9]).** *Let $\mathcal{M} = (W, (\preccurlyeq)_{a \in Agt}, \lambda)$ be an epistemic model, and $\mathcal{A} = (A, (\preccurlyeq_a^{\mathcal{A}})_{a \in Agt}, pre, post)$ be an action model. The* product *of $\mathcal{M}$ and $\mathcal{A}$ is defined as $\mathcal{M} \otimes \mathcal{A} = (W', (\preccurlyeq_a^{\mathcal{A}})', \lambda')$ where:*

- $W' = \{(w, \alpha) \in W \times A \mid \mathcal{M}, w \models pre(\alpha)\}$,
- $(w, \alpha) \preccurlyeq'_a (w', \alpha')$ *if* $w \preccurlyeq_a w'$ *and* $\alpha \preccurlyeq_a^{\mathcal{A}} \alpha'$, *and*
- $\lambda'(w, \alpha) = \lambda(w, \alpha)$.

## 3. Reachability Results

As previously said, reachability goals in DEL games have been considered in two different settings. In the first case, two players (the Controller and the Environment) playing in turn are taken into account, and hence the set of actions is partitioned according to them. In this case, the problem is to decide whether the Controller has a strategy to ensure that some epistemic property holds. The controller synthesis problem is undecidable, in general. There are some special cases in which it becomes decidable. Precisely, when actions are:

- Public announcements, the problem is PSPACE-complete;
- Public actions, the problem is EXPTIME-complete.

The problem can be further investigated by letting the agents be active entities and chose their own actions. Then, one can consider the distributed strategy synthesis as the problem of deciding whether a coalition of agents has a common strategy to let some property hold. This problem is undecidable in general [10], but, as for the controller synthesis problem, there are cases in which some decidability results can be obtained, in particular when actions are:

- Public announcements, the problem is PSPACE-complete;
- Public actions, the problem is EXPTIME-complete.

Further details on the algorithms to prove the results above can be found in [2].

## 4. Concurrency results

In this section, concurrent executions of DEL games are considered and some decidability results are shown.

**Concurrent Actions** To define concurrent actions, atomic propositions are partitioned into *shared propositions* ($AP^s$) that all agents can modify, and *private* ones ($AP^p_a$, for the specific agent $a$). The set $A_a$ denotes the actions that an agent can play without modifying private propositions of others. A *joint action* is a tuple $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_N \rangle \in \prod_{a \in Agt} A_a$, and we let $\vec{\alpha}_b$ denote action $\alpha_b$, and it is available in $w$ when every individual action $\vec{\alpha}_b$ can be executed in $w$.

The formula $\mathsf{noconflict}(\vec{\alpha})(p)$ expresses that all individual actions of a joint action $\vec{\alpha}$ agree on their effect (if any) on proposition $p$, and in general:

$$\mathsf{noconflict}(\vec{\alpha}) := \bigwedge_{p \in AP} \mathsf{noconflict}(\vec{\alpha})(p). \tag{1}$$

Hence, one can say that a joint action $\vec{\alpha}$ is *non-conflicting in* $w$ if $\mathcal{M}, w \models \mathsf{noconflict}(\vec{\alpha})$. Otherwise $\vec{\alpha}$ is *conflicting in* $w$. In case of conflicting available joint action, a scheduler is supposed to select a maximal subset of consistent individual actions, by inhibiting the remaining ones, through a ghost mapping.

### Concurrent Update Product

**Definition 4.1 (Concurrent update product).** *The* concurrent update product *of an epistemic model* $\mathcal{M}$ *and an action model* $\mathcal{A}$ *is the Kripke model* $\mathcal{M} \boxplus \mathcal{A} = (W^{\boxplus}, (\approx^{\boxplus}_a)_{a \in Agt}, \lambda^{\boxplus})$*, where:*

- $W^{\boxplus} = \{(w, \vec{\beta}) \in W \times A^N \mid \vec{\beta} \in \mathtt{Max}(\vec{\alpha}, u)$*,* $\vec{\alpha}$ *available in* $w\}$*;*
- $(w, \vec{\beta}) \approx^{\boxplus}_a (u, \vec{\gamma})$ *if* $w \approx_a u$ *and* $\vec{\beta}_b \approx^{\mathcal{A}}_a \vec{\gamma}_b$ *for all* $b$*;*
- $p \in \lambda^{\boxplus}(u, \vec{\beta})$ *if* $(\mathcal{M}, u) \models \bigwedge_{a \in Agt(\vec{\beta}, p)} post(\beta_a, p)$*.*

*where:*

$$\mathtt{Max}(\vec{\alpha}, w) := \{\vec{\beta} \mid \vec{\beta} \preceq \vec{\alpha} \text{ and } \vec{\beta} \text{ is non-conflicting in } w \text{ and } \vec{\beta} \text{ is } \preceq\text{-maximal}\}$$

*The set of epistemic states that may result when a joint action* $\vec{\alpha}$ *is executed in* $(\mathcal{M}, w)$ *are:*

$$(\mathcal{M}, w) \boxplus \vec{\alpha} := \{(\mathcal{M} \boxplus \mathcal{A}, (w, \vec{\beta})) \mid \vec{\beta} \in \mathtt{Max}(\vec{\alpha}, w)\}.$$

*This set is a singleton when* $\vec{\alpha}$ *is non-conflicting in* $w$*.*

**Decidability Results** By exploiting the result of [6] saying that the model checking of $\mathsf{ATL}^*_{\mathsf{K}}$ on finite deterministic concurrent game structures is 2-Exptime-complete when actions are public, one can conclude that the same decidability results holds for model checking on concurrent DEL games, when actions are public, the ghost mapping is injective, and the scheduler is public.

## 5. Conclusions

This work shows Pspace-completeness both for the controller synthesis and the distributed synthesis problem when actions are public announcements, and Exptime-completeness when actions are public. This last decidability result also holds for model checking on concurrent DEL games. We plan to further investigate the connection of DEL with more powerful logics for strategic reasoning, such as Strategy Logic [11] under imperfect information [12, 13].

## Acknowledgments

This papers is based on my Phd Thesis [14] and the works [2, 7].

I would like to thank Aniello Murano, Bastien Maubert, Sophie Pinchinat, and Francois Schwarzentruber for their mentoring and support.

This work is partially supported by the PRIN project RIPER (No. 20203FFYLK).

## References

[1] H. Van Ditmarsch, W. van Der Hoek, B. Kooi, Dynamic epistemic logic, volume 337, Springer Science & Business Media, 2007.

[2] B. Maubert, A. Murano, S. Pinchinat, F. Schwarzentruber, S. Stranieri, Dynamic epistemic logic games with epistemic temporal goals, in: Proceedings of 24th European Conference on Artificial Intelligence ECAI 2020, IOS Press, 2020, pp. 155–162.

[3] F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin, Verification of broadcasting multi-agent systems against an epistemic strategy logic, in: C. Sierra (Ed.), Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, 2017, pp. 91–97.

[4] F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin, Verification of multi-agent systems with imperfect information and public actions, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, ACM, 2017, pp. 1268–1276.

[5] F. Belardinelli, S. Knight, A. Lomuscio, B. Maubert, A. Murano, S. Rubin, Reasoning about agents that may know other agents' strategies, in: Proc. of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, ijcai.org, 2021, pp. 1787–1793.

[6] F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin, Verification of multi-agent systems with public actions against strategy logic, Artif. Intell. 285 (2020) 103302.

[7] B. Maubert, S. Pinchinat, F. Schwarzentruber, S. Stranieri, Concurrent games in dynamic epistemic logic, in: Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, 2020, pp. 1877–1883.

[8] R. Fagin, J. Halpern, Y. Moses, M. Vardi, Reasoning about Knowledge, MIT Press, 1995.

[9] A. Baltag, L. S. Moss, S. Solecki, The logic of public announcements, common knowledge, and private suspicions, in: TARK'98, 1998, pp. 43–56.

[10] M. J. Coulombe, J. Lynch, Cooperating in video games? impossible! undecidability of team multiplayer games, in: FUN'18, 2018, pp. 14:1–14:16.

[11] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: On the model-checking problem, ACM Trans. Comput. Log. 15 (2014) 34:1–34:47.

[12] W. Jamroga, V. Malvone, A. Murano, Natural strategic ability, Artif. Intell. 277 (2019).

[13] R. Berthon, B. Maubert, A. Murano, S. Rubin, M. Y. Vardi, Strategy logic with imperfect information, ACM Trans. Comput. Log. 22 (2021) 5:1–5:51.

[14] S. Stranieri, Vehicular ad Hoc Networks: an algorithmic and a game-theoretic approach, Ph.D. thesis, Universitá di Napoli, Federico II, 2022.