Automata Techniques for Epistemic Protocol Synthesis

Guillaume Aucher, Bastien Maubert, Sophie Pinchinat

Automata Techniques for Epistemic Protocol Synthesis

Guillaume Aucher
Université de Rennes 1 - INRIA
Rennes, France
guillaume.aucher@irisa.fr

Bastien Maubert
Université de Rennes 1
Rennes, France
bastien.maubert@irisa.fr

Sophie Pinchinat
Université de Rennes 1
Rennes, France
sophie.pinchinat@irisa.fr

In this work we aim at applying automata techniques to problems studied in Dynamic Epistemic Logic, such as epistemic planning. To do so, we first remark that repeatedly executing *ad infinitum* a propositional event model from an initial epistemic model yields a relational structure that can be finitely represented with automata. This correspondence, together with recent results on *uniform strategies*, allows us to give an alternative decidability proof of the epistemic planning problem for propositional events, with as by-products accurate upper-bounds on its time complexity, and the possibility to synthesize a finite word automaton that describes the set of all solution plans. In fact, using automata techniques enables us to solve a much more general problem, that we introduce and call *epistemic protocol synthesis*.

1 Introduction

Automated planning, as defined and studied in [9], consists in computing a finite sequence of actions that takes some given system from its initial state to one of its designated "goal" states. The Dynamic Epistemic Logic (DEL) community has recently investigated a particular case of automated planning, called *epistemic planning* [7, 11, 1]. In DEL, epistemic models and event models can describe accurately how agents perceive the occurrence of events, and how their knowledge or beliefs evolve. Given initial epistemic states of the agents, a finite set of available events, and an epistemic objective, the epistemic planning problem consists in computing (if any) a finite sequence of available events whose occurrence results in a situation satisfying the objective property. While this problem is undecidable in general [7, 1], restricting to *propositional events* (those whose pre and postconditions are propositional) yields decidability [19].

In this paper, preliminary to our main results we bring a new piece to the merging of various frameworks for knowledge and time. Some connections between DEL and Epistemic Temporal Logics (ETL) are already known [10, 4, 2, 18]. We establish that structures generated by iterated execution of an event model from an epistemic model are regular structures, *i.e.* they can be finitely represented with automata, in case the event model is propositional. This allows us to reduce the epistemic planning problem for propositional events to the *uniform strategy problem*, as studied in [13, 14, 12]. The automata techniques developed for uniform strategies then provide an alternative proof of [19], with the additional advantage of bringing accurate upper-bounds on the time complexity of the problem, as well as an effective synthesis procedure to generate the recognizer of all solution plans. In fact, our approach allows us to solve a generalized problem in DEL, that we call *epistemic protocol synthesis problem*, and which is essentially the problem of synthesizing a protocol from an epistemic temporal specification; its semantics relies on the interplay between DEL and ETL. We then make use of the connections with regular structures and uniform strategies to solve this latter general problem.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 97–103, doi:10.4204/EPTCS.146.13

2 DEL models

For this paper we fix Ag, a finite set of *agents*, and AP always denotes a finite set of atomic propositions (which is not fixed). The epistemic language $\mathscr{L}^{EL}$ is simply the language of propositional logic extended with "knowledge" modalities, one for each agent. Intuitively, $K_i\varphi$ reads as "agent i knows φ". The syntax of $\mathscr{L}^{EL}$ is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi, \qquad (\text{where } p \in AP \text{ and } i \in Ag)$$

The semantics of $\mathscr{L}^{EL}$ is given in terms of epistemic models. Intuitively, a (pointed) epistemic model $(\mathscr{M},w)$ represents how the agents perceive the actual world w.

Definition 1 *An* epistemic model *is a tuple* $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$ *where* W *is a finite set of possible* worlds, $R_i \subseteq W \times W$ *is an* accessibility relation *on* W *for agent* $i \in Ag$, *and* $V : AP \to 2^W$ *is a* valuation function.

We write $w \in \mathscr{M}$ for $w \in W$, and we call $(\mathscr{M},w)$ a *pointed epistemic model*. Formally, given a pointed epistemic model $(\mathscr{M},w)$, we define the semantics of $\mathscr{L}^{EL}$ by induction on its formulas: $\mathscr{M},w \models p$ if $w \in V(p)$, $\mathscr{M},w \models \neg\varphi$ if it is not the case that $\mathscr{M},w \models \varphi$, $\mathscr{M},w \models \varphi \vee \psi$ if $\mathscr{M},w \models \varphi$ or $\mathscr{M},w \models \psi$, and $\mathscr{M},w \models K_i\varphi$ if for all w' such that wR_iw', $\mathscr{M},w' \models \varphi$.

Definition 2 *An* event model *is a tuple* $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},pre,post)$ *where* E *is finite set of* events, *for each* $i \in Ag$, $\mathsf{R}_i \subseteq \mathsf{E} \times \mathsf{E}$ *is an* accessibility relation *on* E *for agent* i, $pre : \mathsf{E} \to \mathscr{L}^{EL}$ *is a* precondition function *and* $post : \mathsf{E} \to AP \to \mathscr{L}^{EL}$ *is a* postcondition function.

We write $e \in \mathscr{E}$ for $e \in \mathsf{E}$, and call $(\mathscr{E},e)$ a *pointed event model*. For an event $e \in \mathscr{E}$, the precondition $\mathrm{pre}(e)$ and the postconditions $\mathrm{post}(e)(p)$ ($p \in AP$) are epistemic formulas. They respectively describe the set of worlds where event e may take place and the set of worlds where proposition p will hold after event e has occurred.

Definition 3 *A* proposition event model *is an event model whose preconditions and postconditions all lie in the propositional fragment of* $\mathscr{L}^{EL}$.

We now define the *update product* which, given an epistemic model $\mathscr{M}$ and an event model $\mathscr{E}$, builds the epistemic model $\mathscr{M} \otimes \mathscr{E}$ that represents the new epistemic situation after $\mathscr{E}$ has occurred in $\mathscr{M}$.

Definition 4 *Let* $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$ *be an epistemic model and* $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},pre,post)$ *be an event model. The* update product *of* $\mathscr{M}$ *and* $\mathscr{E}$ *is the epistemic model* $\mathscr{M} \otimes \mathscr{E} = (W^\otimes,\{R_i^\otimes\}_{i\in Ag},V^\otimes)$, *where* $W^\otimes = \{(w,e) \in W \times \mathsf{E} \mid \mathscr{M},w \models pre(e)\}$, $R_i^\otimes(w,e) = \{(w',e') \in W^\otimes \mid w' \in R_i(w)$ *and* $e' \in \mathsf{R}_i(e)\}$, *and* $V^\otimes(p) = \{(w,e) \in W^\otimes \mid \mathscr{M},w \models post(e)(p)\}$.

The update product of a pointed epistemic model $(\mathscr{M},w)$ with a pointed event model $(\mathscr{E},e)$ is $(\mathscr{M},w) \otimes (\mathscr{E},e) = (\mathscr{M} \otimes \mathscr{E},(w,e))$ if $\mathscr{M},w \models \mathrm{pre}(e)$, and it is undefined otherwise.

To finish with this section, we define the *size* of an epistemic model $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$, denoted by $|\mathscr{M}|$, as its number of edges: $|\mathscr{M}| = \sum_{i\in Ag}|R_i|$. The size of an event model $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},\mathrm{pre},\mathrm{post})$, that we note $|\mathscr{E}|$, is its number of edges plus the sizes of precondition and postcondition formulas: $|\mathscr{E}| = \sum_{i\in Ag}|\mathsf{R}_i| + \sum_{e\in\mathsf{E}}(|\mathrm{pre}(e)| + \sum_{p\in AP}|\mathrm{post}(e)(p)|)$.

3 Trees, forests and $\mathsf{CTL}^*\mathsf{K}_n$

A *tree alphabet* is a finite set of *directions* $\Upsilon = \{d_1, d_2 \ldots\}$. A Υ-*tree*, or *tree* for short when Υ is clear from the context, is a set of words $\tau \subseteq \Upsilon^+$ that is closed for nonempty prefixes, and for which there is a direction $r = \tau \cap \Upsilon$, called the *root*, such that for all $x \in \tau$, $x = r \cdot x'$ for some $x' \in \Upsilon^*$. A Υ-*forest*, or *forest* when Υ is understood, is defined likewise, except that it can have several roots. Alternatively a forest can be seen as a union of trees.

We classically allow nodes of trees and forests to carry additional information via labels: given a *labelling alphabet* Σ and a tree alphabet Υ, a Σ-*labelled* Υ-*tree*, or (Σ,Υ)-*tree* for short, is a pair $t = (\tau, \ell)$, where τ is a Υ-tree and $\ell : \tau \to \Sigma$ is a *labelling*. The notion of (Σ,Υ)-*forest* $\mathscr{U} = (u, \ell)$ is defined likewise. Note that we use forests to represent the universe (to be defined) in the semantics of $\mathsf{CTL}^*\mathsf{K}_n$, hence the notations $\mathscr{U}$ and u. Given a Υ-forest u and a node $x = d_1 \ldots d_n$ in the forest u, we define the tree u_x to which this node belongs as the "greatest" tree in the forest u that contains the node x: $u_x = \{y \in u \mid d_1 \preccurlyeq y\}$. Similarly, given a (Σ,Υ)-forest $\mathscr{U} = (u, \ell)$ and a node $x \in u$, $\mathscr{U}_x = (u_x, \ell_x)$, where u_x is as above and ℓ_x is the restriction of ℓ to the tree u_x.

The set of well-formed $\mathsf{CTL}^*\mathsf{K}_n$ formulas is given by the following grammar:

$$\begin{array}{lll} \text{State formulas:} & \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{A}\psi \mid K_i\varphi & (\text{where } p \in AP \text{ and } i \in Ag) \\ \text{Path formulas:} & \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi, & \end{array}$$

Let Υ be a finite set of directions, and let $\Sigma = 2^{AP}$ be the set of possible valuations. A $\mathsf{CTL}^*\mathsf{K}_n$ (state) formula is interpreted in a node of a (Σ,Υ)-tree, but the semantics is parameterized by, first, for each agent $i \in Ag$, a binary relation $\rightsquigarrow_i$ between finite words over Σ, and second, a forest of (Σ,Υ)-trees which we see as the *universe*. Preliminary to defining the semantics of $\mathsf{CTL}^*\mathsf{K}_n$, we let the *node word* of a node $x = d_1 d_2 \ldots d_n \in \tau$ be $w(x) = \ell(d_1)\ell(d_1 d_2)\ldots\ell(d_1 \ldots d_n) \in \Sigma^*$, made of the sequence of labels of all nodes from the root to this node. Now, given a family $\{\rightsquigarrow_i\}_{i\in Ag}$ of binary relations over Σ^*, a (Σ,Υ)-forest $\mathscr{U}$, two nodes $x, y \in \mathscr{U}$ and $i \in Ag$, we let $x \rightsquigarrow_i y$ denote that $w(x) \rightsquigarrow_i w(y)$.

A state formula of $\mathsf{CTL}^*\mathsf{K}_n$ is interpreted over a (Σ,Υ)-tree $t = (\tau, \ell)$ in a node $x \in \tau$, with an implicit universe $\mathscr{U}$ and relations $\{\rightsquigarrow_i\}_{i\in Ag}$, usually clear from the context: the notation $t, x \models \varphi$ means that φ holds at the node x of the labelled tree t. Because all inductive cases but the knowledge operators follow the classic semantics of CTL^* on trees, we only give the semantics for formulas of the form $K_i\varphi$:

$t, x \models K_i\varphi$ if for all $y \in \mathscr{U}$ such that $x \rightsquigarrow y$, $\mathscr{U}_y, y \models \varphi$ [1]

We shall use the notation $t \models \varphi$ for $t, r \models \varphi$, where r is the root of t.

Before stating the problems considered and our results, we establish in the next section a connection between DEL-generated models and regular structures, that allows us to apply automata techniques to planning problems in DEL.

4 DEL-generated models and regular structures

We first briefly recall some basic definitions and facts concerning finite state automata and transducers. A *deterministic word automaton* is a tuple $\mathscr{A} = (\Sigma, Q, \delta, q_t, F)$, where Σ is an *alphabet*, Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is a partial *transition function* and F is a set of *accepting* states. The *language* accepted by a word automaton $\mathscr{A}$ consists in the set of words accepted by $\mathscr{A}$, and it is classically written

[1] Recall that $\mathscr{U}_y$ is the biggest tree in $\mathscr{U}$ that contains y.

$\mathscr{L}(\mathscr{A})$. It is well known that the set of languages accepted by word automata is precisely the set of *regular* word languages. A *finite state synchronous transducer*, or *synchronous transducer* for short, is a finite word automaton with two tapes, that reads one letter from each tape at each transition. Formally, a synchronous transducer is a tuple $T = (\Sigma, Q, \Delta, q_\iota, F)$, where the components are as for word automata, except for the *transition relation* $\Delta \subseteq Q \times \Sigma \times \Sigma \times Q$. The (binary) relation recognized by a transducer T is denoted by $[T] \subseteq \Sigma^* \times \Sigma^*$. Synchronous transducers are known to recognize the set of *regular relations*, also called *synchronized rational relations* in the literature (see [8, 6, 3]). In the following, the size of a transducer T, written $|T|$, will denote the size of its transition relation: $|T| = |\Delta|$.

Definition 5 *A* relational structure *is a tuple $\mathscr{S} = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$ where D is the (possibly infinite)* domain *of $\mathscr{S}$, for each $i \in Ag$, $\rightsquigarrow_i \subseteq D \times D$ is a binary relation and $V : AP \to 2^D$ is a valuation function. V can alternatively be seen as a set of predicate interpretations for atomic propositions in AP.*

Definition 6 *A relational structure $\mathscr{S} = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$ is a* regular structure *over a finite alphabet Σ if its domain $D \subseteq \Sigma^*$ is a regular language over Σ, for each i, $\rightsquigarrow_i \subseteq \Sigma^* \times \Sigma^*$ is a regular relation and for each $p \in AP$, $V(p) \subseteq D$ is a regular language. Given deterministic word automata $\mathscr{A}_\mathscr{S}$ and $\mathscr{A}_p$ ($p \in AP$), as well as transducers T_i for $i \in Ag$, we say that $(\mathscr{A}_\mathscr{S}, \{T_i\}_{i\in Ag}, \{\mathscr{A}_p\}_{p\in AP})$ is a* representation *of $\mathscr{S}$ if $\mathscr{L}(\mathscr{A}_\mathscr{S}) = D$, for each $i \in Ag$, $[T_i] = \rightsquigarrow_i$ and for each $p \in AP$, $\mathscr{L}(\mathscr{A}_p) = V(p)$.*

Definition 7 *For an epistemic model $\mathscr{M} = (W, \{R_i\}_{i\in Ag}, V)$ and an event model $\mathscr{E} = (\mathsf{E}, \{\mathsf{R}_i\}_{i\in Ag}, pre, post)$, we define the family of epistemic models $\{\mathscr{M}\mathscr{E}^n\}_{n\geq 0}$ by letting $\mathscr{M}\mathscr{E}^0 = \mathscr{M}$ and $\mathscr{M}\mathscr{E}^{n+1} = \mathscr{M}\mathscr{E}^n \otimes \mathscr{E}$. Letting, for each n, $\mathscr{M}\mathscr{E}^n = (W^n, \{R_i^n\}_{i\in Ag}, V^n)$, we define the relational structure generated by $\mathscr{M}$ and $\mathscr{E}$ as $\mathscr{M}\mathscr{E}^* = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$, where:*

- $D = \bigcup_{n\geq 0} W^n$,
- *$h \rightsquigarrow_i h'$ if there is some n such that $h, h' \in \mathscr{M}\mathscr{E}^n$ and $h R_i^n h'$, and*
- $V(p) = \bigcup_{n\geq 0} V^n(p)$.

Proposition 1 *If $\mathscr{M}$ is an epistemic model and $\mathscr{E}$ is a propositional event model, then $\mathscr{M}\mathscr{E}^*$ is a regular structure, and it admits a representation of size $2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$.*

Proof Let $\mathscr{M} = (W, R, V)$ be an epistemic model, let $\mathscr{E} = (\mathsf{E}, \mathsf{R}, \mathrm{pre}, \mathrm{post})$ be a propositional event model, and let $\mathscr{M}\mathscr{E}^* = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V_D)$.

Define the word automaton $\mathscr{A}_D = (\Sigma, Q, \delta, q_\iota, F)$, where $\Sigma = W \cup \mathsf{E}$, $F = \{q_\nu \mid \nu \subseteq AP\}$ and $Q = F \uplus \{q_\iota\}$. For a world $w \in W$, we define its *valuation* as $\nu(w) := \{p \in AP \mid w \in V(p)\}$. We now define δ, which is the following partial transition function:

$$\begin{array}{ll} \forall w \in W, \forall e \in \mathsf{E}, & \\ \delta(q_\iota, w) = q_{\nu(w)} & \delta(q_\iota, e) \text{ is undefined,} \\ \delta(q_\nu, w) \text{ is undefined} & \delta(q_\nu, e) = \begin{cases} q_{\nu'}, \text{ with } \nu' = \{p \mid \nu \models \mathrm{post}(e)(p)\} & \text{if } \nu \models \mathrm{pre}(e) \\ \text{undefined} & \text{otherwise.} \end{cases} \end{array}$$

It is not hard to see that $\mathscr{L}(\mathscr{A}_D) = D$, hence D is a regular language. Also, $\mathscr{A}_D$ has $2^{|AP|} + 1$ states, and each state has at most $|\mathscr{M}| + |\mathscr{E}|$ outgoing transitions, so that $|\mathscr{A}_D| = 2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)$.

Concerning valuations, take some $p \in AP$. Let $\mathscr{A}_p = (\Sigma, Q, \delta, q_\iota, F_p)$, where $F_p = \{q_\nu \mid p \in \nu\}$. Clearly, $\mathscr{L}(\mathscr{A}_p) = V_D(p)$, hence $V_D(p)$ is a regular language, and $|\mathscr{A}_p| = |\mathscr{A}_D|$.

For the relations, let $i \in Ag$ and consider the one-state synchronous transducer $T_i = (\Sigma, Q', \Delta_i, q_\iota, F')$, where $Q' = \{q\}$, $q_\iota = q$, $F' = \{q\}$, and $\Delta_i = \{(q, w, w', q) \mid w R_i w'\} \cup \{(q, e, e', q) \mid e \mathsf{R}_i e'\}$. It is easy to see

that $\rightsquigarrow_i = [T_i] \cap D \times D$. Since $[T_i]$ is a regular relation and D is a regular language, $\rightsquigarrow_i$ is a regular relation recognized by $T_i' = T_D \circ T_i \circ T_D$, where T_D is a synchronous transducer that recognizes the identity relation over D (easily obtained from $\mathscr{A}_D$). This transducer is of size $|T_i'| = |T_D|^2 \cdot |T_i| = 2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$. Finally, $\mathscr{M}\mathscr{E}^*$ is a regular structure that accepts $(\mathscr{A}_D, \{T_i'\}_{i \in Ag}, \{\mathscr{A}_p\}_{p \in AP})$ as a regular representation of size $2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$. One can check that this is also an upper bound on the time needed to compute this representation.

□

5 Epistemic protocol synthesis

We first consider the problem of epistemic planning [7, 11] studied in the Dynamic Epistemic Logic community. Note that our formulation slightly differs from the classic one as we consider a unique event model, but both problems can easily be proved inter-reducible in linear time.

Definition 8 (Epistemic planning problem) *Given a pointed epistemic model $(\mathscr{M}_\iota, w_\iota)$, an event model $\mathscr{E}$, a set of events $\mathsf{E} \subseteq \mathscr{E}$ and a goal formula $\varphi \in \mathscr{L}^{EL}$, decide if there exists a finite series of events $e_1 \dots e_n$ in E such that $(\mathscr{M}_\iota, w_\iota) \otimes (\mathscr{E}, e_1) \otimes \dots \otimes (\mathscr{E}, e_n) \models \varphi$. The* propositional epistemic planning problem *is the restriction of the epistemic planning problem to propositional event models.*

The epistemic planning problem is undecidable [7, 1]. However, [7] proved that the problem is decidable in the case of one agent and equivalence accessibility relations in epistemic and event models. More recently, [1] and [19] proved independently that the one agent problem is also decidable for K45 accessibility relations. [19] also proved that restricting to propositional event models yields decidability of the epistemic planning problem, even for several agents and arbitrary accessibility relations.

Theorem 2 ([19]) *The propositional epistemic planning problem is decidable.*

Proposition 1 allows us to establish an alternative proof of this result, with two side-benefits. First, using automata techniques, our decision procedure can synthesize as a by-product a finite word automaton that generates exactly the (possibly infinite) set of all solution plans. Second, we obtain accurate upper-bounds on the time complexity.

For an instance $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$ of the epistemic planning problem, we define its size as the sum of its components' sizes, plus the number of atomic propositions: $|\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi| = |\mathscr{M}| + |\mathscr{E}| + |\mathsf{E}| + |\varphi| + |AP|$.

Theorem 3 *The propositional epistemic planning problem is in $k+1$-*EXPTIME *for formulas of nesting depth k. Moreover, it is possible to build in the same time a finite word automaton $\mathscr{P}$ such that $\mathscr{L}(\mathscr{P})$ is the set of all solution plans.*

Proof sketch Let $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$ be an instance of the problem. By Proposition 1 we obtain an exponential size automatic representation of the forest $\mathscr{M}\mathscr{E}^*$: the set of possible histories, as well as their valuations, are represented by a finite automaton $\mathscr{A}$, and the epistemic relations are given by finite state transducers. Because the epistemic relations are rational, we can use the powerset construction presented in [13] in the context of uniform strategies [13, 14, 12]. Indeed, this construction easily generalizes to the case of n relations, and even though in [13] it is defined on game arenas it can, in our context, be adapted to regular structures. Letting k be the maximal nesting depth of knowledge operators in φ, this construction yields an automaton $\widehat{\mathscr{A}}$ of size k-exponential in the size of $\mathscr{A}$, hence $(k+1)$-exponential in $|\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi|$, that still represents $\mathscr{M}\mathscr{E}^*$, and in which φ can be evaluated positionally. Keeping only transitions labelled by events in E, and choosing for accepting states those that verify φ, we obtain the

automaton $\mathscr{P}$ that recognizes the set of solution plans. Furthermore, solving the epistemic planning problem amounts to solving the nonemptiness problem for $\mathscr{L}(\mathscr{P})$; this can be done in time linear in the size of $\mathscr{P}$, which is $k+1$-exponential in the size of the input $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$. □

In fact, the correspondence between the DEL framework and automatic structures established in Proposition 1 allows us to solve a much more general problem than epistemic planning.

We generalize the notion of epistemic planning in three directions. First, we no longer consider finite sequences of actions but infinite ones. As a consequence, we need not stick to reachability objectives as in planning (where the aim is to reach a state of the world that verifies some formula), and we therefore allow for any epistemic temporal formula as objective, which is the second generalization. Finally, we no longer look for a single series of events, but we try to synthesize a *protocol*, *i.e.* a set of plans.

Definition 9 *Given an epistemic model $\mathscr{M}$ and an event model $\mathscr{E}$, an* epistemic protocol *is a forest $P \subseteq \mathscr{M}\mathscr{E}^*$; it is* rooted *if it is a tree.*

Definition 10 (Epistemic protocol synthesis problem) *Given an initial pointed epistemic model $(\mathscr{M}, w)$, a propositional event model $\mathscr{E}$ and a $\mathsf{CTL}^*\mathsf{K}_n$ formula φ, letting $\mathscr{U} = \mathscr{M}\mathscr{E}^*$ be the universe, decide if there is an epistemic protocol $P \subseteq \mathscr{U}$ rooted in w such that $P \models \varphi$, and synthesize such a protocol if any.*

Again making use of Proposition 1, the epistemic protocol synthesis problem can be reduced to synthesizing a uniform strategy in a game arena with regular relations between plays. This can be solved with the powerset construction from [13] and classic automata techniques for solving games with CTL^* winning condition. We finally obtain the following result.

Theorem 4 *The epistemic protocol synthesis problem is decidable. If the nesting depth of the goal formulas is bounded by k, then the problem is in* $\max(2, k+1)$-EXPTIME.

6 Discussion

We have described a connection between DEL-generated models and regular structures, which enabled us to resort to a combination of mature automata techniques and more recent ones developed for the study of uniform strategies, in order to solve planning problems in the framework of DEL. We believe that this is but a first step in applying classic automata techniques developed for temporal logics to the study of dynamic epistemic logic. As witnessed by classic works on automata-based program synthesis (see for example [15, 17]), automata techniques are well suited to tackle problems such as synthesizing plans, protocols, strategies or programs, and we believe that it should also be the case in the DEL framework; in addition the complexity of solving classic automata problems such as nonemptiness has been extensively studied, and this may help to settle the complexity of problems in DEL, such as the epistemic planning problem.

As for future work, we would like to investigate the optimality of the upper-bounds that we obtained on the time complexity of the epistemic planning problem for propositional event models, as well as for our notion of epistemic protocol synthesis. Another direction for future research concerns the latter problem: a next step would be to apply techniques from control theory and quantified μ-calculus [16] to synthesize *maximal permissive* epistemic protocols. In general such objects only exist for safety objectives, but recently a weaker notion of *permissive strategy* has been studied in the context of parity games [5]. A strategy is permissive if it contains the behaviours of all memoryless strategies, and such strategies always exist in parity games. Similar notions may be introduced for protocols with epistemic temporal objectives to capture concepts of "sufficiently permissive" protocols.

References

[1] Guillaume Aucher & Thomas Bolander (2013): *Undecidability in Epistemic Planning*. In: *IJCAI*. Available at http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6903.

[2] Guillaume Aucher & Andreas Herzig (2011): *Exploring the power of converse events*. In: *Dynamic formal epistemology*, Springer, pp. 51–74, doi:10.1007/978-94-007-0074-1_4.

[3] Pablo Barceló, Diego Figueira & Leonid Libkin (2013): *Graph Logics with Rational Relations*. *Logical Methods in Computer Science* 9(3). Available at http://dx.doi.org/10.2168/LMCS-9(3:1)2013, http://arxiv.org/abs/1304.4150.

[4] Johan van Benthem, Jelle Gerbrandy, Tomohiro Hoshi & Eric Pacuit (2009): *Merging frameworks for interaction*. *Journal of Philosophical Logic* 38(5), pp. 491–526, doi:10.1007/s10992-008-9099-x.

[5] Julien Bernet, David Janin & Igor Walukiewicz (2002): *Permissive strategies: from parity games to safety games*. *ITA* 36(3), pp. 261–275. Available at http://dx.doi.org/10.1051/ita:2002013.

[6] Jean Berstel (1979): *Transductions and context-free languages*. 4, Teubner Stuttgart, doi:10.1007/978-3-663-09367-1.

[7] Thomas Bolander & Mikkel Birkegaard Andersen (2011): *Epistemic planning for single and multi-agent systems*. *Journal of Applied Non-Classical Logics* 21(1), pp. 9–34. Available at http://dx.doi.org/10.3166/jancl.21.9-34.

[8] C.C. Elgot & J.E. Mezei (1965): *On relations defined by generalized finite automata*. *IBM Journal of Research and Development* 9(1), pp. 47–68, doi:10.1147/rd.91.0047.

[9] Malik Ghallab, Dana S. Nau & Paolo Traverso (2004): *Automated planning - theory and practice*. Elsevier.

[10] Tomohiro Hoshi & Audrey Yap (2009): *Dynamic epistemic logic with branching temporal structures*. *Synthese* 169(2), pp. 259–281. Available at http://dx.doi.org/10.1007/s11229-009-9552-6.

[11] Benedikt Löwe, Eric Pacuit & Andreas Witzel (2011): *DEL Planning and Some Tractable Cases*. In Hans P. van Ditmarsch, Jérôme Lang & Shier Ju, editors: *LORI, Lecture Notes in Computer Science* 6953, Springer, pp. 179–192. Available at http://dx.doi.org/10.1007/978-3-642-24130-7_13.

[12] Bastien Maubert & Sophie Pinchinat (2013): *Jumping Automata for Uniform Strategies*. In: *FSTTCS'13*, pp. 287–298, doi:10.4230/LIPIcs.FSTTCS.2013.287.

[13] Bastien Maubert & Sophie Pinchinat (2014): *A General Notion of Uniform Strategies*. *International Game Theory Review* 16(01), doi:10.1142/S0219198914400040.

[14] Bastien Maubert, Sophie Pinchinat & Laura Bozzelli (2013): *The Complexity of Synthesizing Uniform Strategies*. In Fabio Mogavero, Aniello Murano & Moshe Y. Vardi, editors: *SR*, *EPTCS* 112, pp. 115–122. Available at http://dx.doi.org/10.4204/EPTCS.112.17.

[15] A. Pnueli & R. Rosner (1989): *On the Synthesis of an Asynchronous Reactive Module*. In: *Proc. 16th Int. Coll. on Automata, Languages and Programming, ICALP'89, Stresa, Italy, LNCS 372*, Springer-Verlag, pp. 652–671, doi:10.1007/BFb0035790.

[16] Stéphane Riedweg & Sophie Pinchinat (2003): *Quantified Mu-Calculus for Control Synthesis*. In Branislav Rovan & Peter Vojtás, editors: *MFCS*, *Lecture Notes in Computer Science* 2747, Springer, pp. 642–651. Available at http://dx.doi.org/10.1007/978-3-540-45138-9_58.

[17] Wolfgang Thomas (1995): *On the Synthesis of Strategies in Infinite Games*. In: *STACS*, pp. 1–13. Available at http://dx.doi.org/10.1007/3-540-59042-0_57.

[18] Yanjing Wang & Guillaume Aucher (2013): *An Alternative Axiomatization of DEL and Its Applications*. In: *IJCAI*. Available at http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6802.

[19] Quan Yu, Ximing Wen & Yongmei Liu (2013): *Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions*. In: *IJCAI*, pp. 1183–1190. Available at http://ijcai.org/papers13/Papers/IJCAI13-178.pdf.