Automata Techniques for Epistemic Protocol Synthesis in DEL

Guillaume Aucher, Bastien Maubert and Sophie Pinchinat

Highlights of Logic, Games and Automata

September 3rd, 2014
Two important approaches to add dynamics to Epistemic Logics:

**Epistemic Temporal Logics**

A model usually consists of:
- **Dynamics**: A finite transition system
- **Epistemics**: Observational equivalences on states.

**Dynamic Epistemic Logics**

Much finer way to describe the events and how they are perceived.
- **Epistemics**: Epistemic models and event models to represent possible worlds/events, and how they are perceived,
- **Dynamics**: Update product between epistemic and event models
Strategizing/planning

In the context of ETL:

- Many decidability/complexity results
- Rely on the fact that the set of histories is regular
  - Powerset constructions
  - Tree automata techniques

In the context of DEL:

- Very little results
- Because the set of histories is not regular in general?

In this work:

- Identify a condition for DEL-generated structures to be regular
- Use automata techniques to tackle planning problems in DEL
1. Dynamic Epistemic Logic (DEL)

2. From DEL to regular structures

3. Epistemic planning and epistemic protocol synthesis
An example

Epistemic model $\mathcal{M}$:

- The coin is on heads
- Alice believes that it is so
- Bob believes it's on tails
An example

Epistemic model $\mathcal{M}$:
- The coin is on heads
- Alice believes that it is so
- Bob believes it’s on tails

Event model $\mathcal{E}$:
- Alice turns the coin over
- Bob is not sure whether she really does it or not
An example

Epistemic model $\mathcal{M}$:
- The coin is on heads
- Alice believes that it is so
- Bob believes it’s on tails

Event model $\mathcal{E}$:
- Alice turns the coin over
- Bob is not sure whether she really does it or not
An example

Epistemic model $M$:
- The coin is on heads
- Alice believes that it is so
- Bob believes it’s on tails

Event model $E$:
- Alice turns the coin over
- Bob is not sure whether she really does it or not
An example

Epistemic model $\mathcal{M}$:
- The coin is on heads
- Alice believes that it is so
- Bob believes it’s on tails

Event model $\mathcal{E}$:
- Alice turns the coin over
- Bob is not sure whether she really does it or not
An example

Epistemic model $\mathcal{M}$

Event model $\mathcal{E}$
Dynamic Epistemic Logic (DEL)
From DEL to regular structures
Epistemic planning and epistemic protocol synthesis

An example

Epistemic model $\mathcal{M}$

Event model $\mathcal{E}$

The resulting epistemic model $\mathcal{M} \otimes \mathcal{E}$
DEL-generated structures

Structure generated from $\mathcal{M}$ and $\mathcal{E}$

- $\mathcal{ME}^n = \mathcal{M} \otimes \mathcal{E} \otimes \ldots \otimes \mathcal{E}$
- $\mathcal{ME}^* = \bigcup_{n \geq 0} \mathcal{ME}^n = (H, \{\sim_i\}_{i \in Ag}, V)$

An element of $\mathcal{ME}^*$ is a history $we_1 \ldots e_n$
$we_1 \ldots e_n \sim_i w'e'_1 \ldots e'_n$ if $w R_i w'$ and $e_k R_i e'_k$ for all $k$.

Propositional event models

Pre and post-conditions are propositional.
Plan

1. Dynamic Epistemic Logic (DEL)
2. From DEL to regular structures
3. Epistemic planning and epistemic protocol synthesis
Regular relations

A binary relation over words is regular/automatic if it is recognized by a synchronous two-tape automaton.

Example: $\sim_i$ in $\mathcal{ME}^*$ (synchronous perfect recall)

$w/w'$ if $w \mathcal{R}_i w'$

Recognized by:

$q_0$

$e/e'$ if $e \mathcal{R}_i e'$
From DEL to automata

Theorem: from DEL to automata

For every epistemic model $\mathcal{M}$ and propositional event model $\mathcal{E}$, $\mathcal{M}\mathcal{E}^*$ is a regular structure, and we can build recognizers.

Let $\mathcal{M}\mathcal{E}^* = (H, \{\sim_i\}_{i \in Ag}, V)$.

We prove that:

- $H$ is a regular language
- each $\sim_i$ is a regular relation
- each $V(p)$ is a regular sub-language of $H$
From DEL to automata

Theorem: from DEL to automata

For every epistemic model $\mathcal{M}$ and propositional event model $\mathcal{E}$, $\mathcal{ME}^*$ is a regular structure, and we can build recognizers.

Let $\mathcal{ME}^* = (H, \{\sim_i\}_{i \in Ag}, V)$.

We prove that:

- $H$ is a regular language
- each $\sim_i$ is a regular relation ✓
- each $V(p)$ is a regular sub-language of $H$
From DEL to automata

**Theorem: from DEL to automata**

For every epistemic model $\mathcal{M}$ and propositional event model $\mathcal{E}$, $\mathcal{M}\mathcal{E}^*$ is a regular structure, and we can build recognizers.

Let $\mathcal{M}\mathcal{E}^* = (H, \{\sim_i\}_{i \in Ag}, V)$.

**We prove that:**

- $H$ is a regular language ✓
- each $\sim_i$ is a regular relation ✓
- each $V(p)$ is a regular sub-language of $H$
From DEL to automata

Theorem: from DEL to automata

For every epistemic model $\mathcal{M}$ and propositional event model $\mathcal{E}$, $\mathcal{M}\mathcal{E}^*$ is a regular structure, and we can build recognizers.

Let $\mathcal{M}\mathcal{E}^* = (H, \{\sim_i\}_{i \in Ag}, V)$.

We prove that:

- $H$ is a regular language ✓
- each $\sim_i$ is a regular relation ✓
- each $V(p)$ is a regular sub-language of $H$ ✓
Plan

1. Dynamic Epistemic Logic (DEL)
2. From DEL to regular structures
3. Epistemic planning and epistemic protocol synthesis
Epistemic planning

The epistemic planning problem (EPP)

Input:
- a pointed initial epistemic model \((\mathcal{M}, w)\)
- an event model \(\mathcal{E}\)
- a goal formula \(\varphi \in \mathcal{L}^{EL}\)

Output:
- Is there \(e_1 \ldots e_n\) s.t. \((\mathcal{M}, w) \otimes (\mathcal{E}, e_1) \otimes \ldots \otimes (\mathcal{E}, e_n) \models \varphi\)?
The epistemic planning problem (EPP)

**Input:**
- a pointed initial epistemic model \((\mathcal{M}, w)\)
- an event model \(\mathcal{E}\)
- a goal formula \(\varphi \in \mathcal{L}^{EL}\)

**Output:**
- Is there \(e_1 \ldots e_n\) s.t. \(\mathcal{M}\mathcal{E}^*, \forall e_1 \ldots e_n \models \varphi\)?
The propositional epistemic planning problem (propositional EPP)

Input:
- a pointed initial epistemic model \((M, w)\)
- a propositional event model \(\mathcal{E}\)
- a goal formula \(\varphi \in \mathcal{L}^{EL}\)

Output:
- Is there \(e_1 \ldots e_n\) s.t. \(M\mathcal{E}^*, we_1 \ldots e_n \models \varphi\)?

Theorem [Yu et al. 2013]

The propositional epistemic planning problem is decidable.

[Yu et al. 2013] prove that a finite search tree is sufficient.
The following problem is decidable [Bozzelli, M., Pinchinat 2013]:

**Input:**
- A labelled game graph
- $\varphi \in \text{CTL}^* K_n$
- Regular relations $\{\sim_i\}_{i \leq n}$

**Output:**
- Is there a strategy for Player 1 that verifies $\varphi$?

**Given an instance $(\mathcal{M}, \mathcal{E}, \varphi)$ of propositional EPP:**
- Build the recognizers for the regular structure $\mathcal{M}\mathcal{E}^*$
- See the automaton for $H$ as a one-player game graph
- Reduce to the above problem with formula $\mathbf{EF}\varphi$
Benefits of this proof

- Provides better upper bounds on the complexity.
- Builds an automaton that generates all the solution plans.
- Our approach allows us to solve a much more general problem.
Epistemic protocol synthesis in DEL

Epistemic planning
- finite sequence of events
- reach epistemic objective

Theorem
The propositional epistemic protocol synthesis problem is decidable.
Same techniques and same upper bounds as for epistemic planning.
Epistemic protocol synthesis in DEL

Epistemic planning
- finite sequence of events
- reach epistemic objective

Epistemic protocol synthesis
- infinite tree of events
- $\text{CTL}^* K_n$ specification
Epistemic protocol synthesis in DEL

Epistemic planning
- finite sequence of events
- reach epistemic objective

Epistemic protocol synthesis
- infinite tree of events
- CTL*\(K_n\) specification

Theorem
The propositional epistemic protocol synthesis problem is decidable.

Same techniques and same upper bounds as for epistemic planning.
Conclusion

- Connected DEL-generated structures and regular structures
- This bridge allows us to apply existing automata techniques
- Alternative decidability proof for Propositional EPP
- Side results:
  - Improved complexity upper-bounds
  - Synthesize an automaton that recognizes the solution plans
- Same techniques apply to solve the generalized problem of Epistemic protocol synthesis.

Thank you!
Semantics of $\mathcal{O}$ and $\mathcal{O}'$

Blue arrows: $\mathcal{O}$

$\mathcal{O}$: Strict quantifier

$\mathcal{O}'$: Full quantifier
Semantics of $\bowtie$ and $\preceq$

Blue arrows: $\rightsquigarrow$

$\bowtie :$ Strict quantifier

$\preceq :$ Full quantifier
Semantics of $\sqsubseteq$ and $\sqsubseteq$

Blue arrows: $\sim$

$\sqsubseteq$ : Strict quantifier

$\sqsubseteq$ : Full quantifier
Semantics of $\exists$ and $\forall$

Blue arrows: $\sim$

$\exists : \text{Strict quantifier}$

$\forall : \text{Full quantifier}$
Semantics of $\sqsubseteq$ and $\sqsubseteq$:

Blue arrows: $\sim$

$\sqsubseteq$: Strict quantifier

$\sqsubseteq$: Full quantifier