Automatic structures, chain Monadic Second-Order logic, Epistemic planning synthesis.

Gaëtan Douéneau-Tabot
Sophie Pinchinat
François Schwarzentruber

ENS Paris-Saclay, Université Paris-Saclay
IRISA, Université de Rennes
IRISA, ENS Rennes

FMAI 2019, Rennes
Outline

1 Motivation

2 Model checking on automatic structures

3 Regular automatic trees

4 Application: Epistemic planning
Motivations

- Uniform strategies in imperfect information games (Maubert 2014)
- Models of epistemic temporal logic, interpreted systems (Halpern-Vardi 1989)
- Epistemic planning (Pinchinat et al. 2018)

Trade-off between

\[ \{ \text{the class of infinite structures} \quad \text{the considered logic} \} \]
Motivation
Model checking on automatic structures
Regular automatic trees
Application: Epistemic planning

Contribution

DECIDABLE

UNDECIDABLE
1 Motivation

2 Model checking on automatic structures
   • Structures
   • Automatic presentations
   • Logics

3 Regular automatic trees

4 Application: Epistemic planning
Outline

1 Motivation

2 Model checking on automatic structures
   - Structures
     - Automatic presentations
     - Logics

3 Regular automatic trees

4 Application: Epistemic planning
Motivation
Model checking on automatic structures
Regular automatic trees
Application: Epistemic planning

Structures
Automatic presentations
Logics

(Relational) structures
Interpretations for logics like FO and MSO

$S = \langle D, R_1 \ldots R_p \rangle$
- Domain $D \neq \emptyset$
- Relations $R_i \subseteq D^{r_i}$

Example
$\langle \mathbb{N}, \leq \rangle$

Definition (Tree structures)
$T = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle$
- $(n$-ary tree)$
- $D \subseteq \{1, \ldots, n\}^*$ prefix-closed
- $r(\varepsilon)$ only
- $S_j(u, v.j)$ whenever $u.j \in D$
- $R_1, \ldots, R_p$ additional relations on $D$
  such as the “at equal level” binary relation.
Full binary trees

\[ T_2 = \langle \{1, 2\}^*, S_1, S_2 \rangle \]

\[ S_i(u, u.i) \text{ for every } u \in \{1, 2\}^* \]

\[ T_2^{el} = \langle \{1, 2\}^*, S_1, S_2, el \rangle \text{ with relation “at equal level”: } \]

\[ el(u, v) \text{ whenever } |u| = |v| \]
Motivation

Model checking on automatic structures

Regular automatic trees

Application: Epistemic planning

Outline

1. Motivation

2. Model checking on automatic structures
   - Structures
   - Automatic presentations
   - Logics

3. Regular automatic trees

4. Application: Epistemic planning
An example

Take structure $\langle \mathbb{N}, \leq \rangle$.

- Encode $n \in \mathbb{N}$ by $11\ldots1 = \text{enc}(n)$
- Encoding of relation $\leq$: the pair $(1^2, 1^3)$ is represented by
  \[ 1^2 \otimes 1^3 := (\frac{1}{1})(\frac{1}{1})(\square_1) \quad \text{(convolution)} \]

Use two-tape automaton to accept input $1^n \otimes 1^m$ whenever $n \leq m$.

\[ \mathcal{A}_{\leq}: \quad \text{start} \rightarrow q_0 \rightarrow q_1 \]

- $1^2 \otimes 1^3 \in \mathcal{L}(\mathcal{A}_{\leq})$
- $1^3 \otimes 1^2 = (\frac{1}{1})(\frac{1}{1})(\square_1) \notin \mathcal{L}(\mathcal{A}_{\leq})$
\( \mathcal{T}_2 \) and \( \mathcal{T}^\text{el}_2 \) are automatic

Recall \( \mathcal{T}_2 = \langle \{1, 2\}^*, S_1, S_2 \rangle \) and \( \mathcal{T}^\text{el}_2 = \langle \{1, 2\}^*, S_1, S_2, \text{el} \rangle \)

- Encode nodes with their address \( u \in \{1, 2\}^* \)
- \( S_1(u, v) \) iff \( v = u.1 \)

\[ A_{S_1} : \]
\[
\begin{array}{c}
(\frac{1}{1}), (\frac{2}{2}) \\
\text{start} \rightarrow q_0 \rightarrow q_1 \\
(\frac{1}{1}) \rightarrow q_2 \\
(\frac{2}{1}), (\frac{2}{2})
\end{array}
\]

Similarly \( A_{S_2} \ldots \)

- \( \text{el}(u, v) \) iff \( |u| = |v| \)

\[ A_{\text{el}} : \]
\[
\begin{array}{c}
(\frac{1}{1}), (\frac{1}{2}), (\frac{2}{1}), (\frac{2}{2}) \\
\text{start} \rightarrow q_0 \rightarrow q_1 \\
(\frac{1}{1}), (\frac{1}{2}), (\frac{2}{1}), (\frac{2}{2})
\end{array}
\]
Automatic presentations

Definition

An **automatic presentation** of structure $S = \langle D, R_1 \ldots R_p \rangle$ is $(\mathcal{A}_D, \mathcal{A}_1, \ldots, \mathcal{A}_p)$ a tuple of (finite-state) automata with

- a bijective **encoding function** $enc : D \rightarrow \mathcal{L}(\mathcal{A}_D)$
- $\mathcal{L}(\mathcal{A}_i) = enc(R_i)$:

  \[
  u_1 \otimes \cdots \otimes u_{r_i} \in \mathcal{L}(\mathcal{A}_i)
  \]

  iff

  \[
  u_j \in \mathcal{L}(\mathcal{A}_D) \text{ and } (enc^{-1}(u_1), \ldots, enc^{-1}(u_{r_i})) \in R_i
  \]

$(\mathcal{A}_D, \mathcal{A}_1, \ldots, \mathcal{A}_p) \leadsto$ structure $S(\mathcal{A}_D, \mathcal{A}_1, \ldots, \mathcal{A}_p)$
Outline

1 Motivation

2 Model checking on automatic structures
   - Structures
   - Automatic presentations
   - Logics

3 Regular automatic trees

4 Application: Epistemic planning
Motivation
Model checking on automatic structures
Regular automatic trees
Application: Epistemic planning

Structures
Automatic presentations
Logics

Logics FO and MSO

- $V_1 = \{x, x_1, x_2, \ldots\}$ set of first-order variables
- $V_2 = \{X, X_1, \ldots, Y, \ldots\}$ set of second-order variables

$MSO \ni \varphi ::= R_i(x_1 \ldots x_{r_i}) | \neg \varphi | (\varphi \land \varphi) | \exists x \varphi | x \in X | \exists X \varphi$

Gaëtan Douéneau-Tabot Sophie Pinchinat François Schwarzentruber
Model checking on automatic structures against FO

\[ \text{MC-FO} \begin{cases} \text{Input : } (A_D, A_1, \ldots, A_p) \text{ an A.S., FO-formula } \varphi. \\
\text{Output : } S(A_D, A_1, \ldots, A_p) \models \varphi? \end{cases} \]

Theorem ([Khoussainov and Nerode 1995, Grädel and Bumensath 2000, Rubin 2008])

**MC-FO is decidable.**

Main ingredient: for \( \varphi(x_1 \ldots x_n) \in FO \), structure \( S \),

\[ \varphi^S := \{(d_1 \ldots d_n) \in D^n \mid S, [x_i \mapsto d_i] \models \varphi[x_1 \ldots x_n]\} \]

Proposition

**Effective construction of automaton** \( A_\varphi \) s.t. \( L(A_\varphi) = \text{enc}(\varphi^S) \).
Bottom-up construction of $A_\varphi$: intuitive example

Take $\varphi(x) := \exists z R_2(z, x) \land \neg p(x)$

$\varphi^S := \{ d \in D \mid S, [x \mapsto d] \models \varphi[x] \}$

1. Project $A_{R_2(x, z)}$ given by the automatic presentation, onto the first component and get $A_{\exists z R_2(x, z)}$;

2. Take $A_p(x) \in A$, complement it and get $A_c^p(x)$ and compute $A_D \cap A_c^p(x)$ to get the automaton $A_{\neg p(x)}$;

3. Compute $A_{\exists z R_2(x, z)} \cap A_{\neg p(x)}$ to get automaton $A_{\exists z R_2(z, x) \land \neg p(x)}$.

$L(A_\varphi(x)) = \{ \text{enc}(d) \mid d \in \varphi^S \}$. 
Model checking on automatic structures against MSO

\[ \text{MSO } \exists \varphi ::= R_i(x_1 \ldots x_r) | \neg \varphi | (\varphi \land \varphi) | \exists x \varphi | x \in X | \exists X \varphi \]

\[ \text{MC-MSO } \begin{cases} \text{Input} : (A_D, A_1, \ldots, A_p), \text{MSO-formula } \varphi \\ \text{Output} : S(A_D, A_1, \ldots, A_p) \models \varphi ? \end{cases} \]

Proposition (Barany 2007)

MC-MSO is decidable if \( L(A_D) \subseteq \{1\}^* \) (unary alphabet, e.g., \((\mathbb{N}, \leq))\)

Theorem (Rabin 1969)

MSO-theory of \( T_2 = \langle \{1, 2\}^*, S_1, S_2 \rangle \) is decidable (full binary tree)

Theorem (Thomas 1990)

MSO-theory of \( T_2^{el} = \langle \{1, 2\}^*, S_1, S_2, el \rangle \) is undecidable (equal level)
Variants of MSO over trees
Different restrictions of second-order quantifications

(a) MSO quantification over any subset
(b) path-MSO quantification over any path in a tree
(c) cMSO quantification over any chain in a tree
Outline

1. Motivation
2. Model checking on automatic structures
3. Regular automatic trees
   • Chain MSO
4. Application: Epistemic planning
Regular automatic trees (RegAutTrees)

A particular encoding!

**Definition (RegAutTree)**

Tree $\mathcal{T} = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle$ is regular automatic if

- its language of node addresses is regular
- $\text{enc}_{\text{address}} : D \rightarrow \{1, \ldots, n\}^*$ allows an automatic presentation of $\mathcal{T}$

\[
\langle \mathcal{A}_D, \mathcal{A}_r, (\mathcal{A}_{S_i})_{1 \leq i \leq n}, (\mathcal{A}_{R_i})_{1 \leq i \leq p} \rangle
\]

The canonical representation of $\mathcal{T}$.

**Theorem**

*RegAutTrees is a proper subclass of automatic trees.*
Regular automatic trees $\subsetneq$ Automatic trees

Set of addresses
\[ \{1^m2^k \mid 0 \leq k \leq m\} \]

\[ \leadsto \text{not a regular automatic tree} \]

Write $\text{bin}(n)$ for the binary encoding of $n$ with least significant digit first.

\[ \text{enc}(1^m2^k) := \text{bin}(m) \otimes \text{bin}(k) \]

\[ \text{enc}(112) = \text{enc}(1^22^1) \]
\[ = \text{bin}(2) \otimes \text{bin}(1) \]
\[ = (0)(1) \]

$\text{enc}(D), \text{enc}(S_1), \text{enc}(S_2)$ and $\text{enc}(el)$ are regular languages.
Properties of RegAutTrees

In \( \mathcal{T} = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle \), define

- Generalized successor relation \( S := \bigcup_{i=1}^{n} S_i \)
- Reflexive and transitive closure \( S^* \) of \( S \)
- Binary relation \( \preceq \) for ‘deeper in the tree’
- Binary relation \( \text{el} \) for “at equal level”
- Equality relation \( = \)

**Lemma**

\( \mathcal{T} = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle \in \text{RegAutTrees} \) implies
\( \langle D, r, S_1, \ldots, S_n, R_1 \ldots R_p, S^*, \preceq, \text{el}, = \rangle \in \text{RegAutTrees}. \)

FO is decidable on RegAutTrees.
Chains in trees

Tree $\mathcal{T} = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle$.

Definition

$C \subseteq D$ is a chain if it is totally ordered with respect to $S^*$:

for all $u, v \in C$, either $S^*(u, v)$ or $S^*(u, v)$. 

Outline

1 Motivation

2 Model checking on automatic structures

3 Regular automatic trees
   • Chain MSO

4 Application: Epistemic planning
Logic cMSO

cMSO $\exists \varphi ::= R_i(x_1 \ldots x_r) | \neg \varphi | (\varphi \land \varphi) | \exists x \varphi | x \in X | \exists X \varphi$

interpreted over $T = \langle D, r, S_1, \ldots, S_n, R_1, \ldots, R_p \rangle$ + assignment $\sigma$

$T, \sigma \models \exists X \varphi$ iff there exists a chain $C \in Chains(T)$ s.t.
$T, \sigma[X \mapsto C] \models \varphi.$

Example ("$X$ is a maximal path starting at node $x_0$")

$x_0 \in X \land$
$\forall x \{ x \in X \rightarrow [(\exists y S(x, y) \rightarrow \exists y (S(x, y) \land y \in X)) \land \neg S(x, x_0)] \}$

Corollary

cMSO subsumes path-MSO.
Motivation

Model checking on automatic structures
- Regular automatic trees
- Application: Epistemic planning

Chain MSO

Model checking on RAT against cMSO

**Theorem**

\[ \text{MC-RATcMSO} \left\{ \begin{array}{l}
\text{Input : } T \in \text{RAT (canon. pres.), } \varphi \in \text{cMSO} \\
\text{Output : } T \models \varphi?
\end{array} \right. \]

**Theorem**

\[ \text{MC-RATcMSO is decidable.} \]

The proof uses automata constructions, inspired from (Thomas 1997).
Corollaries of MC-RATcMSO decidability

- Since over a unary alphabet every set is a chain:

**Corollary (Barany 2007)**

*The MSO-theory of automatic structures on a unary alphabet is decidable.*
Corollaries of MC-RAT\textsuperscript{cMSO} decidability

- Since over a unary alphabet every set is a chain:

  \textbf{Corollary (Barany 2007)}

  \textit{The MSO-theory of automatic structures on a unary alphabet is decidable.}

- Since in RAT, relations $S^*$, $\preccurlyeq$, $el$, $=$ are regular:

  \textbf{Corollary}

  \textit{The cMSO$[r, S_1, \ldots, S_n, R_1, \ldots, R_p, S^*, \preccurlyeq, el, =]$-theory of RATs is decidable.}
Corollaries of MC-RATcMSO decidability

- Since over a unary alphabet every set is a chain:
  
  **Corollary (Barany 2007)**

  *The MSO-theory of automatic structures on a unary alphabet is decidable.*

- Since in RAT, relations $S^*$, $\triangleleft$, el, $=$ are regular:

  **Corollary**

  *The cMSO[$r$, $S_1$, ..., $S_n$, $R_1$, ..., $R_p$, $S^*$, $\triangleleft$, el, $=$]-theory of RATs is decidable.*

- Since one can express in cMSO that a chain is a path:

  **Corollary**

  *The path-MSO-theory of RATs is decidable.*
Outline

1 Motivation

2 Model checking on automatic structures

3 Regular automatic trees

4 Application: Epistemic planning
   - Dynamic Epistemic Logic
   - Logics of knowledge and time
Outline

1 Motivation

2 Model checking on automatic structures

3 Regular automatic trees

4 Application: Epistemic planning
   - Dynamic Epistemic Logic
   - Logics of knowledge and time
Dynamic Epistemic Logic: DEL presentation \((\mathcal{M}, w), \mathcal{E}\)
(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(\mathcal{M}\) an epistemic model

\[w : \{p\} \quad a \quad b \quad u : \emptyset \]

\(\mathcal{E}\) an event model

\[e : \text{pre} : p \quad \text{post} : p \leftarrow \bot \]

\[f : \text{pre} : \top \quad \text{post} : b \]

\[\mathcal{M} \otimes \mathcal{E} \]

\(\mathcal{M} \otimes \mathcal{E}, w \not\models K\neg p\), but

\(\mathcal{M} \otimes \mathcal{E}, w \models <E, e> K\neg p\)
Dynamic Epistemic Logic: DEL presentation \((M, w), E\)

(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(M\) an epistemic model

\(w : \{p\} \quad u : \emptyset\)

\(E\) an event model

\(e : \begin{align*}
\text{pre} & : p \\
\text{post} & : p \leftarrow \perp
\end{align*}\)

\(f : \begin{align*}
\text{pre} & : \top \\
\text{post} & :
\end{align*}\)

\(M \otimes E\)
Dynamic Epistemic Logic: DEL presentation \((M, w), \mathcal{E}\)  
(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(M\) an epistemic model

\(w : \{p\}\)

\(u : \emptyset\)

\(a\)

\(b\)

\(E\) an event model

\(e : \begin{align*}
\text{pre} & : p \\
\text{post} & : p \leftarrow \bot
\end{align*}\)

\(f : \begin{align*}
\text{pre} & : \top \\
\text{post} & :
\end{align*}\)

\(a\)

\(b\)

\(M \otimes \mathcal{E}\)

\(we : \emptyset\)

\(wf : \{p\}\)

\(uf : \emptyset\)

\(a\)

\(b\)
Dynamic Epistemic Logic: DEL presentation \((M, w), E\)
(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(M\) an epistemic model

\(w : \{p\}\)

\(u : \emptyset\)

\(a\)

\(b\)

\(f\)

\(M \otimes E\)

\(we : \emptyset\)

\(wf : \{p\}\)

\(uf : \emptyset\)

\(a\)

\(b\)

Gaëtan Douéneau-Tabot Sophie Pinchinat François Schwarzentruber
Dynamic Epistemic Logic: DEL presentation \((\mathcal{M}, w), \mathcal{E}\)

(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(\mathcal{M}\) an epistemic model

\(\mathcal{M} \otimes \mathcal{E}\)

\(\mathcal{M}, w \not\models K_a \neg p\)
Dynamic Epistemic Logic: DEL presentation \((M, w), E\) 
(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(M\) an epistemic model

\[\mathcal{M} \otimes \mathcal{E}\]

\(\mathcal{M}, w \not\models K_a \neg p\), but \(\mathcal{M}, w \models \langle \mathcal{E}, e \rangle K_a \neg p\)
Dynamic Epistemic Logic: DEL presentation \((\mathcal{M}, w), \mathcal{E}\)
(Baltag et al. 1998, van Ditmarsch et al. 2007)

\(\mathcal{M}\) an epistemic model

\(\mathcal{M} \otimes \mathcal{E}\)

\(\mathcal{M}, w \not\models K_a \neg p\), but \(\mathcal{M}, w \models < \mathcal{E}, e > K_a \neg p\)

\(\mathcal{M} \otimes \mathcal{E}, we \models K_a \neg p\)
Epistemic planning

**Epistemic planning**

\[
\begin{align*}
\textbf{Input:} & \quad \text{DEL presentation } (M, w), \mathcal{E}, \text{ epistemic formula } \psi \\
\textbf{Output:} & \quad \text{Is there } e_1, \ldots, e_n \text{ in } \mathcal{E} \text{ s.t. } M \mathcal{E}^n, we_1 \ldots e_n \models \psi? \\
\end{align*}
\]

**UNDECIDABLE** by (Bolander et al. 2011, Yu et al. 2013)

Automatic structures, chain Monadic Second-Order logic, Epistemic planning.
Propositional DEL presentations

Events have **propositional preconditions**

\[
\begin{align*}
    e : & \quad \text{pre} : \text{formula} \in \text{Bool}(AP) \\
    & \quad \text{post} : p \leftarrow \bot
\end{align*}
\]

\[
\begin{align*}
    f : & \quad \text{pre} : \text{formula} \in \text{Bool}(AP) \\
    & \quad \text{post} :
\end{align*}
\]

**Theorem (Maubert et al. 2014)**

*Propositional DEL structures* \( \in \text{RegAutTrees} \).
Epistemic planning for PDEL specifications

**Corollary (Maubert et al. 2014, AiML2018)**

Epistemic planning is decidable for propositional DEL presentations.

- On the PDEL structure, model check the property $\exists x \varphi_G(x)$.
- Automaton $A_{\varphi(x)}$ provides all solution plans, since the encoding of nodes in the PDEL structure is the sequence of events that has led to that node, i.e. the plan.

**Corollary**

The set of solution plans is a regular language regular.

- One can exploit $A_{\varphi_G(x)}$ to decide other properties, e.g. if there are infinitely many plans.
Outline

1 Motivation

2 Model checking on automatic structures

3 Regular automatic trees

4 Application: Epistemic planning
   - Dynamic Epistemic Logic
   - Logics of knowledge and time
Motivation
Model checking on automatic structures
Regular automatic trees
Application: Epistemic planning

Dynamic Epistemic Logic
Logics of knowledge and time

Gaëtan Douéneau-Tabot Sophie Pinchinat François Schwarzentruber
Automatic structures, chain Monadic Second-Order logic, Epistemic planning
Logics of knowledge and time (Halpern-Vardi 1989)

The models are particular trees $\mathcal{T} = \langle D, r, S_1, \ldots, S_n, K_1 \ldots K_m, (p)_{p \in AP} \rangle$

- time progress is given by $S := \bigcup_{i=1}^{n} S_i$ (temporal logics, CTL, etc.)
- atomic propositions in $AP = \{p, q, \ldots\}$ are unary relations
- knowledge modalities $K_1 \ldots K_m$ are binary relations.

Write $cMSOK$ for $cMSO[r, S_1, \ldots, S_n, (p)_{p \in AP}, K_1 \ldots K_m]$

Corollary

Model checking on RATs against any of these logics is decidable.