Model Checking Complex Systems against ITL Specifications with Regular Expressions

Alberto Molinari
joint work with Laura Bozzelli, Angelo Montanari, and Adriano Peron

Jan 20, 2017

University of Udine, Department of Mathematics, Computer Science, and Physics (DIMA)

HS (state-based) semantics and model checking

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models p$ iff $p \in \bigcap_{w \in \text{states}(\rho)} \mu(w)$, for any letter $p \in \mathcal{AP}$ (homogeneity assumption) [4];

HS (state-based) semantics and model checking

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models p$ iff $p \in \bigcap_{w \in \text{states}(\rho)} \mu(w)$, for any letter $p \in \mathcal{AP}$ (homogeneity assumption) [4];
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle \text{A} \rangle \psi$ iff there is a track ρ' s.t. $\text{lst}(\rho) = \text{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle \text{B} \rangle \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle \text{E} \rangle \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{\text{A}} \rangle$, $\langle \overline{\text{B}} \rangle$, and $\langle \overline{\text{E}} \rangle$ are similar

Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

HS (state-based) semantics and model checking

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models p$ iff $p \in \mu(\text{fst}(\rho), \text{lst}(\rho))$, for any letter $p \in \mathcal{AP}$ (endpoint-based labeling) [1, 2];
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle A \rangle \psi$ iff there is a track ρ' s.t. $\text{lst}(\rho) = \text{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle B \rangle \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle E \rangle \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ are similar

Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

HS (state-based) semantics and model checking

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models p$ other definitions?

- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle A \rangle \psi$ iff there is a track ρ' s.t. $\text{lst}(\rho) = \text{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle B \rangle \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle E \rangle \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ are similar

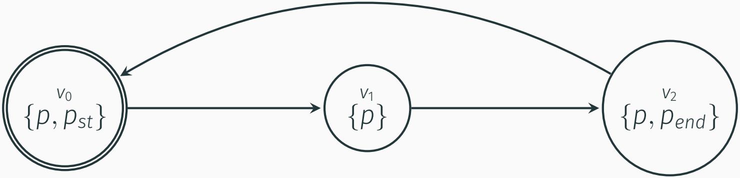
Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

Example—Printer

Adapted from [3].

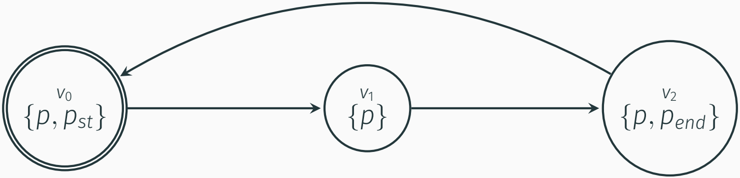


Imagine we want to label the process of printing a single sheet of paper with p.

- Under homogeneity,
  $v_0 v_1 v_2$ labeled by $p \Rightarrow v_0 v_1$ and $v_1 v_2$ labeled by p

Example—Printer

Adapted from [3].



Imagine we want to label the process of printing a single sheet of paper with p.

- Under endpoint-based labeling, assuming $p \in \mu(v_0, v_2)$, then $(v_0 v_1 v_2)^n$ are all labeled by p

(Usual) regular expressions

$$r ::= \varepsilon \mid a \mid r \cup r \mid r \cdot r \mid r^*$$

for $a \in \mathcal{A}$.

Examples:

- $r_1 = a \cdot (b \cup c)^* \cdot b$
  - $abb, acb, abccbb, \ldots \in \mathcal{L}(r_1)$

(Usual) regular expressions

$$r ::= \varepsilon \mid a \mid r \cup r \mid r \cdot r \mid r^*$$

for $a \in \mathcal{A}$.

Examples:

- $r_1 = a \cdot (b \cup c)^* \cdot b$
  - $abb, acb, abccbb, \ldots \in \mathcal{L}(r_1)$
- $r_2 = \Big((a \cdot b) \cup (a \cdot c)\Big)^*$
  - $\varepsilon, ab, ac, acabac, \ldots \in \mathcal{L}(r_2)$

(Usual) regular expressions

$$r ::= \varepsilon \mid a \mid r \cup r \mid r \cdot r \mid r^*$$

for $a \in \mathcal{A}$.

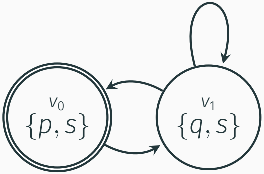
Examples:

- $r_1 = a \cdot (b \cup c)^* \cdot b$
    - $abb, acb, abccbb, \ldots \in \mathcal{L}(r_1)$
- $r_2 = \Big((a \cdot b) \cup (a \cdot c)\Big)^*$
    - $\varepsilon, ab, ac, acabac, \ldots \in \mathcal{L}(r_2)$
- $r_3 = \varepsilon \cdot (a \cup c)^*$
    - $\varepsilon, a, ca, aac, \ldots \in \mathcal{L}(r_3)$

Our regular expressions

$$r ::= \varepsilon \mid \phi \mid r \cup r \mid r \cdot r \mid r^*$$

where ϕ is a Boolean (propositional) formula over $\mathcal{AP}$.



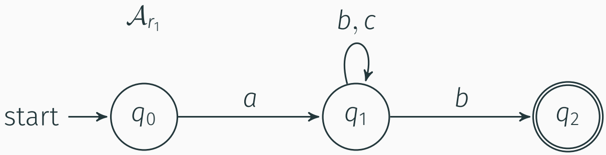
- $\rho = v_0 v_1 v_0 v_1 v_1$
- $\mu(\rho) = \{p, s\}\{q, s\}\{p, s\}\{q, s\}\{q, s\}$
- $\rho' = v_0 v_1 v_1 v_1 v_0$
- $\mu(\rho') = \{p, s\}\{q, s\}\{q, s\}\{q, s\}\{p, s\}$

Examples:

- $r_1 = (\mathsf{p} \wedge \mathsf{s}) \cdot \mathsf{s}^* \cdot (\mathsf{p} \wedge \mathsf{s})$
  - $\mu(\rho) \notin \mathcal{L}(r_1)$, but $\mu(\rho') \in \mathcal{L}(r_1)$
- $r_2 = (\neg \mathsf{p})^*$
  - $\mu(\rho) \notin \mathcal{L}(r_2)$, and $\mu(\rho) \notin \mathcal{L}(r_2)$

Nondeterministic finite automata (NFA)

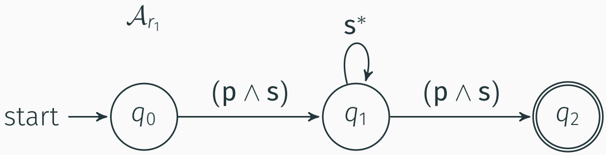
$$r_1 = a \cdot (b \cup c)^* \cdot b$$



- $abb, acb, abccbb, \ldots \in \mathcal{L}(r_1) = \mathcal{L}(\mathcal{A}_{r_1})$

Nondeterministic finite automata (NFA)

$$r_1 = (\mathsf{p} \wedge \mathsf{s}) \cdot \mathsf{s}^* \cdot (\mathsf{p} \wedge \mathsf{s})$$



- $\mu(\rho') = \{p, s\}\{q, s\}\{q, s\}\{q, s\}\{p, s\} \in \mathcal{L}(r_1) = \mathcal{L}(\mathcal{A}_{r_1})$

HS semantics with regular expressions

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models r$ iff $\mu(\rho) \in \mathcal{L}(r)$;
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle A \rangle \psi$ iff there is a track ρ' s.t. $\mathrm{lst}(\rho) = \mathrm{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle B \rangle \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle E \rangle \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ are similar

Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

HS semantics with regular expressions

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models r$ iff $\mu(\rho) \in \mathcal{L}(r)$;

- To force homogeneity, all regular expressions in the formula:

$$p \cdot (p)^*$$

- for endpoint-based labeling, regular expressions in the formula:

$$\bigcup_{(i,j)\in I} (q_i \cdot \top^* \cdot q_j)$$

for some $I \subseteq \{1, \ldots, |W|\}^2$, where $q_i \in \mathcal{AP}$ labels only $w_i \in W$.

for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

HS semantics with regular expressions

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models r$ iff $\mu(\rho) \in \mathcal{L}(r)$;
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle A \rangle\, \psi$ iff there is a track ρ' s.t. $\mathrm{lst}(\rho) = \mathrm{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle B \rangle\, \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle E \rangle\, \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ are similar

Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

HS semantics with regular expressions

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{K}, \rho \models r$ iff $\mu(\rho) \in \mathcal{L}(r)$;
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle A \rangle \psi$ iff there is a track ρ' s.t. $\mathrm{lst}(\rho) = \mathrm{fst}(\rho')$ and $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle B \rangle \psi$ iff there is a prefix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- $\mathcal{K}, \rho \models \langle E \rangle \psi$ iff there is a suffix ρ' of ρ s.t. $\mathcal{K}, \rho' \models \psi$;
- the semantic clauses for $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ are similar

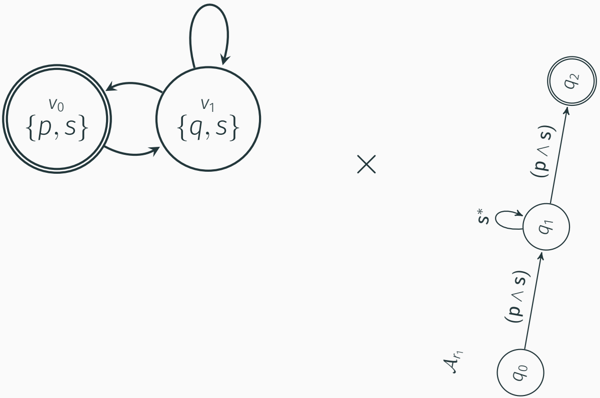
Model Checking

$\mathcal{K} \models \psi \iff$ for all *initial* tracks ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many tracks!

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

Idea: for a regular expression r



Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);

Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

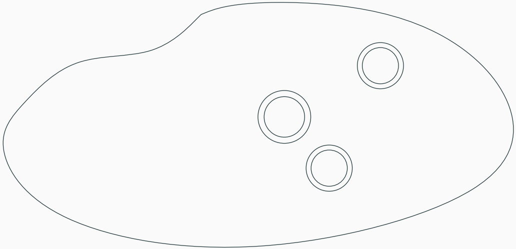
- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle \, \psi'$: we add a sink final state reachable from the “old” final states;

Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle \, \psi'$: we add a sink final state reachable from the "old" final states;

$\mathcal{A}_{\psi'}$:

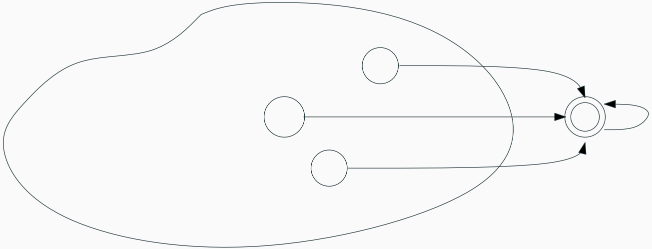


Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle\, \psi'$: we add a sink final state reachable from the "old" final states;

$\mathcal{A}_\psi$:



Decidability of MC for HS + regular expressions

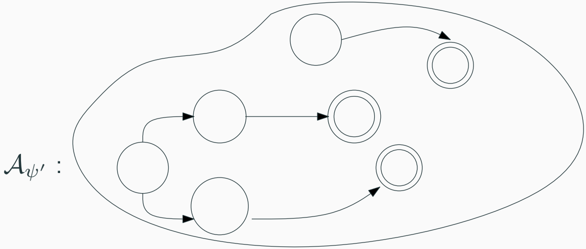
Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle\, \psi'$: we add a sink final state reachable from the “old” final states;
- for $\psi = \langle \overline{\mathrm{B}} \rangle\, \psi'$: we mark as final all states backward reachable from the “old” final states.

Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle\, \psi'$: we add a sink final state reachable from the "old" final states;
- for $\psi = \langle \overline{\mathrm{B}} \rangle\, \psi'$: we mark as final all states backward reachable from the "old" final states.

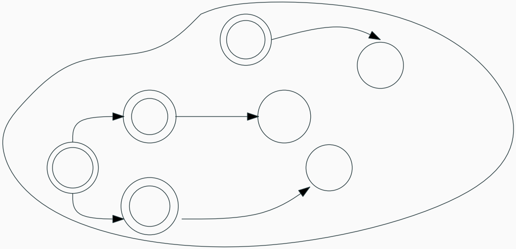


Decidability of MC for HS + regular expressions

Given $\mathcal{K}$ and an HS formula φ over $\mathcal{AP}$, we build an NFA over $\mathcal{K}$ accepting the set of tracks ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \wedge \psi_2$, $\psi = \psi_1 \vee \psi_2$: we do the usual constructions;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathrm{B} \rangle \, \psi'$: we add a sink final state reachable from the "old" final states;
- for $\psi = \langle \overline{\mathrm{B}} \rangle \, \psi'$: we mark as final all states backward reachable from the "old" final states.

$\mathcal{A}_\psi$:



The $A\overline{A}B\overline{B}$ fragment + regular expressions

We want to show that formulas of $A\overline{A}B\overline{B}$ + regular expressions can be checked by using polynomial working space.

To start with, we prove the following theorem, which is a building block of the **PSPACE**-model checking algorithm for $A\overline{A}B\overline{B}$.

Theorem

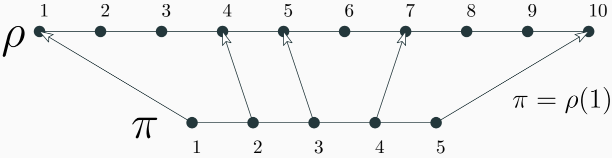
Let ρ be a track of $\mathcal{K}$ and φ be an $A\overline{A}B\overline{B}$ formula with RE's $r_1, \ldots, r_u$ such that

$$\mathcal{K}, \rho \models \varphi.$$

Then, there exists a track π of $\mathcal{K}$ such that

$$\mathcal{K}, \pi \models \varphi \quad \textit{and} \quad |\pi| \leq |W| \cdot (|\varphi| + 1) \cdot 2^{2 \sum_{\ell=1}^{u} |r_\ell|}.$$

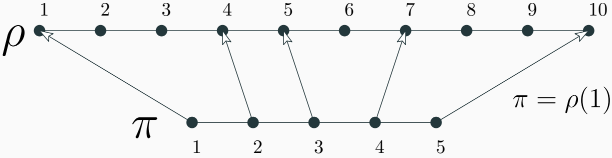
Small model for $A\overline{A}B\overline{B}$



$\pi = \rho(1)\rho(4)\rho(5)\rho(7)\rho(10)$

We want to guarantee that: for all π-positions j, with corresponding ρ-positions i_j, and for all $s = 1, \ldots, u$, $\mathcal{A}^s(\mu(\pi^j)) = \mathcal{A}^s(\mu(\rho^{i_j}))$

Small model for A$\overline{\text{A}}$B$\overline{\text{B}}$



$\pi = \rho(1)\rho(4)\rho(5)\rho(7)\rho(10)$

We want to guarantee that: for all π-positions j, with corresponding ρ-positions i_j, and for all $s = 1, \ldots, u$, $\mathcal{A}^s(\mu(\pi^j)) = \mathcal{A}^s(\mu(\rho^{i_j}))$

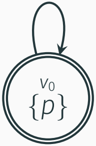
Theorem (Exponential small-model for A$\overline{\text{A}}$B$\overline{\text{B}}$)

Let ρ be a track of $\mathcal{K}$ and φ be an A$\overline{\text{A}}$B$\overline{\text{B}}$ formula with RE's $r_1, \ldots, r_u$ such that $\mathcal{K}, \rho \models \varphi$.
Then, there exists a track π of $\mathcal{K}$, induced by ρ, such that

$$\mathcal{K}, \pi \models \varphi \quad \text{and} \quad |\pi| \leq |W| \cdot (|\varphi| + 1) \cdot 2^{2\sum_{\ell=1}^{u} |r_\ell|}.$$

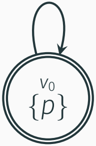
Small model for $A\overline{A}B\overline{B}$

The small model is “strict”:

- Let pr_i be the i-th smallest prime.
  It is well-known that $pr_i \in O(i \log i)$.
- Let $\mathcal{K} =$ 

Small model for A$\overline{\text{A}}$B$\overline{\text{B}}$

The small model is "strict":

- Let pr_i be the i-th smallest prime.
  It is well-known that $pr_i \in O(i \log i)$.
- Let $\mathcal{K} =$ 

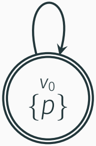
- Let us fix some $n \in \mathbb{N}$. The shortest track satisfying

$$\psi = \bigwedge_{i=1}^{n} (p^{pr_i})^*$$

  is $\rho = v_0^{pr_1 \cdots pr_n}$.

Small model for $A\bar{A}B\bar{B}$

The small model is “strict”:

- Let pr_i be the i-th smallest prime.
  It is well-known that $pr_i \in O(i \log i)$.
- Let $\mathcal{K} =$ 

- Let us fix some $n \in \mathbb{N}$. The shortest track satisfying

$$\psi = \bigwedge_{i=1}^{n} (p^{pr_i})^*$$

  is $\rho = v_0^{pr_1 \cdots pr_n}$.
- The length of ψ is $O(n \cdot pr_n) = O(n^2 \log n)$,
  but the length of ρ is $pr_1 \cdots pr_n \geq 2^n$.

A PSPACE MC algorithm for $A\overline{A}B\overline{B}$—Trials

- The algorithm can consider only tracks having length bounded by the exponential small model
- However,

A PSPACE MC algorithm for $\mathsf{A\overline{A}B\overline{B}}$—Trials

- The algorithm can consider only tracks having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples $(G, D(\psi), w)$ summarizing tracks, where
  - $G \subseteq \mathrm{Subf}_{\langle B \rangle}(\psi)$ contains the subformulas that hold on some prefix
  - $D(\psi)$ is the configuration of the DFAs after reading the track,
  - and w is the last state of the track

Lemma

For all formulas ψ of $\mathsf{B\overline{B}}$, and for all tracks ρ, ρ' of $\mathcal{K}$, if ρ and ρ' are summarized by the same triple $(G, D(\psi), w)$, then

$$\mathcal{K}, \rho \models \psi \iff \mathcal{K}, \rho' \models \psi.$$

A PSPACE MC algorithm for $A\overline{A}B\overline{B}$—Trials

- The algorithm cannot explicitly store the DFAs for the regular expressions occurring in $\psi \Rightarrow$ just store the current states of the computations of the DFAs and calculates on-the-fly the successor states

A PSPACE MC algorithm for $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{B}}$

Algorithm 1 Check($\mathcal{K}, \psi, w, G, D(\psi)$)

```
if ψ = r then                                          ◁ r is a regular expression
    if the current state of the DFA for r in advance(D(ψ), μ(w)) is final then
        return ⊤
    else
        return ⊥
else if ψ = ¬ψ′ or ψ = ψ1 ∧ ψ2 then
    Recursively
else if ψ = ⟨B⟩ ψ′ then
    return ψ′ ∈ G
else if ψ = ⟨B̄⟩ ψ′ then
    for all b ∈ {1, ..., |W| · (2|ψ′| + 1) · 2^(2 Σ_{ℓ=1}^{u} |r_ℓ|) − 1}, all (G′, D(ψ)′, w′) do
        if Reach(K, ψ′, (G, D(ψ), w), (G′, D(ψ)′, w′), b) and Check(K, ψ′, w′, G′, D(ψ)′) then
            return ⊤
    return ⊥
```

A PSPACE MC algorithm for $\mathsf{A\overline{A}B\overline{B}}$

Algorithm 2 Check($\mathcal{K}, \psi, w, G, D(\psi)$)

if $\psi = r$ **then** ◁ *r is a regular expression*
  if the current state of the DFA for r in advance($D(\psi), \mu(w)$) is final **then**
    return $\top$
  else
    return $\bot$
else if $\psi = \neg\psi'$ or $\psi = \psi_1 \wedge \psi_2$ **then**
  Recursively
else if $\psi = \langle \mathsf{B} \rangle\, \psi'$ **then**
  return $\psi' \in G$
else if $\psi = \langle \overline{\mathsf{B}} \rangle\, \psi'$ **then**
  for all $b \in \{1, \ldots, |W| \cdot (2|\psi'| + 1) \cdot 2^{2^{\sum_{\ell=1}^{u} |r_\ell|}} - 1\}$, all $(G', D(\psi)', w')$ **do**
    if Reach($\mathcal{K}, \psi', (G, D(\psi), w), (G', D(\psi)', w'), b$) and Check($\mathcal{K}, \psi', w', G', D(\psi)'$) **then**
      return $\top$
  return $\bot$

$\mathsf{A\overline{A}B\overline{B}}$ is PSPACE-complete

- We replace the sub-formulas $\langle A \rangle\, \psi$ and $\langle \overline{A} \rangle\, \psi$ with the regular expressions $r_{\langle A \rangle\, \psi}$ and $r_{\langle \overline{A} \rangle\, \psi}$:

$$r_{\langle A \rangle\, \psi} := \top^* \cdot \Big(\bigcup_{w \in W_{\langle A \rangle\, \psi}} q_w \Big); \qquad r_{\langle \overline{A} \rangle\, \psi} := \Big(\bigcup_{w \in W_{\langle \overline{A} \rangle\, \psi}} q_w \Big) \cdot \top^*.$$

- To determine $W_{\langle A \rangle\, \psi}$ and $W_{\langle \overline{A} \rangle\, \psi}$, we iterate the previous algorithm

Theorem

The MC problem for formulas of $\mathsf{A\overline{A}B\overline{B}}$ *over finite Kripke structures is* PSPACE*-complete.*

$\mathsf{A\bar{A}B\bar{B}}$ is PSPACE-complete

- We replace the sub-formulas $\langle \mathsf{A} \rangle\, \psi$ and $\langle \bar{\mathsf{A}} \rangle\, \psi$ with the regular expressions $r_{\langle \mathsf{A} \rangle\, \psi}$ and $r_{\langle \bar{\mathsf{A}} \rangle\, \psi}$:

$$r_{\langle \mathsf{A} \rangle\, \psi} := \top^* \cdot \Big(\bigcup_{w \in W_{\langle \mathsf{A} \rangle\, \psi}} q_w \Big); \qquad r_{\langle \bar{\mathsf{A}} \rangle\, \psi} := \Big(\bigcup_{w \in W_{\langle \bar{\mathsf{A}} \rangle\, \psi}} q_w \Big) \cdot \top^*.$$

- To determine $W_{\langle \mathsf{A} \rangle\, \psi}$ and $W_{\langle \bar{\mathsf{A}} \rangle\, \psi}$, we iterate the previous algorithm

Theorem

The MC problem for formulas of $\mathsf{A\bar{A}B\bar{B}}$ *over finite Kripke structures is* PSPACE*-complete.*

Proof.

The purely propositional fragment of HS is hard for **PSPACE**: we prove this fact by a reduction from the **PSPACE**-complete *universality problem for regular expressions.* □

Complexity results

	Homogeneity	Regular expressions
Full HS, BE	non-elementary **EXPSPACE**-hard	non-elementary **EXPSPACE**-hard
$A\bar{A}B\bar{B}\bar{E}$, $A\bar{A}E\bar{B}\bar{E}$	**EXPSPACE** **PSPACE**-hard	non-elementary **PSPACE**-hard
$A\bar{A}\bar{B}\bar{E}$	**PSPACE**-complete	non-elementary **PSPACE**-hard
$A\bar{A}B\bar{B}$, $B\bar{B}$, $\bar{B}$, $A\bar{A}E\bar{E}$, $E\bar{E}$, $\bar{E}$	**PSPACE**-complete	**PSPACE**-complete
$A\bar{A}B$, $A\bar{A}E$, AB, $\bar{A}E$	$\mathbf{P}^{\mathbf{NP}}$-complete	**PSPACE**-complete
$A\bar{A}$, $\bar{A}B$, AE, A, $\bar{A}$	$\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$-hard	**PSPACE**-complete
Prop, B, E	**co-NP**-complete	**PSPACE**-complete

Complexity results

	Homogeneity	Regular expressions
Full HS, BE	non-elementary **EXPSPACE**-hard	non-elementary **EXPSPACE**-hard
$A\overline{A}B\overline{B}\overline{E}$, $A\overline{A}E\overline{B}\overline{E}$	**EXPSPACE** **PSPACE**-hard	non-elementary **PSPACE**-hard
$A\overline{A}\overline{B}\overline{E}$	**PSPACE**-complete	non-elementary **PSPACE**-hard
$A\overline{A}B\overline{B}$, $B\overline{B}$, $\overline{B}$, $A\overline{A}E\overline{E}$, $E\overline{E}$, $\overline{E}$	**PSPACE**-complete	**PSPACE**-complete
$A\overline{A}B$, $A\overline{A}E$, AB, $\overline{A}E$	$\mathbf{P}^{\mathbf{NP}}$-complete	**PSPACE**-complete
$A\overline{A}$, $\overline{A}B$, AE, A, $\overline{A}$	$\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$-hard	**PSPACE**-complete
Prop, B, E	**co-NP**-complete	**PSPACE**-complete

References

A. Lomuscio and J. Michaliszyn.
An epistemic Halpern-Shoham logic.
In *IJCAI*, pages 1010–1016, 2013.

A. Lomuscio and J. Michaliszyn.
Decidability of model checking multi-agent systems against a class of EHS specifications.
In *ECAI*, pages 543–548, 2014.

A. Lomuscio and J. Michaliszyn.
Model checking multi-agent systems against epistemic HS specifications with regular expressions.
In *KR*, pages 298–308, 2016.

A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron.
Checking interval properties of computations.
Acta Informatica, 2016.