An in-Depth Investigation of Interval Temporal Logic Model Checking with Regular Expressions

SEFM 2017, Trento, IT

Laura Bozzelli[#], Alberto Molinari[*], Angelo Montanari[*], Adriano Peron[#]

September 6, 2017

[#] University of Napoli "Federico II", Department of Electronic Engineering and IT
[*] University of Udine, Department of Mathematics, Computer Science, and Physics

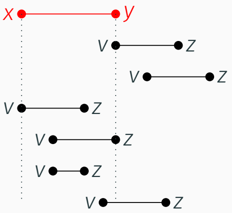
Point-based vs. interval-based model checking

- Model checking is usually point-based:
  - properties express requirements over points (snapshots) of a computation (states of the state-transition system)
  - they are specified by means of point-based temporal logics such as LTL and CTL
- Interval-based model checking:
  - Interval-based properties express conditions on computation stretches: accomplishments, actions with duration, and temporal aggregations
  - they are specified by means of interval temporal logics, e.g., HS

The logic HS

HS features a modality for any Allen ordering relation between pairs of intervals (except for equality)

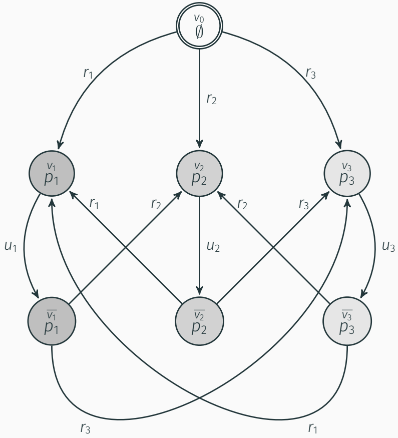
Allen rel.	HS	Definition	Example
meets	$\langle A \rangle$	$[x,y]\mathcal{R}_A[v,z] \iff y = v$	
before	$\langle L \rangle$	$[x,y]\mathcal{R}_L[v,z] \iff y < v$	
started-by	$\langle B \rangle$	$[x,y]\mathcal{R}_B[v,z] \iff x = v \wedge z < y$	
finished-by	$\langle E \rangle$	$[x,y]\mathcal{R}_E[v,z] \iff y = z \wedge x < v$	
contains	$\langle D \rangle$	$[x,y]\mathcal{R}_D[v,z] \iff x < v \wedge z < y$	
overlaps	$\langle O \rangle$	$[x,y]\mathcal{R}_O[v,z] \iff x < v < y < z$	



$$\boxed{\psi ::= p \mid \neg\psi \mid \psi \vee \psi \mid \langle X \rangle \psi \mid \langle \overline{X} \rangle \psi} \qquad X \in \{A, L, B, E, D, O\}.$$

However, all modalities can be expressed by means of $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$ and their inverse modalities $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$ only

Kripke structures

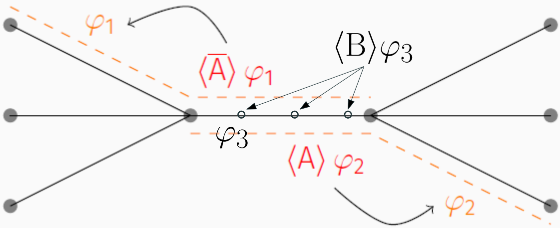


An example of Kripke structure

- HS formulas are interpreted over (finite) state-transition systems whose states are labelled with sets of proposition letters (Kripke structures)
- An interval is a trace (finite path) in a Kripke structure
- How to determine the properties of a trace (i.e. its proposition letters), in terms of its states?

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

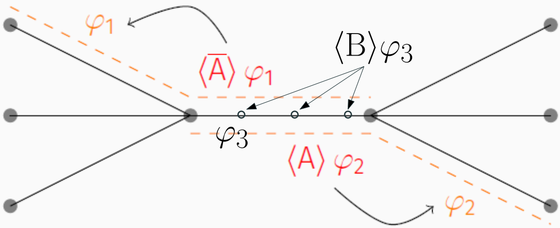
- $\mathcal{K}, \rho \models p$ iff $p \in \bigcap_{w \in \mathsf{states}(\rho)} \mu(w)$, for any letter $p \in \mathcal{AP}$ (homogeneity assumption)



HS semantics and model checking

Truth of a formula ψ over a trace 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)



Model Checking

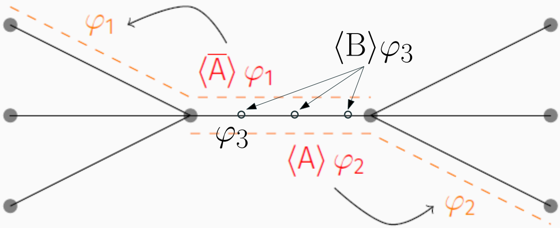
$\mathcal{K} \models \psi \iff$ for all *initial* traces ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many traces!

HS semantics and model checking

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

- $\mathcal{K}, \rho \models p$ iff $p \in \mu(\mathsf{fst}(\rho), \mathsf{lst}(\rho))$, for any letter $p \in \mathcal{AP}$ (endpoint-based labeling);



Model Checking

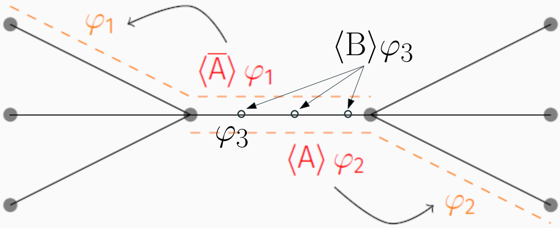
$\mathcal{K} \models \psi \iff$ for all *initial* traces ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many traces!

HS semantics and model checking

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

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



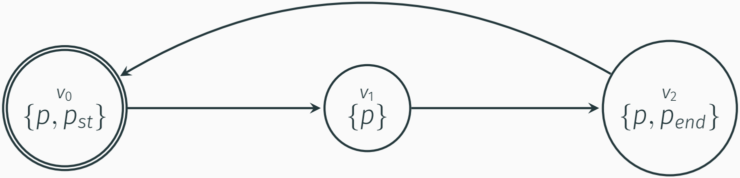
Model Checking

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

Possibly infinitely many traces!

Example—Printer

Adapted from [Lomuscio A., Michaliszyn J.]

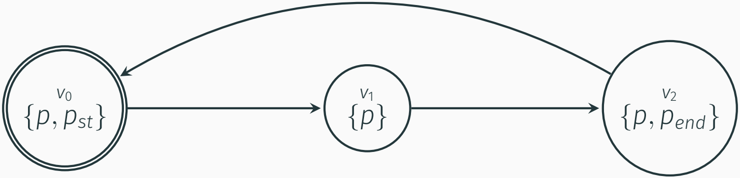


Imagine we want to label the process of printing a single sheet of paper with p.

- Under homogeneity,
  $v_0v_1v_2$ labeled by $p \Rightarrow v_0v_1$ and v_1v_2 labeled by p

Example—Printer

Adapted from [Lomuscio A., Michaliszyn J.]



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

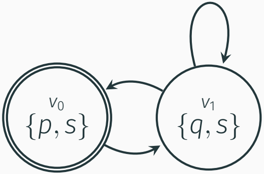
Our regular expressions

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

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

Examples:

- $r_1 = (\mathsf{p} \wedge \mathsf{s}) \cdot \mathsf{s}^* \cdot (\mathsf{p} \wedge \mathsf{s})$
- $r_2 = (\neg \mathsf{p})^*$

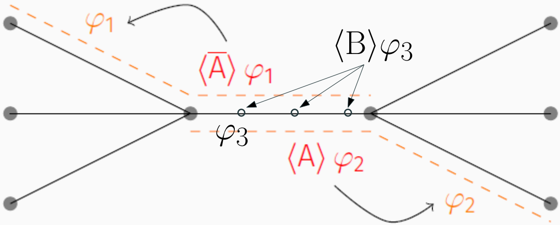


- $\rho = v_0 v_1 v_0 v_1 v_1$
- $\mu(\rho) = \{p, s\}\{q, s\}\{p, s\}\{q, s\}\{q, s\}$
- $\rho' = v_0 v_1 v_1 v_1 v_0$
- $\mu(\rho') = \{p, s\}\{q, s\}\{q, s\}\{q, s\}\{p, s\}$
  - $\mu(\rho) \notin \mathcal{L}(r_1)$, but $\mu(\rho') \in \mathcal{L}(r_1)$
  - $\mu(\rho) \notin \mathcal{L}(r_2)$, and $\mu(\rho) \notin \mathcal{L}(r_2)$

HS semantics with regular expressions

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

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



Model Checking

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

Possibly infinitely many traces!

Truth of a formula ψ over a trace 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 $w_i \in W$.

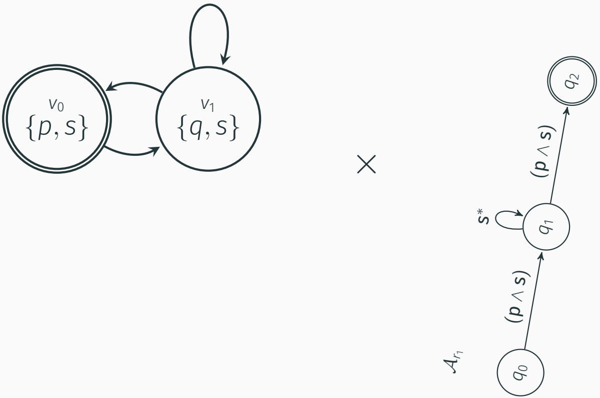
$\mathcal{K} \models \psi \iff$ for all *initial* traces ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models \psi$

Possibly infinitely many traces!

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 traces 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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

- for $\psi = \psi_1 \vee \psi_2$: we do the usual construction;
- 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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

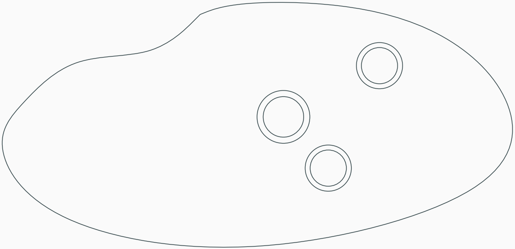
- for $\psi = \psi_1 \vee \psi_2$: we do the usual construction;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathsf{B} \rangle \, \psi'$: we add a sink final state reachable from the "old" final states;

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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

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

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

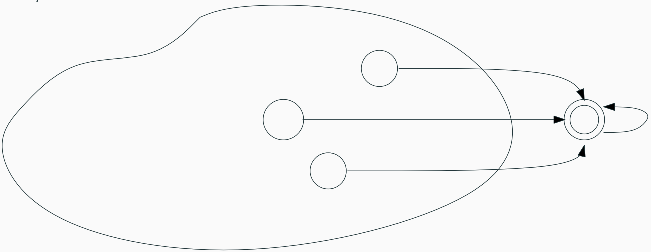


Decidability of MC for HS + regular expressions

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

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

$\mathcal{A}_{\langle B \rangle \psi'}$:



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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

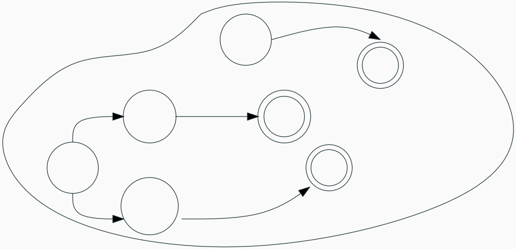
- for $\psi = \psi_1 \vee \psi_2$: we do the usual construction;
- for $\psi = \neg\psi'$: we complement the automaton (only the part for regular expressions);
- for $\psi = \langle \mathsf{B} \rangle\, \psi'$: we add a sink final state reachable from the “old” final states;
- for $\psi = \langle \overline{\mathsf{B}} \rangle\, \psi'$: 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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

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

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

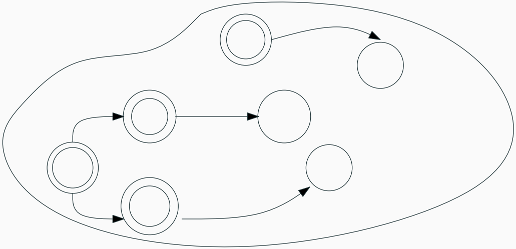


Decidability of MC for HS + regular expressions

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

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

$\mathcal{A}_{\langle \overline{B} \rangle \psi'}$:



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 traces ρ such that $\mathcal{K}, \rho \models \varphi$.

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

Theorem

The MC problem for HS (+ regular expressions) over finite Kripke structures is decidable.

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

Formulas of $A\overline{A}B\overline{B}$ + regular expressions can be checked by using polynomial working space.

The following theorem is a building block of the **PSPACE**-model checking algorithm for $A\overline{A}B\overline{B}$.

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

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

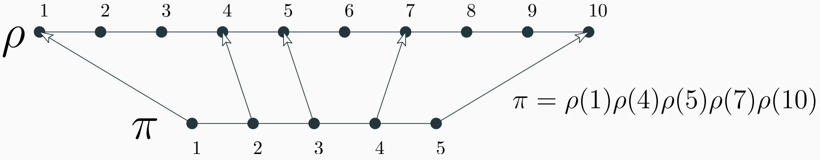
$$\mathcal{K}, \rho \models \varphi.$$

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

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

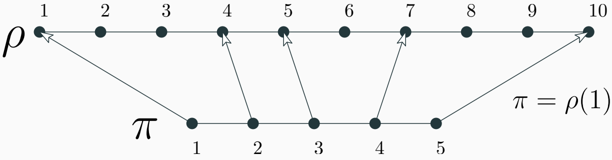
Proved by a contraction technique.

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



We guarantee that:
for all corresponding positions i, j : $\mathcal{A}^s(\mu(\pi[1 \ldots j])) = \mathcal{A}^s(\mu(\rho[1 \ldots i]))$

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



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

We guarantee that:
for all corresponding positions i, j : $\mathcal{A}^s(\mu(\pi[1 \ldots j])) = \mathcal{A}^s(\mu(\rho[1 \ldots i]))$

If $\mathcal{K}, \rho \models \varphi \quad \Rightarrow$ there is a trace π of $\mathcal{K}$, induced by ρ, such that

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

This small model is “strict”!

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

- The algorithm can consider only traces having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples summarizing traces,

$$(G, D(\psi), w)$$

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

- The algorithm can consider only traces having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples summarizing traces,

$$(G, D(\psi), w)$$

  - $G \subseteq \mathsf{Subf}_{\langle \mathsf{B} \rangle}(\psi)$
    contains the subformulas that hold on some prefix

- The algorithm can consider only traces having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples summarizing traces,

$$(G, D(\psi), w)$$

  - $D(\psi)$ is the configuration of the DFAs after reading the trace (only the current states, not the whole DFAs!),

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

- The algorithm can consider only traces having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples summarizing traces,

$$(G, D(\psi), w)$$

  - w is the last state of the trace

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

- The algorithm can consider only traces having length bounded by the exponential small model
- However, they are still too long! $\Rightarrow$ triples summarizing traces,

$$(G, D(\psi), w)$$

Lemma

If ρ and ρ' summarized by the same triple $(G, D(\psi), w)$, then $(\psi \in \mathsf{B\overline{B}})$

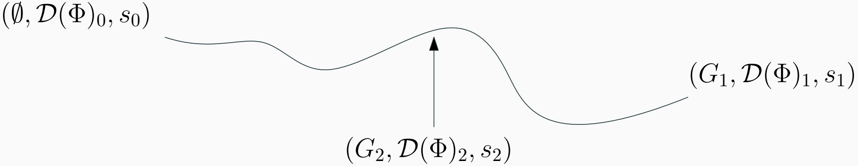
$$\mathcal{K}, \rho \models \psi \iff \mathcal{K}, \rho' \models \psi.$$

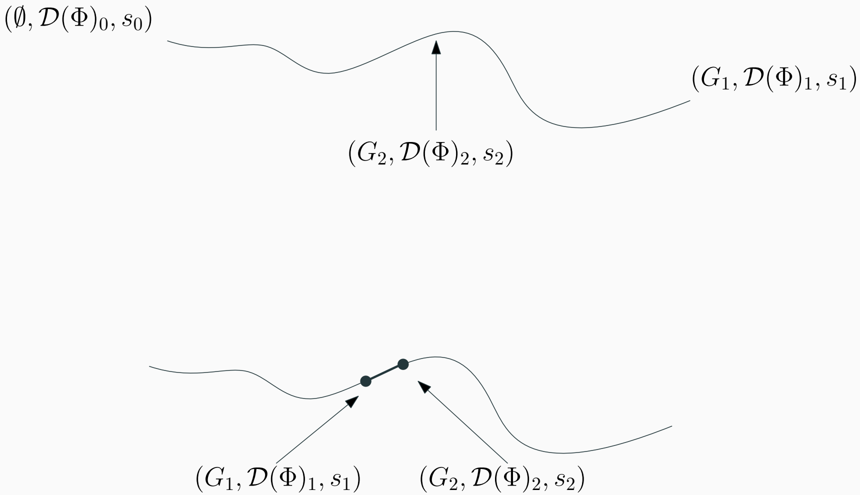
A PSPACE MC algorithm for $A\overline{A}B\overline{B}$

$(\emptyset, \mathcal{D}(\Phi)_0, s_0)$

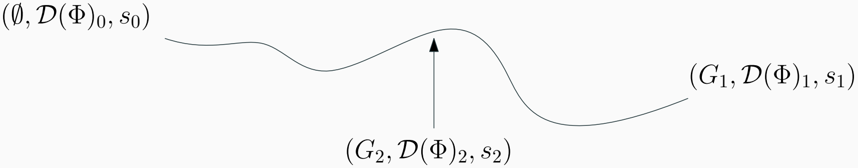
$(G_1, \mathcal{D}(\Phi)_1, s_1)$

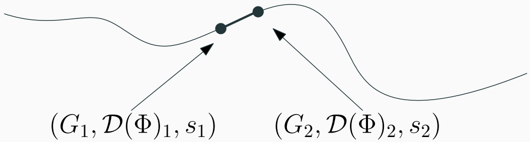
A PSPACE MC algorithm for $\mathsf{A\overline{A}B\overline{B}}$




$(\emptyset, \mathcal{D}(\Phi)_0, s_0)$
$(G_1, \mathcal{D}(\Phi)_1, s_1)$
$(G_2, \mathcal{D}(\Phi)_2, s_2)$
$(G_1, \mathcal{D}(\Phi)_1, s_1)$
$(G_2, \mathcal{D}(\Phi)_2, s_2)$

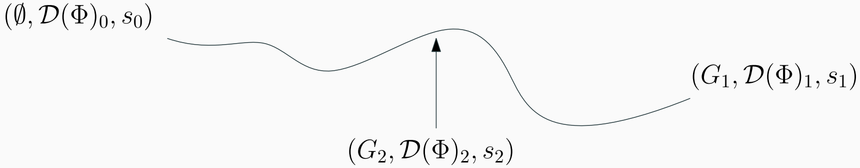
A PSPACE MC algorithm for $\mathsf{A\overline{A}B\overline{B}}$



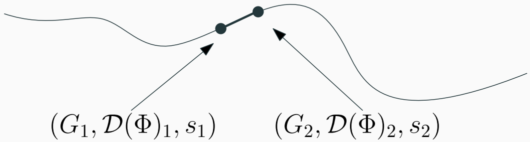


Compatible?

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



Bisection $\rightsquigarrow$ polynomial depth recursion



Compatible?

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

Algorithm 1 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**
  Call Check recursively
else if $\psi = \langle B \rangle\, \psi'$ **then**
  return $\psi' \in G$
else if $\psi = \langle \overline{B} \rangle\, \psi'$ **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 $\bot$

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**
  Call Check 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 $\bot$

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

Theorem

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

Proof.

The purely propositional fragment of HS is hard for PSPACE: 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

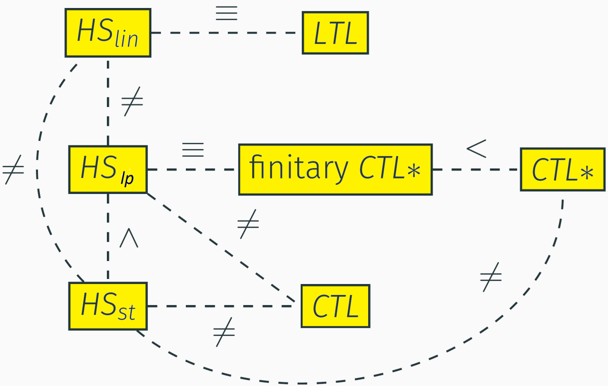
Complexity results

	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

Future work

- Determining the precise complexity of full HS
- MC over visibly pushdown systems (for recursive programs, infinite state systems)
- Application: Planning as Model Checking with HS
- Come up with inherently interval-based models of systems (Kripke structures are oriented to the description of point-based properties of systems, and their state-by-state evolution)

Expressiveness comparison



References I

L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala.
Interval Temporal Logic MC: the Border Between Good and Bad HS Fragments.
In *IJCAR*, LNAI 9706, pages 389–405. Springer, 2016.

L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala.
Interval vs. Point Temporal Logic Model Checking: an Expressiveness Comparison.
In *FSTTCS*, pages 26:1–14, 2016.

L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala.
Model Checking the Logic of Allen's Relations Meets and Started-by is P^{NP}-Complete.
In *GandALF*, pages 76–90, 2016.

References II

J. Y. Halpern and Y. Shoham.
A propositional modal logic of time intervals.
Journal of the ACM, 38(4):935–962, 1991.

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.

References III

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

A. Molinari, A. Montanari, and A. Peron.
Complexity of ITL model checking: some well-behaved fragments of the interval logic HS.
In *TIME*, pages 90–100, 2015.

A. Molinari, A. Montanari, and A. Peron.
A model checking procedure for interval temporal logics based on track representatives.
In *CSL*, pages 193–210, 2015.

A. Molinari, A. Montanari, A. Peron, and P. Sala.
Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture.
In *KR*, pages 473–483, 2016.