Model Checking the Logic of Allen's Relations *Meets* and *Started-by* is P^{NP}-Complete

GandALF 2016, Catania, Italy

Laura Bozzelli, Alberto Molinari, Angelo Montanari, Adriano Peron, Pietro Sala

September 14–16, 2016

Model checking

- Model checking: the desired properties of a system are checked against a model of the system
    - the model is a (finite) state-transition graph
    - system properties are specified by a temporal logic (e.g., LTL, CTL, CTL*, ...)
- Distinctive features of model checking:
    - exhaustive verification of all the possible behaviours
    - fully automatic process
    - a counterexample is produced for a violated property

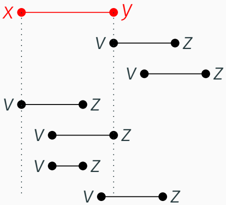
Point-based vs. interval-based model checking

- 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

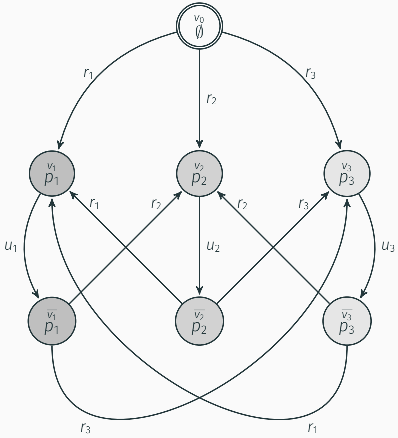
HS features a modality for any Allen ordering relation between pairs of intervals (except for equality)

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	$\langle O \rangle$	$[x, y]\mathcal{R}_O[v, z] \iff x < v < y < z$	



All modalities can be expressed by means of $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$ and their transposed modalities 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 track (finite path) in a Kripke structure

HS semantics and model checking

Truth of a formula ψ over a track ρ of a Kripke structure $\mathcal{K}$:

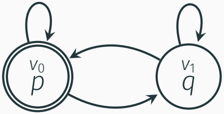
- $\mathcal{K}, \rho \models p$ iff $p \in \bigcap_{w \in \mathrm{states}(\rho)} \mu(w)$, for any letter $p \in \mathcal{AP}$ (homogeneity assumption);
- negation, disjunction, and conjunction are standard;
- $\mathcal{K}, \rho \models \langle \mathrm{A} \rangle\, \psi$ iff there is a track ρ' s.t. $\mathrm{lst}(\rho)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

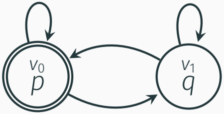
BE-descriptors



BE_2-descriptor for the track $\rho = v_0 v_1 v_0^4 v_1$
(only the part for prefixes is shown)

- $(v_0, \{v_0, v_1\}, v_1)$
  - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \emptyset, v_1)$

BE-descriptors



BE_2-descriptor for the track $\rho = v_0 v_1 v_0^4 v_1$
(only the part for prefixes is shown)

- $(v_0, \{v_0, v_1\}, v_1)$
  - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \{v_0, v_1\}, v_0)$
    - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \{v_1\}, v_0)$
    - $(v_0, \emptyset, v_1)$
  - $(v_0, \emptyset, v_1)$

- FACT 1: For any Kripke structure $\mathcal{K}$ the number of different descriptors of bounded depth k is finite
- FACT 2: Two tracks ρ and ρ' of a Kripke structure $\mathcal{K}$ described by the same BE_k-descriptor are k-equivalent

Decidability of HS model checking

Theorem

The model checking problem for full HS on Kripke structures is decidable (non-elementary *algorithm)*

Reference

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

Acta Informatica, 2016

Decidability of HS model checking

Theorem

The model checking problem for full HS on Kripke structures is decidable (non-elementary *algorithm)*

Reference

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

Acta Informatica, 2016

Theorem

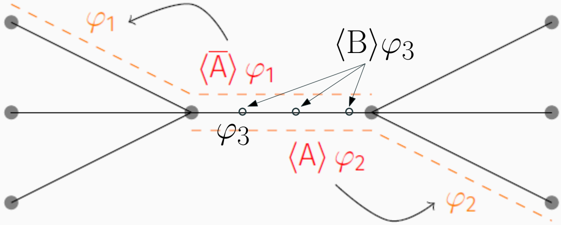
The model checking problem for BE *on Kripke structures is EXPSPACE-hard*

Reference

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

The fragment $A\bar{A}B$



- Branching semantics of $\langle A \rangle / \langle \bar{A} \rangle$
- MC for $A\bar{A}B$ is complete for $\mathsf{P}^{\mathsf{NP}} = \Delta_2^p$

P^{NP} MC algorithm for $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}$ formulas

Algorithm 1 $\mathrm{MC}(\mathcal{K}, \psi, \textsc{direction})$

for all $\langle \mathrm{A} \rangle\, \phi \in \mathrm{ModSubf}_{\mathrm{A}\overline{\mathrm{A}}}(\psi)$ **do**
    $\mathrm{MC}(\mathcal{K}, \phi, \textsc{forward})$
for all $\langle \overline{\mathrm{A}} \rangle\, \phi \in \mathrm{ModSubf}_{\mathrm{A}\overline{\mathrm{A}}}(\psi)$ **do**
    $\mathrm{MC}(\mathcal{K}, \phi, \textsc{backward})$
for all $v \in \mathrm{states}(\mathcal{K})$ **do**
    if $\textsc{direction}$ is $\textsc{forward}$ **then**
        $V_{\mathrm{A}}(\psi, v) \leftarrow \mathit{Success}(\mathtt{Oracle}(\mathcal{K}, \psi, v, \textsc{forward}, V_{\mathrm{A}} \cup V_{\overline{\mathrm{A}}}))$
    else if $\textsc{direction}$ is $\textsc{backward}$ **then**
        $V_{\overline{\mathrm{A}}}(\psi, v) \leftarrow \mathit{Success}(\mathtt{Oracle}(\mathcal{K}, \psi, v, \textsc{backward}, V_{\mathrm{A}} \cup V_{\overline{\mathrm{A}}}))$

- $\mathrm{ModSubf}_{\mathrm{A}\overline{\mathrm{A}}}(\psi)$: $\mathrm{A}\overline{\mathrm{A}}$-modal-subformulas of ψ

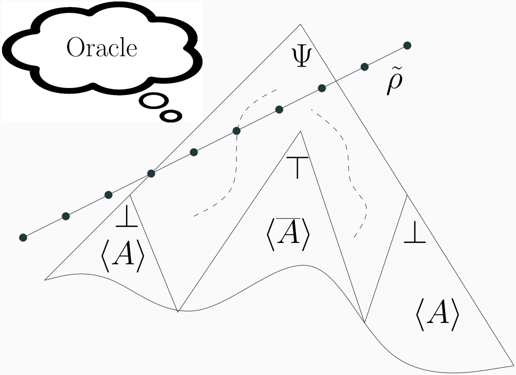
P^{NP} MC algorithm for $\mathsf{A\overline{A}B}$ formulas

Algorithm 2 MC($\mathcal{K}, \psi$, DIRECTION)

for all $\langle A \rangle \phi \in \mathrm{ModSubf}_{A\overline{A}}(\psi)$ **do**
  MC($\mathcal{K}, \phi$, FORWARD)
for all $\langle \overline{A} \rangle \phi \in \mathrm{ModSubf}_{A\overline{A}}(\psi)$ **do**
  MC($\mathcal{K}, \phi$, BACKWARD)
for all $v \in \mathrm{states}(\mathcal{K})$ **do**
  if DIRECTION is FORWARD **then**
    $V_A(\psi, v) \leftarrow \mathit{Success}(\texttt{Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_A \cup V_{\overline{A}}))$
  else if DIRECTION is BACKWARD **then**
    $V_{\overline{A}}(\psi, v) \leftarrow \mathit{Success}(\texttt{Oracle}(\mathcal{K}, \psi, v, \text{BACKWARD}, V_A \cup V_{\overline{A}}))$

- $\texttt{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\overline{A}})$ is called for all $v \in \mathrm{states}(\mathcal{K})$
- $V_A(\phi, v) = \top \iff \exists$ a track $\rho \in \mathrm{Trk}_{\mathcal{K}}$ starting from v s.t. $\mathcal{K}, \rho \models \phi$
- DIRECTION = FORWARD / BACKWARD (for $\langle A \rangle$ / $\langle \overline{A} \rangle$)

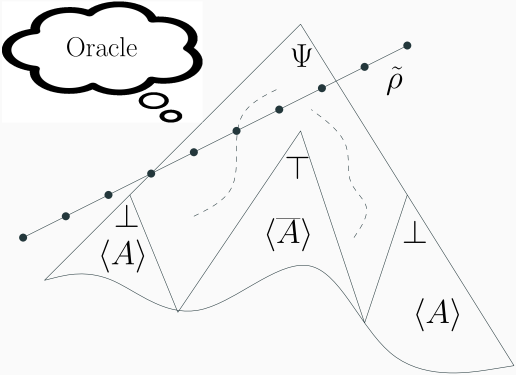
NP oracle



The oracle:

- generates $\tilde{\rho}$ by non-deterministically visiting the unravelling of the Kripke structure.
- performs a bottom-up deterministic verification of Ψ against $\tilde{\rho}$ (for all the subformulas / for all the prefixes).

NP oracle



- "polynomial-size model-track property":
  if ρ is a track of $\mathcal{K}$, ϕ is an A$\overline{\text{A}}$B formula, and $\mathcal{K}, \rho \models \phi \Rightarrow$ $\exists \rho'$ such that $|\rho'| \leq |W| \cdot (2|\phi| + 1)^2$ and $\mathcal{K}, \rho' \models \phi$.

P^{NP} MC algorithm for $\mathsf{A\overline{A}B}$ formulas

Theorem

Let $\mathcal{K}$ be a finite Kripke structure, w_0 be its initial state, and ψ an $\mathsf{A\overline{A}B}$ formula. If $\mathtt{MC}(\mathcal{K}, \neg\psi, \textit{FORWARD})$ is executed, then

$$V_{\mathsf{A}}(\neg\psi, w_0) = \bot \iff \mathcal{K} \models \psi.$$

Corollary

The model checking problem for $\mathsf{A\overline{A}B}$ formulas over finite Kripke structures is in P^{NP}.

P^{NP}-hardness of MC for AB formulas

Definition (SNSAT: a P^{NP}-complete problem)

An instance $\mathcal{I}$ of SNSAT:

- a set of Boolean variables $X = \{x_1, \cdots, x_n\}$
- a set of Boolean formulas $\{F_1(Z_1), F_2(x_1, Z_2), \cdots, F_n(x_1, \cdots, x_{n-1}, Z_n)\}$ (where Z_i are private variables)

P^{NP}-hardness of MC for AB formulas

Definition (SNSAT: a P^{NP}-complete problem)

An instance $\mathcal{I}$ of SNSAT:

- a set of Boolean variables $X = \{x_1, \cdots, x_n\}$
- a set of Boolean formulas $\{F_1(Z_1), F_2(x_1, Z_2), \cdots, F_n(x_1, \cdots, x_{n-1}, Z_n)\}$ (where Z_i are private variables)

$v_{\mathcal{I}}$ is the valuation of the variables in X defined as:

$$v_{\mathcal{I}}(x_i) = \top \iff F_i(v_{\mathcal{I}}(x_1), \cdots, v_{\mathcal{I}}(x_{i-1}), Z_i) \text{ is satisfiable.}$$

SNSAT: to decide whether $v_{\mathcal{I}}(x_n) = \top$.

P^{NP}-hardness of MC for AB formulas

Definition (SNSAT: a P^{NP}-complete problem)

An instance $\mathcal{I}$ of SNSAT:

- a set of Boolean variables $X = \{x_1, \cdots, x_n\}$
- a set of Boolean formulas $\{F_1(Z_1), F_2(x_1, Z_2), \cdots, F_n(x_1, \cdots, x_{n-1}, Z_n)\}$ (where Z_i are private variables)

$v_{\mathcal{I}}$ is the valuation of the variables in X defined as:

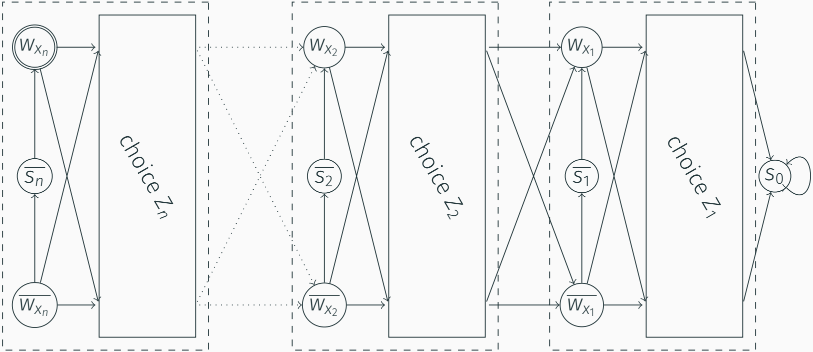
$$v_{\mathcal{I}}(x_i) = \top \iff F_i(v_{\mathcal{I}}(x_1), \cdots, v_{\mathcal{I}}(x_{i-1}), Z_i) \text{ is satisfiable.}$$

SNSAT: to decide whether $v_{\mathcal{I}}(x_n) = \top$.

Given $\mathcal{I}$, we build a Kripke structure $\mathcal{K}_{\mathcal{I}}$ and an AB formula $\Phi_{\mathcal{I}}$ s.t.

$$v_{\mathcal{I}}(x_n) = \top \iff \mathcal{K}_{\mathcal{I}} \models \Phi_{\mathcal{I}}.$$

P^{NP}-hardness of MC for AB formulas



$\psi_n =$ (1) $\bigwedge_i x_i \Rightarrow F_i(x_1, \cdots, x_{i-1}, Z_i)$ is true
(2) $\bigwedge_i \neg x_i \Rightarrow F_i(x_1, \cdots, x_{i-1}, Z_i)$ is unsat for any choice of Z_i
(3) the track reaches the last state s_0

P^{NP}-hardness of MC for AB formulas

Theorem

$$v_{\mathcal{I}}(x_n) = \top \iff \mathcal{K}_{\mathcal{I}} \models [B]\bot \to \psi_n.$$

Corollary

The model checking problem for AB *formulas over finite Kripke structures is* P^{NP}*-hard (under* LOGSPACE *reductions).*

Therefore AB, $A\overline{A}B$, $\overline{A}E$, $A\overline{A}E$ are P^{NP}-complete.

The fragment $\overline{\mathsf{A}}\mathsf{B}$

- AB allows one to impose specific constraints on the branches departing from a state occurring in a given path... $\langle B \rangle \langle A \rangle \theta$
  $\Rightarrow$ P^{NP}-hardness of AB.

- $\overline{\mathsf{A}}\mathsf{B}$...

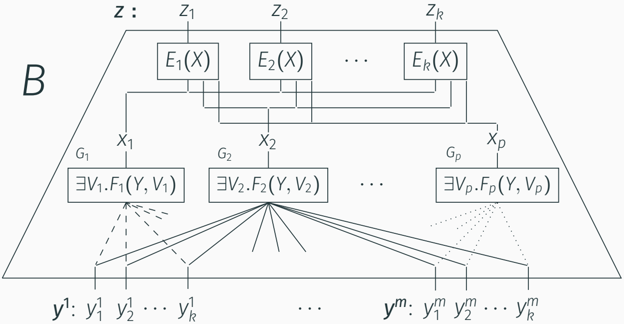
The fragment $\overline{\mathrm{A}}\mathrm{B}$

- AB allows one to impose specific constraints on the branches departing from a state occurring in a given path... $\langle \mathrm{B} \rangle \langle \mathrm{A} \rangle \theta$
  $\Rightarrow$ P^{NP}-hardness of AB.

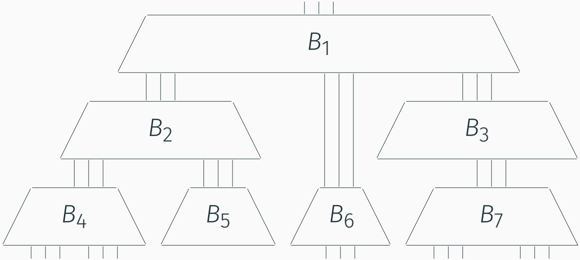
- $\overline{\mathrm{A}}\mathrm{B}$... can't express constraints of this form: pairing $\langle \overline{\mathrm{A}} \rangle$ and $\langle \mathrm{B} \rangle$ does not give any advantage in terms of expressiveness
  $\Rightarrow$ MC for $\overline{\mathrm{A}}\mathrm{B}$ in $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$.

The fragment $\overline{\mathrm{A}}\mathrm{B}$

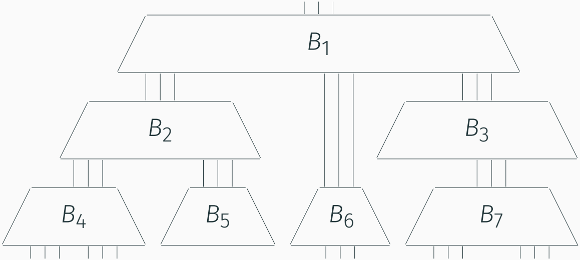
Membership to $\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$ is proved by means of Boolean circuits with SAT oracles



The fragment $\overline{\mathrm{A}}\mathrm{B}$: tree of blocks



The fragment $\overline{\mathsf{A}}\mathsf{B}$: tree of blocks



Theorem

TB(SAT) is P^{NP}*-complete.*
TB(SAT)$_{1\times M}$ *(i.e., F_i can use* only one bit from each input vector *of B) is* $\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$*-complete.*

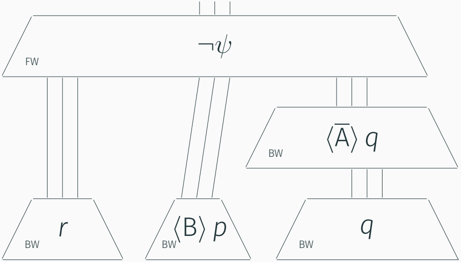
Reference

P. Schnoebelen. Oracle circuits for branching-time model checking.

In *ICALP*, pages 790–801, 2003

The fragment $\overline{\mathsf{A}}\mathsf{B}$: from a formula to a tree of blocks

$$\psi = ((\langle\overline{\mathsf{A}}\rangle\, r \wedge \langle\overline{\mathsf{A}}\rangle\langle\overline{\mathsf{A}}\rangle\, q) \rightarrow \langle\overline{\mathsf{A}}\rangle\langle\mathsf{B}\rangle\, p)$$



Every formula F_i of a block B:

1. is a translation of the oracle algorithm;
2. is built starting from the $\overline{\mathsf{A}}\mathsf{B}$ formula associated with B

The fragment $\overline{\mathsf{A}}\mathsf{B}$: complexity

Theorem

$B_{root}(z_1) = \bot \iff \mathcal{K} \models \psi.$

Corollary

The model checking problem for $\overline{\mathsf{A}}\mathsf{B}$ *formulas, over finite Kripke structures, is in* $\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$.

Theorem

$B_{root}(z_1) = \bot \iff \mathcal{K} \models \psi.$

Corollary

The model checking problem for $\overline{\mathsf{A}}\mathsf{B}$ *formulas, over finite Kripke structures, is in* $\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$.

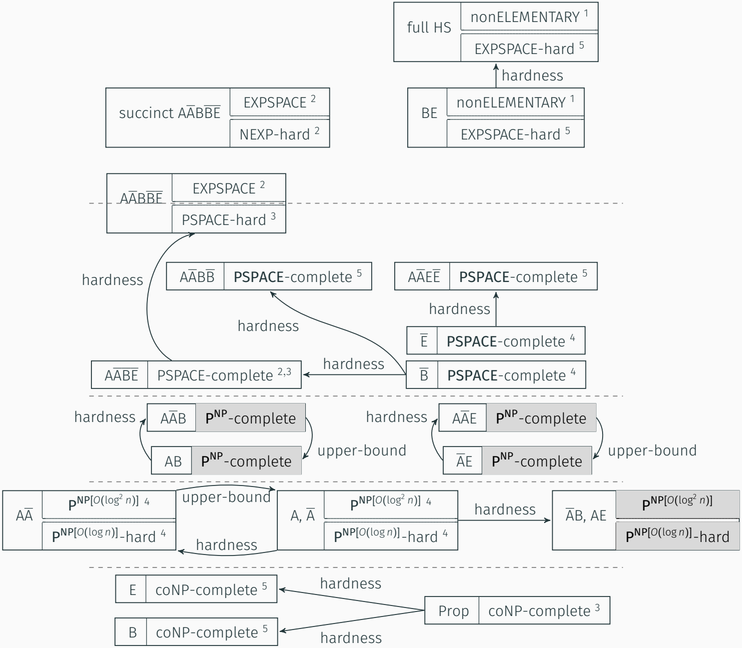
$\mathsf{P}^{\mathsf{NP}[O(\log n)]}$-hardness follows immediately from that of $\overline{\mathsf{A}}$

Reference

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

Complexity picture



Current/future work

- Determining the precise complexity of full HS

Current/future work

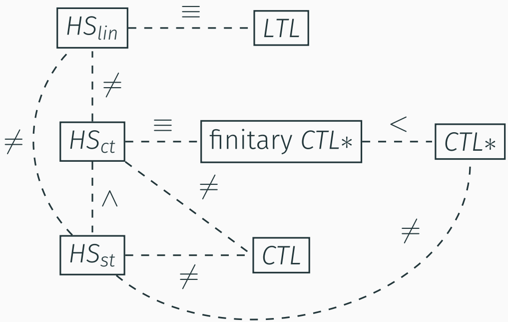
- Determining the precise complexity of full HS
- Relaxing the homogeneity assumption

Current/future work

- Determining the precise complexity of full HS
- Relaxing the homogeneity assumption
- Comparison of HS model checking with LTL, CTL, and CTL* one (two new semantic variants of the problem introduced, respectively based on the linear-past semantics and the linear semantics) - *DONE*

- Determining the precise complexity of full HS
- Relaxing the homogeneity assumption
- Comparison of HS model checking with LTL, CTL, and CTL* one (two new semantic variants of the problem introduced, respectively based on the linear-past semantics and the linear semantics) - *DONE*
- Application: Planning as Model Checking in Interval Temporal Logic - *IN PROGRESS*

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.

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.

References II

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.

P. Schnoebelen.
Oracle circuits for branching-time model checking.
In *ICALP*, pages 790–801, 2003.