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)

<table>
<thead>
<tr>
<th>Allen rel.</th>
<th>HS</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>meets</td>
<td>⟨A⟩</td>
<td>[x, y]RA[v, z] ↔ y = v</td>
<td></td>
</tr>
<tr>
<td>before</td>
<td>⟨L⟩</td>
<td>[x, y]RL[v, z] ↔ y &lt; v</td>
<td></td>
</tr>
<tr>
<td>started-by</td>
<td>⟨B⟩</td>
<td>[x, y]RB[v, z] ↔ x = v ∧ z &lt; y</td>
<td></td>
</tr>
<tr>
<td>finished-by</td>
<td>⟨E⟩</td>
<td>[x, y]RE[v, z] ↔ y = z ∧ x &lt; v</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>⟨D⟩</td>
<td>[x, y]RD[v, z] ↔ x &lt; v ∧ z &lt; y</td>
<td></td>
</tr>
<tr>
<td>overlaps</td>
<td>⟨O⟩</td>
<td>[x, y]RO[v, z] ↔ x &lt; v &lt; y &lt; z</td>
<td></td>
</tr>
</tbody>
</table>

All modalities can be expressed by means of ⟨A⟩, ⟨B⟩, ⟨E⟩ and their transposed modalities only
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 $\psi$ over a track $\rho$ of a Kripke structure $\mathcal{K}$:

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

Model Checking

$\mathcal{K} \models \psi \iff$ for all initial tracks $\rho$ 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)

FACT 1: For any Kripke structure $K$ the number of different descriptors of bounded depth $k$ is finite.

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

FACT 2: Two tracks $\rho$ and $\rho'$ of a Kripke structure $\mathcal{K}$ described by the same $BE_k$-descriptor are $k$-equivalent
## Decidability of HS model checking

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>The model checking problem for full HS on Kripke structures is decidable (non-elementary algorithm)</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
</tr>
</thead>
</table>

*Acta Informatica, 2016*
## Decidability of HS model checking

<table>
<thead>
<tr>
<th>Theorem</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>The model checking problem for full HS on Kripke structures is decidable (non-elementary algorithm)</td>
<td>A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. Checking interval properties of computations.</td>
</tr>
<tr>
<td></td>
<td><em>Acta Informatica</em>, 2016</td>
</tr>
<tr>
<td>The model checking problem for BE on Kripke structures is EXPSPACE-hard</td>
<td>L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Interval Temporal Logic MC: the Border Between Good and Bad HS Fragments.</td>
</tr>
</tbody>
</table>
The fragment $\langle A \rangle A\langle B \rangle$

- Branching semantics of $\langle A \rangle / \langle A \rangle$
- MC for $A\langle A \rangle B$ is complete for $P^{NP} = \Delta_2^p$
Algorithm 1 MC(\(\mathcal{K}, \psi, \text{DIRECTION}\))

1: \textbf{for all} \(\langle A \rangle \phi \in \text{ModSubf}_{\overline{A}}(\psi)\) \textbf{do}
2: \hspace{1em} \text{MC}(\mathcal{K}, \phi, \text{FORWARD})
3: \textbf{for all} \(\langle \overline{A} \rangle \phi \in \text{ModSubf}_{\overline{A}}(\psi)\) \textbf{do}
4: \hspace{1em} \text{MC}(\mathcal{K}, \phi, \text{BACKWARD})
5: \textbf{for all} \(v \in \text{states}(\mathcal{K})\) \textbf{do}
6: \hspace{1em} \textbf{if} \ DIRECTION \ is \ FORWARD \ \textbf{then}
7: \hspace{2em} V_A(\psi, v) \leftarrow \text{Success}(\text{Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_A \cup V_{\overline{A}}))
8: \hspace{1em} \textbf{else if} \ DIRECTION \ is \ BACKWARD \ \textbf{then}
9: \hspace{2em} V_{\overline{A}}(\psi, v) \leftarrow \text{Success}(\text{Oracle}(\mathcal{K}, \psi, v, \text{BACKWARD}, V_A \cup V_{\overline{A}}))

- \text{ModSubf}_{\overline{A}}(\psi): \overline{A}\text{-modal-subformulas of } \psi
Algorithm 2 $\text{MC}(\mathcal{K}, \psi, \text{DIRECTION})$

1: for all $\langle A \rangle \phi \in \text{ModSubf}_{\text{AA}}(\psi)$ do
2: \hspace{1em} $\text{MC}(\mathcal{K}, \phi, \text{FORWARD})$
3: for all $\langle \overline{A} \rangle \phi \in \text{ModSubf}_{\overline{A}\overline{A}}(\psi)$ do
4: \hspace{1em} $\text{MC}(\mathcal{K}, \phi, \text{BACKWARD})$
5: for all $v \in \text{states}(\mathcal{K})$ do
6: \hspace{1em} if DIRECTION is FORWARD then
7: \hspace{2em} $V_A(\psi, v) \leftarrow \text{Success(Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_A \cup V_{\overline{A}}))$
8: \hspace{1em} else if DIRECTION is BACKWARD then
9: \hspace{2em} $V_{\overline{A}}(\psi, v) \leftarrow \text{Success(Oracle}(\mathcal{K}, \psi, v, \text{BACKWARD}, V_A \cup V_{\overline{A}}))$

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

- generates $\tilde{\rho}$ by non-deterministically visiting the unravelling of the Kripke structure.
- performs a bottom-up deterministic verification of $\Psi$ against $\tilde{\rho}$ (for all the subformulas / for all the prefixes).
“polynomial-size model-track property”:
if $\rho$ is a track of $\mathcal{K}$, $\phi$ is an $\overline{AAB}$ formula, and $\mathcal{K}, \rho \models \phi \Rightarrow \exists \rho' \text{ such that } |\rho'| \leq |W| \cdot (2|\phi| + 1)^2$ and $\mathcal{K}, \rho' \models \phi$.
**Theorem**

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

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

**Corollary**

The model checking problem for $\overline{A\overline{A}B}$ formulas over finite Kripke structures is in $\mathbf{P}^{\mathbf{NP}}$. 
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 $X = \{x_1, \ldots, x_n\}$
- a set of Boolean formulas $\{F_1(Z_1), F_2(x_1, Z_2), \ldots, F_n(x_1, \ldots, x_{n-1}, Z_n)\}$
  (where $Z_i$ are private variables)
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\(^{\text{NP}}\)-hardness of MC for AB formulas**

**Definition (SNSAT: a P\(^{\text{NP}}\)-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

\[ \psi_n = \begin{align*}
(1) & \quad \bigwedge_i x_i \Rightarrow F_i(x_1, \cdots, x_{i-1}, Z_i) \text{ is true} \\
(2) & \quad \bigwedge_i \neg x_i \Rightarrow F_i(x_1, \cdots, x_{i-1}, Z_i) \text{ is unsat for any choice of } Z_i \\
(3) & \quad \text{the track reaches the last state } s_0
\end{align*} \]
**Theorem**

\[ \nu_I(x_n) = T \iff K_I \models [B] \bot \rightarrow \psi_n. \]

**Corollary**

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

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

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

- $\overline{AB}$...
The fragment $\overline{AB}$

- $AB$ 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 $AB$.

- $\overline{AB}$... can’t express constraints of this form: pairing $\langle \overline{A} \rangle$ and $\langle B \rangle$ does not give any advantage in terms of expressiveness $\Rightarrow MC$ for $\overline{AB}$ in $P^{NP}[O(\log^2 n)]$. 
Membership to $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ is proved by means of Boolean circuits with SAT oracles.
The fragment $\overline{AB}$: tree of blocks

\begin{enumerate}
\item $B_1$
\item $B_2$
\item $B_3$
\item $B_4$
\item $B_5$
\item $B_6$
\item $B_7$
\end{enumerate}

Theorem $TB(SAT)$ is $\mathsf{P}$-complete.

$TB(SAT)$ \text{(i.e., $F_i$ can use only one bit from each input vector of $B$)}$ is $\mathsf{P}$-complete.

Reference

The fragment $\overline{AB}$: 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 $P^{NP[O(\log^2 n)]}$-complete.

Reference

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

In *ICALP*, pages 790–801, 2003
The fragment $\overline{A}B$: from a formula to a tree of blocks

$$\psi = (((\langle A \rangle r \land \langle A \rangle \langle A \rangle q) \rightarrow \langle A \rangle \langle B \rangle p)$$

Every formula $F_i$ of a block $B$:

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

**Theorem**

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

**Corollary**

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

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{root}(z_1) = \bot \iff \mathcal{K} \models \psi.$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Corollary</th>
</tr>
</thead>
<tbody>
<tr>
<td>The model checking problem for $\overline{AB}$ formulas, over finite Kripke structures, is in $P^{NP}[O(\log^2 n)]$.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Reference</th>
</tr>
</thead>
</table>

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
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
- Application: Planning as Model Checking in Interval Temporal Logic - IN PROGRESS
Expressiveness comparison

\[ HS_{lin} \equiv LTL \]

\[ HS_{ct} \not\equiv \text{finitary } CTL^* \]

\[ HS_{st} \not\equiv CTL \]

\[ CTL^* \prec CTL \]


A. Molinari, A. Montanari, and A. Peron.  
**A model checking procedure for interval temporal logics based on track representatives.**  

A. Molinari, A. Montanari, A. Peron, and P. Sala.  
**Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture.**  

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