Interval Temporal Logic Model Checking

(Invited talk)

Angelo Montanari
Dept. of Mathematics, Computer Science, and Physics
University of Udine, Italy

TIME 2016
Technical University of Denmark, Lyngby (Denmark)
October 18, 2016

Model checking

Model checking: the desired properties of a system are checked against a model of it

- the model is usually a (finite) state-transition system
- system properties are specified by a temporal logic (LTL, CTL, and the like)

Distinctive features of model checking:

- exaustive check 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

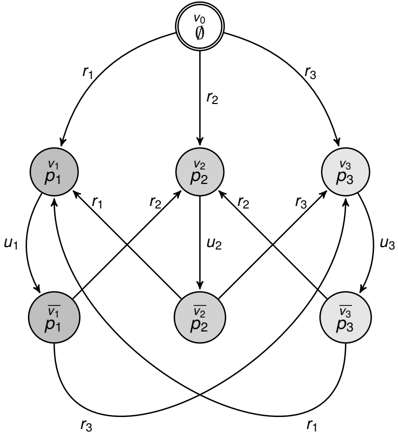
Interval-based properties express conditions on computation stretches, e.g., actions with duration, accomplishments, and temporal aggregations, instead of on computation states

Little work has been done on interval temporal logic (ITL) model checking (Bozzelli, Lomuscio, Michaliszyn, Molinari, Montanari, Murano, Perelli, Peron, Sala)

Outline of the talk

- the model checking problem for interval temporal logics
- complexity results: the general picture
- the case of the interval temporal logic $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{BE}}$
- model checking epistemic interval temporal logics
- a comparison with LTL, CTL, and CTL* model checking

The modeling of the system: Kripke structures



A finite Kripke structure

- HS formulas are interpreted over (finite) state-transition systems, whose states are labeled with sets of proposition letters (Kripke structures)
- An interval is a trace (finite path) in a Kripke structure

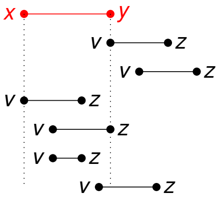
HS: the modal logic of Allen's interval relations

The thirteen binary ordering relations between two intervals on a linear order form the set of *Allen's interval relations*

They give rise to corresponding unary modalities over frames where intervals are primitive entities:

- 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

HS semantics and model checking

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

- $\mathcal{K}, \rho \models p$ 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;
- $\mathcal{K}, \rho \models \langle \text{A} \rangle\, \psi$ iff there is a trace 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

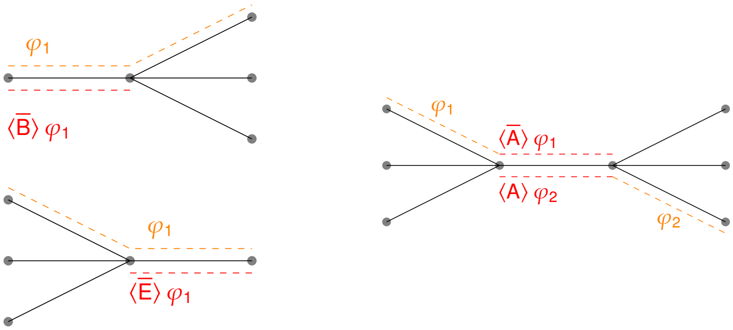
Model Checking

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

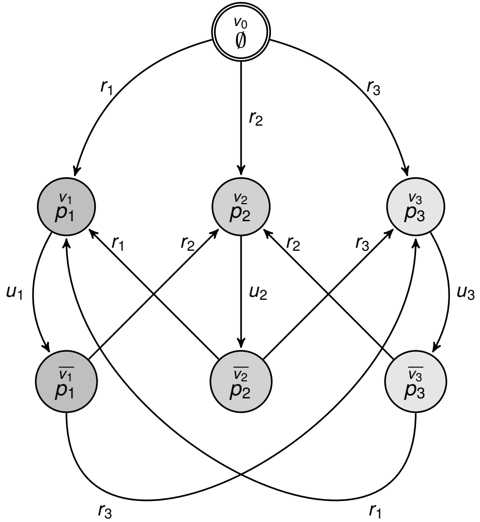
Possibly infinitely many traces!

Remark: HS state semantics (HS_{st})

- According to the given semantics, HS modalities allow one to branch both in the past and in the future



An example: the Kripke structure $\mathcal{K}_{Sched}$



A short account of $\mathcal{K}_{Sched}$

$\mathcal{K}_{Sched}$ models the behaviour of a scheduler serving 3 processes which are continuously requesting the use of a common resource

Initial state: v_0 (no process is served in that state)

In v_i and $\overline{v}_i$ the i-th process is served (p_i holds in those states)

The scheduler cannot serve the same process twice in two successive rounds:

- process i is served in state v_i, then, after "some time", a transition u_i from v_i to $\overline{v}_i$ is taken; subsequently, process i cannot be served again immediately, as v_i is not directly reachable from $\overline{v}_i$
- a transition r_j, with $j \neq i$, from $\overline{v}_i$ to v_j is then taken and process j is served

It can be easily generalised to an arbitrary number of processes

Some meaningful properties to be checked over $\mathcal{K}_{Sched}$

Validity of properties over all legal computation intervals can be forced by modality $[E]$ (they are suffixes of at least one initial trace)

Property 1: in any computation interval of length at least 4, at least 2 processes are witnessed (YES/no process can be executed twice in a row)

$$\mathcal{K}_{Sched} \models [E](\langle \mathrm{E} \rangle^3 \top \rightarrow (\chi(p_1, p_2) \vee \chi(p_1, p_3) \vee \chi(p_2, p_3))),$$

where $\chi(p, q) = \langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle p \wedge \langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle q$

Property 2: in any computation interval of length at least 11, process 3 is executed at least once (NO/the scheduler can postpone the execution of a process ad libitum)

$$\mathcal{K}_{Sched} \not\models [E](\langle \mathrm{E} \rangle^{10} \top \rightarrow \langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle p_3)$$

Property 3: in any computation interval of length at least 6, all processes are witnessed (NO/the scheduler should be forced to execute them in a strictly periodic manner, which is not the case)

$$\mathcal{K}_{Sched} \not\models [E](\langle \mathrm{E} \rangle^5 \rightarrow (\langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle p_1 \wedge \langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle p_2 \wedge \langle \mathrm{E} \rangle \langle \overline{\mathrm{A}} \rangle p_3))$$

Model checking: the key notion of BE_k-descriptor

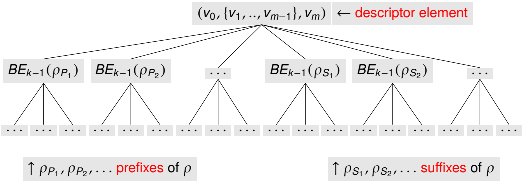
- The BE-nesting depth of an HS formula ψ ($\mathrm{Nest}_{\mathrm{BE}}(\psi)$) is the maximum degree of nesting of modalities B and E in ψ
- Two traces ρ and ρ' of a Kripke structure $\mathcal{K}$ are k-equivalent if and only if $\mathcal{K}, \rho \models \psi$ iff $\mathcal{K}, \rho' \models \psi$ for all HS-formulas ψ with $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$

Model checking: the key notion of BE_k-descriptor

- The BE-nesting depth of an HS formula ψ ($\text{Nest}_{\text{BE}}(\psi)$) is the maximum degree of nesting of modalities B and E in ψ
- Two traces ρ and ρ' of a Kripke structure $\mathcal{K}$ are k-equivalent if and only if $\mathcal{K}, \rho \models \psi$ iff $\mathcal{K}, \rho' \models \psi$ for all HS-formulas ψ with $\text{Nest}_{\text{BE}}(\psi) \leq k$

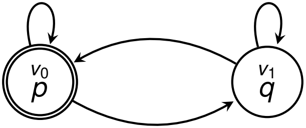
We provide a suitable tree representation for a trace, called a BE_k-descriptor

The BE_k-descriptor for a trace $\rho = v_0 v_1 .. v_{m-1}\ v_m$, denoted $BE_k(\rho)$, is defined as follows:

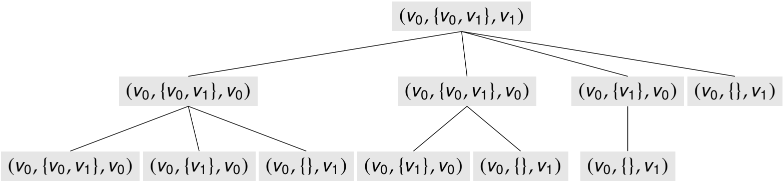


Remark: the descriptor does not feature sibling isomorphic subtrees

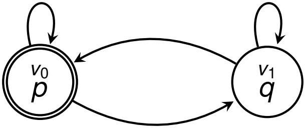
An example of a BE_2-descriptor



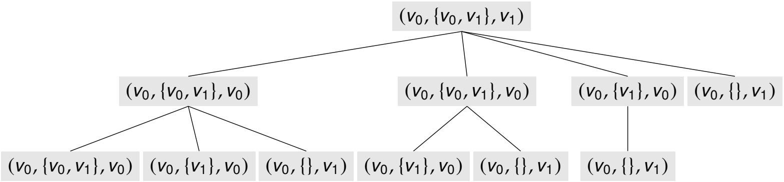
The BE_2-descriptor for the trace $\rho = v_0 v_1 v_0^4 v_1$ (for the sake of readability, only the subtrees for prefixes are displayed)



An example of a BE_2-descriptor



The BE_2-descriptor for the trace $\rho = v_0 v_1 v_0^4 v_1$ (for the sake of readability, only the subtrees for prefixes are displayed)



Remark: the subtree to the left is associated with both prefixes $v_0 v_1 v_0^3$ and $v_0 v_1 v_0^4$ (there are no sibling isomorphic subtrees in the descriptor)

Decidability of model checking for full HS

FACT 1: For any Kripke structure $\mathcal{K}$ and any BE-nesting depth $k \geq 0$, the number of different BE_k-descriptors is finite (and thus at least one descriptor has to be associated with infinitely many traces)

Decidability of model checking for full HS

FACT 1: For any Kripke structure $\mathcal{K}$ and any BE-nesting depth $k \geq 0$, the number of different BE_k-descriptors is finite (and thus at least one descriptor has to be associated with infinitely many traces)

FACT 2: Two traces ρ and ρ' of a Kripke structure $\mathcal{K}$ described by the same BE_k descriptor are k-equivalent

Decidability of model checking for full HS

FACT 1: For any Kripke structure $\mathcal{K}$ and any BE-nesting depth $k \geq 0$, the number of different BE_k-descriptors is finite (and thus at least one descriptor has to be associated with infinitely many traces)

FACT 2: Two traces ρ and ρ' of a Kripke structure $\mathcal{K}$ described by the same BE_k descriptor are k-equivalent

Theorem

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

A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron, Checking Interval Properties of Computations, Acta Informatica, Special Issue: Temporal Representation and Reasoning (TIME'14), Vol. 56, n. 6-8, October 2016, pp. 587-619

Decidability of model checking for full HS

FACT 1: For any Kripke structure $\mathcal{K}$ and any BE-nesting depth $k \geq 0$, the number of different BE_k-descriptors is finite (and thus at least one descriptor has to be associated with infinitely many traces)

FACT 2: Two traces ρ and ρ' of a Kripke structure $\mathcal{K}$ described by the same BE_k descriptor are k-equivalent

Theorem

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

A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron, Checking Interval Properties of Computations, Acta Informatica, Special Issue: Temporal Representation and Reasoning (TIME'14), Vol. 56, n. 6-8, October 2016, pp. 587-619

What about lower bounds?

The logic BE

Theorem

The model checking problem for BE*, over finite Kripke structures, is EXPSPACE-hard*

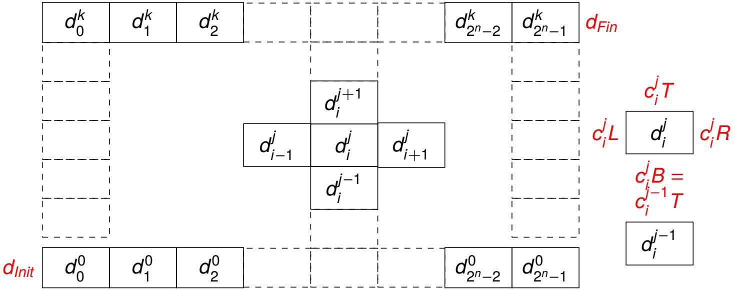
L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala, Interval Temporal Logic Model Checking: The Border Between Good and Bad HS Fragments, IJCAR 2016

Proof (sketch): a polynomial-time reduction from a domino-tiling problem for grids with rows of single exponential length

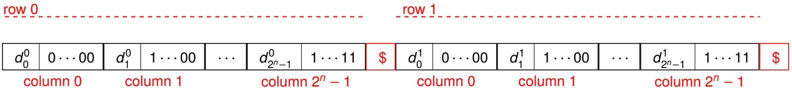
- for an instance $\mathcal{I}$ of the problem, we build a Kripke structure $\mathcal{K}_{\mathcal{I}}$ and a BE formula $\varphi_{\mathcal{I}}$ in polynomial time
- there is an initial trace of $\mathcal{K}_{\mathcal{I}}$ satisfying $\varphi_{\mathcal{I}}$ iff there is a tiling of $\mathcal{I}$
- $\mathcal{K}_{\mathcal{I}} \models \neg\varphi_{\mathcal{I}}$ iff there exists no tiling of $\mathcal{I}$

BE hardness: encoding of the domino-tiling problem

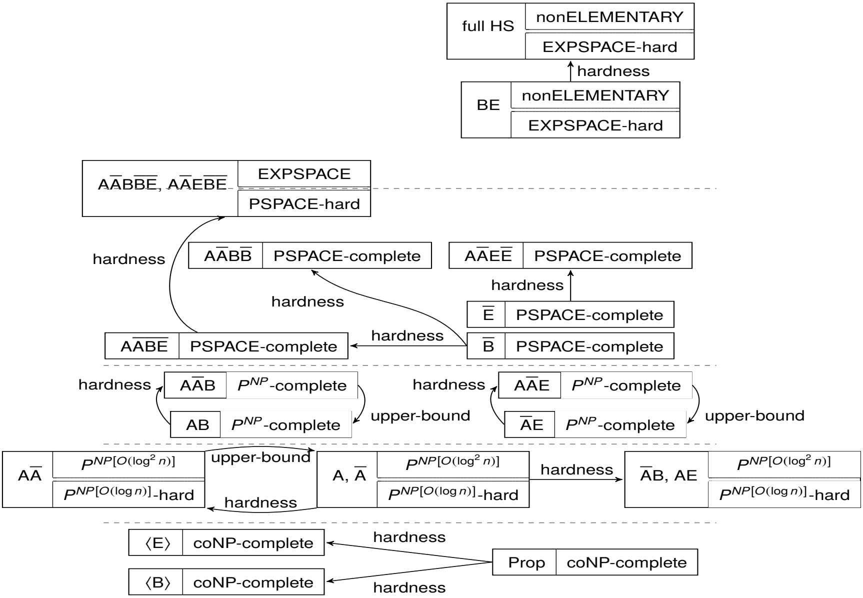
Instance of the tiling problem: $(C, \Delta, n, d_{init}, d_{final})$, with C a finite set of colors and $\Delta \subseteq C \times C \times C \times C$ a set of tuples (c_B, c_L, c_T, c_R)



String (interval) encoding of the problem



The complexity picture



Three main gaps to fill

The picture shows that there three main gaps to fill:

- full HS and BE are in between nonELEMENTARY and EXPSPACE
- $A\overline{A}B\overline{B}\overline{E}$, $A\overline{A}E\overline{B}\overline{E}$, $AB\overline{B}\overline{E}$, $AE\overline{B}\overline{E}$, $\overline{A}B\overline{B}\overline{E}$, and $\overline{A}E\overline{B}\overline{E}$ are in between EXPSPACE and PSPACE
- A, $\overline{A}$, $A\overline{A}$, $\overline{A}B$, and AE are in between $P^{NP[O(\log^2 n)]}$ and $P^{NP[O(\log n)]}$

The logic $A\overline{A}B\overline{B}\overline{E}$

Let us consider the case of the logic $A\overline{A}B\overline{B}\overline{E}$, which is obtained from full HS ($A\overline{A}B\overline{B}E\overline{E}$) by removing modality $\langle E \rangle$

The logic $A\overline{A}B\overline{B}\overline{E}$

Let us consider the case of the logic $A\overline{A}B\overline{B}\overline{E}$, which is obtained from full HS ($A\overline{A}B\overline{B}E\overline{E}$) by removing modality $\langle E \rangle$

A high-level account of the solution:

- we can restrict our attention to prefixes (B_k-descriptors suffice)

The logic $A\overline{A}B\overline{BE}$

Let us consider the case of the logic $A\overline{A}B\overline{BE}$, which is obtained from full HS ($A\overline{A}B\overline{B}E\overline{E}$) by removing modality $\langle E \rangle$

A high-level account of the solution:

- we can restrict our attention to prefixes (B_k-descriptors suffice)
- the size of the tree representation of B_k-descriptors is larger than necessary (redundancy) and it prevents their efficient exploitation in model checking algorithms

The logic $A\overline{A}B\overline{B}\overline{E}$

Let us consider the case of the logic $A\overline{A}B\overline{B}\overline{E}$, which is obtained from full HS ($A\overline{A}B\overline{B}E\overline{E}$) by removing modality $\langle E \rangle$

A high-level account of the solution:

- we can restrict our attention to prefixes (B_k-descriptors suffice)
- the size of the tree representation of B_k-descriptors is larger than necessary (redundancy) and it prevents their efficient exploitation in model checking algorithms
- a trace representative can be chosen to represent a (possibly infinite) set of traces with the same B_k-descriptor

The logic $A\overline{A}B\overline{B}\overline{E}$

Let us consider the case of the logic $A\overline{A}B\overline{B}\overline{E}$, which is obtained from full HS ($A\overline{A}B\overline{B}E\overline{E}$) by removing modality $\langle E \rangle$

A high-level account of the solution:

- we can restrict our attention to prefixes (B_k-descriptors suffice)
- the size of the tree representation of B_k-descriptors is larger than necessary (redundancy) and it prevents their efficient exploitation in model checking algorithms
- a trace representative can be chosen to represent a (possibly infinite) set of traces with the same B_k-descriptor
- a bound, which depends on both the number $|W|$ of states of the Kripke structure and the B-nesting depth k, can be given to the length of trace representatives

Prefix-bisimilarity

Definition (Prefix-bisimilarity)

Two traces ρ and ρ' are *h*-prefix bisimilar if the following conditions inductively hold:

- for $h = 0$:
  $\text{fst}(\rho) = \text{fst}(\rho')$, $\text{lst}(\rho) = \text{lst}(\rho')$, and $\text{states}(\rho) = \text{states}(\rho')$
- for $h > 0$:
  ρ and ρ' are 0-prefix bisimilar and for each proper prefix ν of ρ (resp., proper prefix ν' of ρ'), there exists a proper prefix ν' of ρ' (resp., proper prefix ν of ρ) such that ν and ν' are $(h-1)$-prefix bisimilar

- *h*-prefix bisimilarity is an equivalence relation over the set of traces
- *h*-prefix bisimilarity propagates downwards

h-prefix bisimilarity $\Rightarrow$ h-equivalence

Proposition

Let $h \geq 0$, and ρ and ρ' be two h-prefix bisimilar traces of a Kripke structure $\mathcal{K}$. For each $\mathsf{A\overline{A}B\overline{BE}}$ formula ψ, with B-nesting of ψ less than or equal to h, it holds that

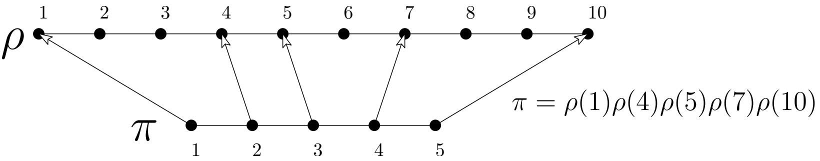
$$\mathcal{K}, \rho \models \psi \iff \mathcal{K}, \rho' \models \psi$$

Induced trace

Definition (Induced trace)

Let ρ be a trace of length n of a Kripke structure $\mathcal{K}$. A trace induced by ρ is a trace π of $\mathcal{K}$ such that there exists an increasing sequence of ρ-positions $i_1 < \ldots < i_k$, where $i_1 = 1$, $i_k = n$, and

$$\pi = \rho(i_1) \cdots \rho(i_k)$$



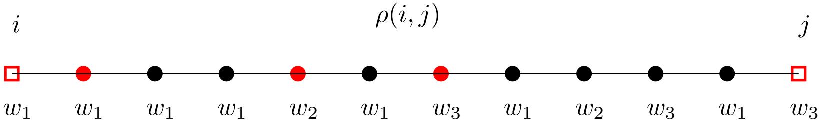
If π is induced by $\rho \Rightarrow \text{fst}(\pi) = \text{fst}(\rho)$, $\text{lst}(\pi) = \text{lst}(\rho)$, and $|\pi| \leq |\rho|$

Prefix-skeleton sampling

Definition (Prefix-skeleton sampling)

Let ρ be a trace of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$. Given two ρ-positions i and j, with $i \leq j$, the prefix-skeleton sampling of $\rho(i, j)$ is the minimal set P of ρ-positions in the interval $[i, j]$ satisfying:

- $i, j \in P$;
- for each state $w \in W$ occurring along $\rho(i+1, j-1)$, the minimal position $k \in [i+1, j-1]$ such that $\rho(k) = w$ is in P



$$P = \{i, i+1, i+4, i+6, j\}$$

h-prefix sampling

Definition (h-prefix sampling)

For each $h \geq 1$, the h-prefix sampling of ρ is the minimal set P_h of ρ-positions inductively satisfying the following conditions:

- for $h = 1$: P_1 is the prefix-skeleton sampling of ρ;
- for $h > 1$:
  - $P_h \supseteq P_{h-1}$ and
  - for all pairs of consecutive positions i, j in P_{h-1}, the prefix-skeleton sampling of $\rho(i, j)$ is in P_h

Proposition

The h-prefix sampling P_h of (any) ρ is such that $|P_h| \leq (|W| + 2)^h$

A small model (trace) result

Given a trace ρ, we can derive another trace ρ', induced by ρ and h-prefix bisimilar to ρ, such that $|\rho'| \leq (|W| + 2)^{h+2}$ as follows:

A small model (trace) result

Given a trace ρ, we can derive another trace ρ', induced by ρ and h-prefix bisimilar to ρ, such that $|\rho'| \leq (|W| + 2)^{h+2}$ as follows:

1. we first compute the $(h+1)$-prefix sampling P_{h+1} of ρ;

A small model (trace) result

Given a trace ρ, we can derive another trace ρ', induced by ρ and h-prefix bisimilar to ρ, such that $|\rho'| \leq (|W| + 2)^{h+2}$ as follows:

1. we first compute the $(h+1)$-prefix sampling P_{h+1} of ρ;
2. then, for all pairs of consecutive ρ-positions i, j in P_{h+1}, we consider a trace induced by $\rho(i, j)$, with no repeated occurrences of any state, except at most the first and last ones (hence no longer than $(|W| + 2)$);

A small model (trace) result

Given a trace ρ, we can derive another trace ρ', induced by ρ and h-prefix bisimilar to ρ, such that $|\rho'| \leq (|W| + 2)^{h+2}$ as follows:

1. we first compute the $(h+1)$-prefix sampling P_{h+1} of ρ;
2. then, for all pairs of consecutive ρ-positions i, j in P_{h+1}, we consider a trace induced by $\rho(i, j)$, with no repeated occurrences of any state, except at most the first and last ones (hence no longer than $(|W| + 2)$);
3. ρ' is just the ordered concatenation of all these traces

A small model (trace) result

Given a trace ρ, we can derive another trace ρ', induced by ρ and h-prefix bisimilar to ρ, such that $|\rho'| \leq (|W| + 2)^{h+2}$ as follows:

1. we first compute the $(h + 1)$-prefix sampling P_{h+1} of ρ;
2. then, for all pairs of consecutive ρ-positions i, j in P_{h+1}, we consider a trace induced by $\rho(i, j)$, with no repeated occurrences of any state, except at most the first and last ones (hence no longer than $(|W| + 2)$);
3. ρ' is just the ordered concatenation of all these traces

ρ and ρ' can be proved to be h-prefix bisimilar, and thus ρ' is indistinguishable from ρ with respect to the fulfilment of any formula ψ, with B-nesting of ψ (abbreviated $\mathrm{Nest_B}(\psi)$) $\leq h$

By the previous bound on $|P_h|$, it holds that $|\rho'| \leq (|W| + 2)^{h+2}$

An EXPSPACE model checking algorithm for $A\overline{A}B\overline{B}\overline{E}$

Algorithm 1 $\texttt{ModCheck}(\mathcal{K}, \psi)$

```
h ← Nest_B(ψ)
u ← New (Unravelling(K, w_0, h))                                  ◃ w_0 initial state of K
while u.hasMoreTracks() do
    ρ′ ← u.getNextTrack()
    if Check(K, h, ψ, ρ′) = 0 then return 0: "K, ρ′ ⊭ ψ"          ◃ Counterexample found ✗
return 1: "K ⊨ ψ"                                                  ◃ Model checking OK ✓
```


L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala, Interval Temporal Logic Model Checking Based on Track Bisimilarity and Prefix Sampling, ICTCS 2016

PSPACE-hardness of $A\overline{A}B\overline{B}\overline{E}$ model checking

PSPACE-hardness of the model checking problem for the fragment $\overline{B}$ (and thus for $A\overline{A}B\overline{B}\overline{E}$) can be proved by a reduction from the QBF problem

Theorem

The model checking problem for $\overline{B}$, and thus for $A\overline{A}B\overline{B}\overline{E}$, over finite Kripke structures is PSPACE-hard

$A\overline{A}B\overline{B}\overline{E}$ model checking is thus in between PSPACE and EXPSPACE (remind: BE model checking is EXPSPACE-hard)

A. Molinari, A. Montanari, A. Peron, and P. Sala, Model Checking Well-Behaved Fragments of HS: The (Almost) Final Picture, KR 2016

Epistemic HS (Lomuscio and Michaliszyn)

Distinctive feature of Epistemic HS (EHS for short): the labelling function is defined on the endpoints of the (finite) traces/intervals

Lomuscio and Michaliszyn proved that the local model checking problem (verification of a given specification against a single initial trace) for the fragment EHS[BE] is PSPACE-complete

If epistemic modalities are removed, it is in PTIME (notice that modalities $\langle B \rangle$ and $\langle E \rangle$ allow one to access only sub-intervals of the given initial one, whose number is quadratic in the length of it)

A. Lomuscio and J. Michaliszyn, An Epistemic Halpern-Shoham Logic, IJCAI 2013

Epistemic HS (Lomuscio and Michaliszyn) - cont'd

Later on, they showed that the picture drastically changes with other fragments of HS that allow one to access infinitely many traces

They proved that the model checking problem for the HS fragment $A\overline{B}$, extended with epistemic modalities, is decidable, with a non-elementary upper bound

Notice that formulas of this logic can possibly refer to infinitely many (future) traces

A. Lomuscio and J. Michaliszyn, Decidability of model checking multi-agent systems against a class of EHS specifications, ECAI 2014

Epistemic HS (Lomuscio and Michaliszyn) - cont'd

In their most recent contribution, Lomuscio and Michaliszyn generalized the labeling function by allowing it to be given by any regular expression on the states of intervals

Such a generalization results in a considerable increase in the expressiveness of the specifications at no computational cost in terms of the corresponding model checking problem

A. Lomuscio and J. Michaliszyn, Model Checking Multi-Agent Systems against Epistemic HS Specifications with Regular Expressions, KR 2016

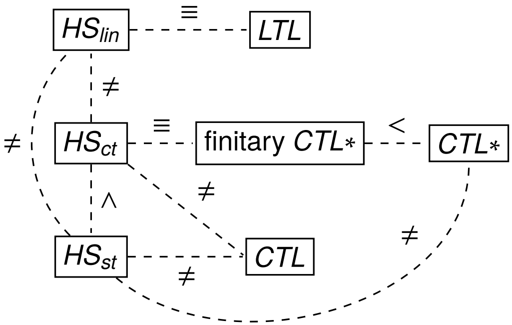
A comparison with LTL, CTL, and CTL* model checking

In order to compare the expressiveness of HS model checking with that of LTL, CTL, and CTL* model checking, we define alternative semantics for HS:

- trace semantics (HS_{lin}) - the infinite paths (computations) of the Kripke structure are the main semantic entities
- computation tree semantics (HS_{ct}) - the future is branching, but the past is linear (as well as finite and cumulative)

Trace (resp., computation tree) semantics allowed us to establish a bridge between HS model checking and LTL (resp., CTL/CTL*) model checking

The expressiveness picture



L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala, Interval vs. Point Temporal Logic Model Checking: an Expressiveness Comparison, FSTTCS 2016

Current and future research agenda

- To complete the picture of interval temporal logic model checking under the homogeneity assumption (and the HS_{st} state semantics)
- Planning as Model Checking in Interval Temporal Logic
- To remove the homogeneity assumption
- To replace finite Kripke structures with more complex ones (pushdown systems, other infinite state transition systems, systems based on timelines)

People

- **Bozzelli, Laura** — Universidad Politécnica de Madrid, Spain
- **Lomuscio Alessio** — Imperial College, UK
- **Michaliszyn Jakub** — University of Wroclaw, Poland
- **Molinari Alberto; Montanari, Angelo** — University of Udine, Italy
- **Murano Aniello, Peron Adriano** — University of Napoli, Italy
- **Perelli Giuseppe**— Oxford University, UK
- **Sala, Pietro** — University of Verona, Italy