

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

Laura Bozzelli

Technical University of Madrid (UPM), Madrid, Spain

laura.bozzelli@fi.upm.es

Alberto Molinari

University of Udine, Udine, Italy

molinari.alberto@gmail.com

Angelo Montanari

angelo.montanari@uniud.it

Adriano Peron

University of Napoli “Federico II”, Napoli, Italy

adrperon@unina.it

Pietro Sala

University of Verona, Verona, Italy

pietro.sala@univr.it

In the plethora of fragments of Halpern and Shoham’s modal logic of time intervals (HS), the logic AB of Allen’s relations *Meets* and *Started-by* is at a central position. Statements that may be true at certain intervals, but at no sub-interval of them, such as accomplishments, as well as metric constraints about the length of intervals, that force, for instance, an interval to be at least (resp., at most, exactly) k points long, can be expressed in AB. Moreover, over the linear order of the natural numbers \mathbb{N} , it subsumes the (point-based) logic LTL, as it can easily encode the next and until modalities. Finally, it is expressive enough to capture the ω -regular languages, that is, for each ω -regular expression R there exists an AB formula φ such that the language defined by R coincides with the set of models of φ over \mathbb{N} . It has been shown that the satisfiability problem for AB over \mathbb{N} is **EXPSpace**-complete. Here we prove that, under the homogeneity assumption, its model checking problem is $\Delta_2^P = \mathbf{P}^{\mathbf{NP}}$ -complete (for the sake of comparison, the model checking problem for full HS is **EXPSpace**-hard, and the only known decision procedure is nonelementary). Moreover, we show that the modality for the Allen relation *Met-by* can be added to AB at no extra cost ($\overline{\text{AAB}}$ is $\mathbf{P}^{\mathbf{NP}}$ -complete as well).

1 Introduction

In this paper, we investigate the model checking problem for the interval logic of Allen’s Relations *Meets* and *Started-by*. Given a model of a system (generally, a Kripke structure) and a temporal logic formula, which specifies the expected properties of the system, model checking algorithms verify, in fully automatic way, whether the model satisfies the formula; if this is not the case, they provide a counterexample, that is, a computation of the system failing to satisfy some property. Model checking has been successfully employed in formal verification as well as in various areas of AI, ranging from planning to configuration and multi-agent systems [8, 15].

Standard point-based temporal logics, such as LTL, CTL, and CTL* [7, 21], are commonly used as specification languages. Even though they turn out to be well-suited for a variety of application domains, there are relevant system properties, involving, for instance, actions with duration, accomplishments, and temporal aggregations, which are inherently “interval-based” and thus cannot be properly dealt with by temporal logics that allow one to predicate over computation states only. To overcome these limitations, one can resort to *interval temporal logics* (ITLs), that take intervals—instead of points—as their primitive entities [10], which have been successfully applied in various areas of computer science and AI, including hardware and software verification, computational linguistics, and planning [12, 20, 22, 25].

ITL model checking is the context of this paper. In order to check interval properties of computations, one needs to collect information about states into computation stretches: each finite path of a Kripke

structure is interpreted as an interval, whose labelling is defined on the basis of the labelling of the component states. Among ITLs, Halpern and Shoham’s modal logic of time intervals HS [10] is the main reference. It features one modality for each possible ordering relation between a pair of intervals apart from equality (the so-called Allen’s relations [1]). The *satisfiability problem* for HS has been thoroughly studied, and it turns out to be highly undecidable for all relevant (classes of) linear orders [10]. The same holds for most HS fragments [4]; however, some meaningful exceptions exist, including the logic of temporal neighbourhood $A\bar{A}$ and the logic of sub-intervals D [5, 6]. The *model checking problem* for HS has entered the research agenda only recently [2, 12, 13, 14, 16, 17, 18, 19]. In [16], Molinari et al. deal with model checking for full HS over Kripke structures under the homogeneity assumption [23], showing its non-elementary decidability by means of a suitable small model theorem (**EXPSPACE**-hardness has been proved in [2]). Since then, the attention was brought to HS fragments, which, similarly to what happens with satisfiability, are often computationally better.

In this paper we first prove that model checking for the logic $A\bar{A}B$ (resp., $A\bar{A}E$) of Allen’s relations *Meets*, *Met-by*, and *Started-by* (resp., *Finished-by*) is in $\mathbf{P}^{\mathbf{NP}}$; then we prove that its fragment AB (resp., $\bar{A}E$) is $\mathbf{P}^{\mathbf{NP}}$ -hard; finally we show that its fragment $\bar{A}B$ (resp., AE) belongs to $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ and it is $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hard. $\mathbf{P}^{\mathbf{NP}}$ (also denoted as Δ_2^p) is the class of problems decided by a deterministic polynomial time Turing machine that queries an NP oracle. The classes $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ are analogous, but the number of queries is bounded by $O(\log n)$ and $O(\log^2 n)$, respectively, being n the input size [9, 24]. These three classes are higher than both \mathbf{NP} and $\mathbf{co-NP}$ in the *polynomial time hierarchy*, and closed under complement. It is worth noticing that, whereas we know many natural problems which are complete for Σ_2^p or Π_2^p (in general, for Σ_k^p and Π_k^p , with $k \geq 2$), the classes $\mathbf{P}^{\mathbf{NP}}$, $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$, and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ are not so “populated” (and neither are the classes Δ_k^p , for $k > 2$). Among the few natural problems complete for $\mathbf{P}^{\mathbf{NP}}$, we would like to mention model checking for several fragments of CTL*, including CTL⁺, ECTL⁺, and FCTL [11]. As for the other two classes, very recently Molinari et al. have shown that model checking A , \bar{A} , or $A\bar{A}$ formulas is in $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ and hard for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ [19].

Related work. In [12, 13], Lomuscio and Michaliszyn address the model checking problem for some fragments of HS extended with epistemic modalities. Their semantic assumptions considerably differ from those made in [16], making it difficult to compare the outcomes of the two research lines. Moreover, they consider a restricted form of model checking, which verifies a specification against a single (finite) initial computation interval (this is in general a limitation, unless some operators of HS are available, such as $\langle A \rangle$): their goal is indeed to reason about a given computation of a multi-agent system, rather than on all its admissible computations. Recently they have shown how to exploit regular expressions in order to specify the way in which the intervals of a Kripke structure get labelled [14]. Such an extension leads to a significant increase in the expressiveness of HS formulas.

Organization of the paper. In the next section we introduce the fundamental elements of the model checking problem for HS and its fragments. Then, in Section 3, we provide a $\mathbf{P}^{\mathbf{NP}}$ model checking algorithm for $A\bar{A}B$ (and $A\bar{A}E$) formulas. In Section 4 we prove the $\mathbf{P}^{\mathbf{NP}}$ -hardness of model checking for AB (and $\bar{A}E$). $\mathbf{P}^{\mathbf{NP}}$ -completeness of AB , $\bar{A}E$, $A\bar{A}B$ and $A\bar{A}E$ follows. Finally we show that the problem for formulas of $\bar{A}B$ and AE is in $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ and hard for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$.

2 Preliminaries

The interval temporal logic HS. An interval algebra to reason about intervals and their relative order was proposed by Allen in [1], while a systematic logical study of interval representation and reasoning

Table 1: Allen's relations and corresponding HS modalities.

Allen relation	HS	Definition w.r.t. interval structures	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	$\langle O \rangle$	$[x, y] \mathcal{R}_O [v, z] \iff x < v < y < z$	

was done a few years later by Halpern and Shoham, who introduced the interval temporal logic HS featuring one modality for each Allen relation, but equality [10]. Table 1 depicts 6 of the 13 Allen's relations, together with the corresponding HS (existential) modalities. The other 7 relations are the 6 inverses (given a binary relation \mathcal{R} , the inverse $\bar{\mathcal{R}}$ is such that $b\bar{\mathcal{R}}a$ if and only if $a\mathcal{R}b$) and equality.

The HS language consists of a set of proposition letters \mathcal{AP} , the Boolean connectives \neg and \wedge , and a temporal modality for each of the (non trivial) Allen's relations, i.e., $\langle A \rangle$, $\langle L \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle D \rangle$, $\langle O \rangle$, $\langle \bar{A} \rangle$, $\langle \bar{L} \rangle$, $\langle \bar{B} \rangle$, $\langle \bar{E} \rangle$, $\langle \bar{D} \rangle$, and $\langle \bar{O} \rangle$. HS formulas are defined by the grammar $\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle \psi \mid \langle \bar{X} \rangle \psi$, where $p \in \mathcal{AP}$ and $X \in \{A, L, B, E, D, O\}$. In the following, we shall also exploit as abbreviations the standard logical connectives for disjunction \vee , implication \rightarrow , and double implication \leftrightarrow . Furthermore, for any modality X , the dual universal modalities $[X]\psi$ and $[\bar{X}]\psi$ are defined as $\neg\langle X \rangle\neg\psi$ and $\neg\langle \bar{X} \rangle\neg\psi$, respectively. Finally, given any subset of Allen's relations $\{X_1, \dots, X_n\}$, we denote by $X_1 \cdots X_n$ the HS fragment featuring existential (and universal) modalities for X_1, \dots, X_n only.

W.l.o.g., we assume the *non-strict semantics of HS*, which admits intervals consisting of a single point¹. Under such an assumption, all HS modalities can be expressed in terms of modalities $\langle B \rangle$, $\langle E \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$ [10]. HS can thus be seen as a multi-modal logic with these 4 primitive modalities and its semantics can be defined over a multi-modal Kripke structure, called *abstract interval model*, where intervals are treated as atomic objects and Allen's relations as binary relations between pairs of intervals. Since later we will focus on some HS fragments excluding $\langle \bar{B} \rangle$ and $\langle \bar{E} \rangle$, we add both $\langle A \rangle$ and $\langle \bar{A} \rangle$ to the considered set of HS modalities.

Definition 1. [16] An abstract interval model is a tuple $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where \mathcal{AP} is a set of proposition letters, \mathbb{I} is a possibly infinite set of atomic objects (worlds), $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are three binary relations over \mathbb{I} , and $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is a (total) labeling function, which assigns a set of proposition letters to each world.

In the interval setting, \mathbb{I} is interpreted as a set of intervals and $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ as Allen's relations A (meets), B (started-by), and E (finished-by), respectively; σ assigns to each interval in \mathbb{I} the set of proposition letters that hold over it.

Given an abstract interval model $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$ and an interval $I \in \mathbb{I}$, the truth of an HS formula over I is inductively defined as follows:

- $\mathcal{A}, I \models p$ iff $p \in \sigma(I)$, for any $p \in \mathcal{AP}$;
- $\mathcal{A}, I \models \neg\psi$ iff it is not true that $\mathcal{A}, I \models \psi$ (also denoted as $\mathcal{A}, I \not\models \psi$);
- $\mathcal{A}, I \models \psi \wedge \phi$ iff $\mathcal{A}, I \models \psi$ and $\mathcal{A}, I \models \phi$;
- $\mathcal{A}, I \models \langle X \rangle \psi$, for $X \in \{A, B, E\}$, iff there exists $J \in \mathbb{I}$ such that $IX_{\mathbb{I}}J$ and $\mathcal{A}, J \models \psi$;
- $\mathcal{A}, I \models \langle \bar{X} \rangle \psi$, for $\bar{X} \in \{\bar{A}, \bar{B}, \bar{E}\}$, iff there exists $J \in \mathbb{I}$ such that $JX_{\mathbb{I}}I$ and $\mathcal{A}, J \models \psi$.

¹All the results we prove in the paper hold for the strict semantics as well.

Kripke structures and abstract interval models. In model checking, finite state systems are usually modelled as Kripke structures. In [16], the authors define a mapping from Kripke structures to abstract interval models, that allows one to specify interval properties of computations by means of HS formulas.

Definition 2. A finite Kripke structure is a tuple $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$, where \mathcal{AP} is a set of proposition letters, W is a finite set of states, $\delta \subseteq W \times W$ is a left-total relation between pairs of states, $\mu : W \mapsto 2^{\mathcal{AP}}$ is a total labelling function, and $w_0 \in W$ is the initial state.

For all $w \in W$, $\mu(w)$ is the set of proposition letters that hold at w , while δ is the transition relation that describes the evolution of the system over time.

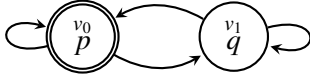


Figure 1: The Kripke structure \mathcal{K}_2 .

Figure 1 depicts the finite Kripke structure $\mathcal{K}_2 = (\{p, q\}, \{v_0, v_1\}, \delta, \mu, v_0)$, where $\delta = \{(v_0, v_0), (v_0, v_1), (v_1, v_0), (v_1, v_1)\}$, $\mu(v_0) = \{p\}$, and $\mu(v_1) = \{q\}$. The initial state v_0 is identified by a double circle.

Definition 3. A track ρ over a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is a finite sequence of states $v_1 \cdots v_n$, with $n \geq 1$, such that $(v_i, v_{i+1}) \in \delta$ for $i = 1, \dots, n-1$.

Let $\text{Trk}_{\mathcal{K}}$ be the (infinite) set of all tracks over a finite Kripke structure \mathcal{K} . For any track $\rho = v_1 \cdots v_n \in \text{Trk}_{\mathcal{K}}$, we define:

- $|\rho| = n$, $\text{fst}(\rho) = v_1$, $\text{lst}(\rho) = v_n$, and for $1 \leq i \leq |\rho|$, $\rho(i) = v_i$;
- $\text{states}(\rho) = \{v_1, \dots, v_n\} \subseteq W$;
- $\rho(i, j) = v_i \cdots v_j$, with $1 \leq i \leq j \leq |\rho|$, is the subtrack of ρ bounded by i and j ;
- $\text{Pref}(\rho) = \{\rho(1, i) \mid 1 \leq i \leq |\rho| - 1\}$ and $\text{Suff}(\rho) = \{\rho(i, |\rho|) \mid 2 \leq i \leq |\rho|\}$ are the sets of all proper prefixes and suffixes of ρ , respectively.

Finally, if $\text{fst}(\rho) = w_0$ (the initial state of \mathcal{K}), ρ is called an *initial track*.

An abstract interval model (over $\text{Trk}_{\mathcal{K}}$) can be naturally associated with a finite Kripke structure \mathcal{K} by considering the set of intervals as the set of tracks of \mathcal{K} . Since \mathcal{K} has loops (δ is left-total), the number of tracks in $\text{Trk}_{\mathcal{K}}$, and thus the number of intervals, is infinite.

Definition 4. The abstract interval model induced by a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where $\mathbb{I} = \text{Trk}_{\mathcal{K}}$, $A_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \text{lst}(\rho) = \text{fst}(\rho')\}$, $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Pref}(\rho)\}$, $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Suff}(\rho)\}$, and $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is such that $\sigma(\rho) = \bigcap_{w \in \text{states}(\rho)} \mu(w)$, for all $\rho \in \mathbb{I}$.

Relations $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as the Allen's relations A, B , and E , respectively. Moreover, according to the definition of σ , $p \in \mathcal{AP}$ holds over $\rho = v_1 \cdots v_n$ iff it holds over all the states v_1, \dots, v_n of ρ . This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval if and only if it holds over all its subintervals [23].

Definition 5. Let \mathcal{K} be a finite Kripke structure and ψ be an HS formula; we say that a track $\rho \in \text{Trk}_{\mathcal{K}}$ satisfies ψ , denoted as $\mathcal{K}, \rho \models \psi$, iff it holds that $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$. Moreover, we say that \mathcal{K} models ψ , denoted as $\mathcal{K} \models \psi$, iff for all initial tracks $\rho' \in \text{Trk}_{\mathcal{K}}$ it holds that $\mathcal{K}, \rho' \models \psi$. The model checking problem for HS over finite Kripke structures is the problem of deciding whether $\mathcal{K} \models \psi$.

We conclude with a simple example (a simplified version of the one given in [16]), showing that the fragments investigated in this paper can express meaningful properties of state transition systems.

In Figure 2, we provide an example of a finite Kripke structure $\mathcal{K}_{\text{Sched}}$ that models the behaviour of a scheduler serving three processes which are continuously requesting the use of a common resource. The initial state is v_0 : no process is served in that state. In the states v_i and \bar{v}_i , with $i \in \{1, 2, 3\}$, the i -th process is served (this is denoted by the fact that p_i holds in those states). For the sake of readability,

edges are marked either by r_i , for *request*(i), or by u_i , for *unlock*(i). Edge labels do not have a semantic value, that is, they are neither part of the structure definition, nor proposition letters; they are simply used to ease reference to edges. Process i is served in state v_i , then, after “some time”, a transition u_i from v_i to \bar{v}_i is taken; subsequently, process i cannot be served again immediately, as v_i is not directly reachable from \bar{v}_i (the scheduler cannot serve the same process twice in two successive rounds). A transition r_j , with $j \neq i$, from \bar{v}_i to v_j is then taken and process j is served. This structure can easily be generalised to a higher number of processes.

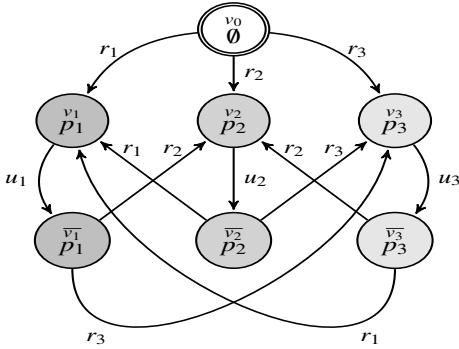


Figure 2: The Kripke structure \mathcal{K}_{Sched} .

requires that, in any suffix of length at least 11 of an initial track, process 3 is executed at least once in some internal states (*non starvation*). \mathcal{K}_{Sched} does not satisfy it, because the scheduler can postpone the execution of a process ad libitum: $\mathcal{K}_{Sched} \not\models [E](\langle E \rangle^{10} \top \rightarrow \langle E \rangle \langle \bar{A} \rangle p_3)$. The third formula requires that, in any suffix of length at least 6 of an initial track, p_1, p_2 , and p_3 are all witnessed. The only way to satisfy this property would be to force the scheduler to execute the three processes in a strictly periodic manner (*strict alternation*), that is, $p_i p_j p_k p_i p_j p_k p_i p_j p_k \dots$, for $i, j, k \in \{1, 2, 3\}$ and $i \neq j \neq k \neq i$, but \mathcal{K}_{Sched} does not meet such a requirement: $\mathcal{K}_{Sched} \not\models [E](\langle E \rangle^5 \rightarrow (\langle E \rangle \langle \bar{A} \rangle p_1 \wedge \langle E \rangle \langle \bar{A} \rangle p_2 \wedge \langle E \rangle \langle \bar{A} \rangle p_3))$.

The general picture. We now describe known and new complexity results about the model checking problem for HS fragments (see Figure 3 for a graphical account).

In [16], Molinari et al. have shown that, given a Kripke structure \mathcal{K} and a bound k on the structural complexity of HS formulas, i.e., on the nesting depth of $\langle E \rangle$ and $\langle B \rangle$ modalities, it is possible to obtain a *finite* representation for $\mathcal{A}_{\mathcal{K}}$, which is equivalent to $\mathcal{A}_{\mathcal{K}}$ with respect to satisfiability of HS formulas with structural complexity less than or equal to k . Then, by exploiting such a representation, they proved that the model checking problem for (full) HS is decidable, providing an algorithm with non-elementary complexity. In [2], **EXPSpace**-hardness of the fragment BE, and thus of full HS, has been shown.

The fragments $\overline{A\bar{A}B\bar{B}E}$ and $\overline{A\bar{A}E\bar{B}E}$ have been systematically studied in [18]. For each of them, an **EXPSpace** model checking algorithm has been devised that, for any track of the Kripke structure, finds a satisfiability-preserving track of bounded length (*track representative*). In this way, the model checking algorithm needs to check only tracks with a bounded maximum length. **PSPACE**-hardness of the model checking problem for $\overline{A\bar{A}B\bar{B}E}$ and $\overline{A\bar{A}E\bar{B}E}$ has been proved in [17] (if a succinct encoding of formulas is exploited, the algorithm remains in **EXPSpace**, but a **NEXPTIME** lower bound can be given [18]). Finally, it has been shown that formulas satisfying a constant bound on the nesting depth of $\langle B \rangle$ (respectively, $\langle E \rangle$) can be checked in polynomial working space [18].

Some well-behaved HS fragments, namely, $\overline{A\bar{A}B\bar{E}}$, \bar{B} , \bar{E} , $\overline{A\bar{A}}$, A , and \bar{A} , which are still expressive enough to capture meaningful interval properties of state transition systems and whose model checking problem has a computational complexity markedly lower than that of full HS, have been identified in

We now show how some meaningful properties to be checked over \mathcal{K}_{Sched} can be expressed in the HS fragment $\overline{A\bar{E}}$. In all the following formulas, we force the validity of the considered properties over all legal computation sub-intervals by using the modality $[E]$ (all computation sub-intervals are suffixes of at least one initial track of the Kripke structure). The first formula requires that at least 2 proposition letters are witnessed in any suffix of length at least 4 of an initial track. Since a process cannot be executed twice in a row, it is satisfied by \mathcal{K}_{Sched} : $\mathcal{K}_{Sched} \models [E](\langle E \rangle^3 \top \rightarrow (\chi(p_1, p_2) \vee \chi(p_1, p_3) \vee \chi(p_2, p_3)))$ where $\chi(p, q) = \langle E \rangle \langle \bar{A} \rangle p \wedge \langle E \rangle \langle \bar{A} \rangle q$. The second formula

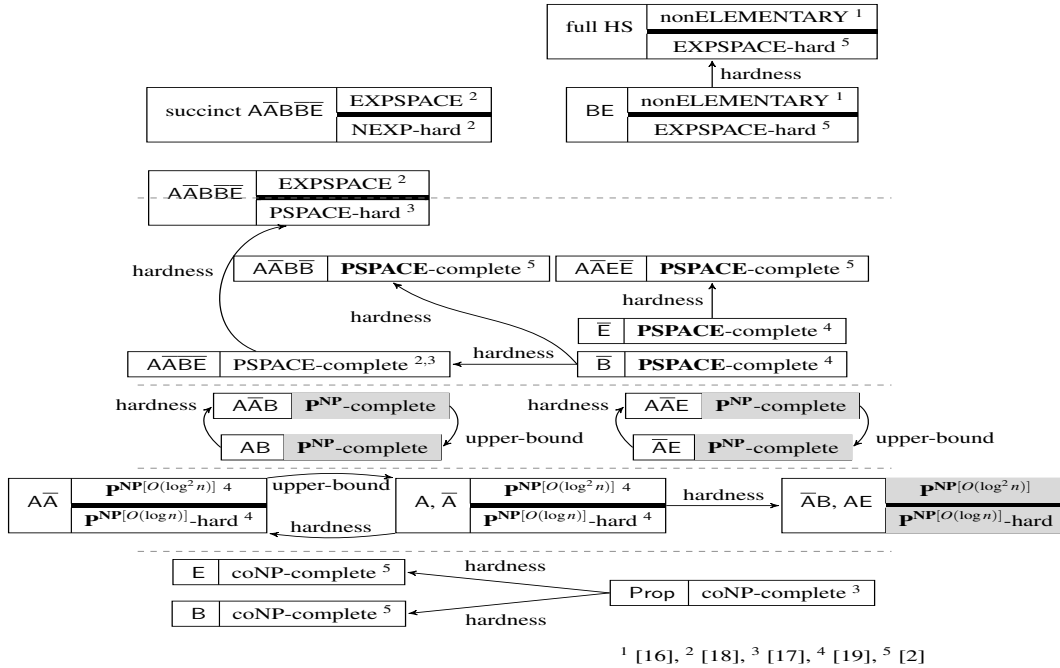


Figure 3: Complexity of the model checking problem for HS fragments: known results are depicted in white boxes, new ones in gray boxes.

[17, 19]. In particular the authors proved that the problem is **PSPACE**-complete for the fragments \overline{AABE} , \overline{B} , and \overline{E} , and in between $\mathbf{PNP}^{[O(\log n)]}$ and $\mathbf{PNP}^{[O(\log^2 n)]}$ [9, 24] for \overline{AA} , \overline{A} , and \overline{A} . Two other well-behaved fragments, namely, \overline{AABB} and \overline{AAEE} , have been investigated in [2], showing that their model checking problem is **PSPACE**-complete. In addition, the authors showed that \overline{B} and \overline{E} are **co-NP**-complete (the same complexity as the model checking problem for the purely propositional HS fragment $\overline{\text{Prop}}$ [17]).

In this paper, we complete the analysis of the sub-fragments of \overline{AABB} (resp., \overline{AAEE}). In Section 3, we devise a \mathbf{PNP} model checking algorithm for \overline{AAB} (resp., \overline{AAE}). Then, in Section 4, we prove that \overline{AB} (resp., \overline{AE}) is hard for \mathbf{PNP} . It immediately follows that model checking for \overline{AB} and \overline{AAB} (resp., \overline{AE} and \overline{AAE}) formulas over finite Kripke structures is \mathbf{PNP} -complete. Finally, we show that \overline{AB} (resp., \overline{AE}) are in $\mathbf{PNP}^{[O(\log^2 n)]}$ (the proof is reported in [3]) and hard for $\mathbf{PNP}^{[O(\log n)]}$ (the hardness follows from that of \overline{A} , resp., \overline{A} [19]).

It is worth pointing out that the fragment \overline{AB} belongs to a lower complexity class than the fragment \overline{AB} (the same for the symmetric fragments \overline{AE} and \overline{AE}). Such a difference can be explained as follows.

Let us consider a formula $\langle \overline{B} \rangle \langle \overline{A} \rangle \theta$, which belongs to \overline{AB} . A track ρ satisfies $\langle \overline{B} \rangle \langle \overline{A} \rangle \theta$ if there exists a prefix $\tilde{\rho}$ of ρ from which a branch satisfying θ departs, i.e., a track starting from $\text{fst}(\tilde{\rho})$. This amounts to say that \overline{AB} allows one to impose specific constraints on the branches departing from a state occurring in a given path. Such an ability will be exploited in Section 4 to prove the \mathbf{PNP} -hardness of \overline{AB} .

Conversely, the fragment \overline{AB} cannot express constraints of this form. For any given track ρ , modality $\langle \overline{A} \rangle$ only allows one to constrain tracks leading to the first state of ρ . As for modality $\langle \overline{B} \rangle$, if we consider a prefix $\tilde{\rho}$ of ρ , the set of tracks leading to its first state is exactly the same as the set of those leading to the first state of ρ , as $\text{fst}(\tilde{\rho}) = \text{fst}(\rho)$. Therefore, pairing $\langle \overline{A} \rangle$ and $\langle \overline{B} \rangle$ does not give any advantage in terms of expressiveness. Such a weakness of \overline{AB} represents the reason why \overline{AB} formulas can be checked in time $\mathbf{PNP}^{[O(\log^2 n)]}$, instead of time \mathbf{PNP} .

Algorithm 1 $\text{MC}(\mathcal{X}, \psi, \text{DIRECTION})$

```

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

```

3 A $\mathbf{P}^{\mathbf{NP}}$ model checking algorithm for $A\bar{A}B$ formulas

In this section, we present a model checking algorithm for $A\bar{A}B$ formulas (Algorithm 1) belonging to the complexity class $\mathbf{P}^{\mathbf{NP}}$. We recall that $\mathbf{P}^{\mathbf{NP}}$ is the class of problems solvable in (deterministic) polynomial time exploiting an oracle for an \mathbf{NP} -complete problem. W.l.o.g., we restrict our attention to $A\bar{A}B$ formulas devoid of occurrences of conjunctions and universal modalities (definable, as usual, in terms of disjunctions, negations, and existential modalities).

Algorithm 1 presents the model checking procedure for a formula ψ against a Kripke structure \mathcal{X} . It exploits two global vectors, V_A and $V_{\bar{A}}$, which can be seen as the tabular representations of two Boolean functions taking as arguments a subformula ϕ of ψ and a state v of \mathcal{X} . The intuition is that the function $V_A(\phi, v)$ (resp., $V_{\bar{A}}(\phi, v)$) returns \top if and only if there exists a track $\rho \in \text{Trk}_{\mathcal{X}}$ starting from the state v (resp., leading to the state v) such that $\mathcal{X}, \rho \models \phi$. The procedure MC is initially invoked with parameters $(\mathcal{X}, \neg\psi, \text{FORWARD})$. During the execution, it instantiates the entries of V_A and $V_{\bar{A}}$, which are exploited in order to answer the model checking problem $\mathcal{X} \models \psi$; this is, in the end, equivalent to checking whether $V_A(\neg\psi, w_0) = \perp$, where w_0 is the initial state of \mathcal{X} .

Let us consider the model checking procedure MC in more detail. Besides the Kripke structure \mathcal{X} and the formula ψ , MC features a third parameter, DIRECTION , which can be assigned the value FORWARD (resp., BACKWARD), that is used in combination with the modality $\langle A \rangle$ (resp., $\langle \bar{A} \rangle$) for a forward (resp., backward) unravelling of \mathcal{X} . MC is applied recursively on the nesting of modalities $\langle A \rangle$ and $\langle \bar{A} \rangle$ in the formula ψ (in the base case, ψ features no occurrences of $\langle A \rangle$ or $\langle \bar{A} \rangle$). In order to instantiate the Boolean vectors V_A and $V_{\bar{A}}$, an oracle is invoked (lines 5–9) for each state v of the Kripke structure. Such an invocation is syntactically represented by $\text{Success}(\text{Oracle}(\mathcal{X}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}))$, and it returns \top whenever there exists a computation of the non-deterministic algorithm $\text{Oracle}(\mathcal{X}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}})$ returning \top , namely, whenever there is a suitable track starting from, or leading to v (depending on the value of the parameter DIRECTION), and satisfying ψ .

We now introduce the notion of $A\bar{A}$ -modal subformulas of ψ ; these subformulas “direct” the recursive calls of MC .

Definition 6. *The set of $A\bar{A}$ -modal subformulas of an $A\bar{A}B$ formula ψ , denoted by $\text{ModSubf}_{A\bar{A}}(\psi)$, is the set of subformulas of ψ having either the form $\langle A \rangle \psi'$ or the form $\langle \bar{A} \rangle \psi'$, for some ψ' , which are not in the scope of any $\langle A \rangle$ or $\langle \bar{A} \rangle$ modality.*

For instance, $\text{ModSubf}_{A\bar{A}}(\langle A \rangle \langle \bar{A} \rangle q) = \{\langle A \rangle \langle \bar{A} \rangle q\}$ and $\text{ModSubf}_{A\bar{A}}((\langle A \rangle p \wedge \langle A \rangle \langle \bar{A} \rangle q) \rightarrow \langle A \rangle p) = \{\langle A \rangle p, \langle A \rangle \langle \bar{A} \rangle q\}$.

MC is recursively called on each formula ϕ such that $\langle A \rangle \phi$ or $\langle \bar{A} \rangle \phi$ belongs to $\text{ModSubf}_{A\bar{A}}(\psi)$ (lines 1–4). In this way, we can recursively gather in the Boolean vectors V_A and $V_{\bar{A}}$, by increasing nesting

Algorithm 2 Oracle($\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}$)

```

1:  $\tilde{\rho} \leftarrow \text{A\_track}(\mathcal{K}, v, |W| \cdot (2|\psi| + 1)^2, \text{DIRECTION}) \quad \triangleleft a$ 
   track of  $\mathcal{K}$  from/to  $v$  of length  $\leq |W| \cdot (2|\psi| + 1)^2$ 
2: for all  $\langle A \rangle \phi \in \text{ModSubf}_{A\bar{A}}(\psi)$  do
3:   for  $i = 1, \dots, |\tilde{\rho}|$  do
4:      $T[\langle A \rangle \phi, i] \leftarrow V_A(\phi, \tilde{\rho}(i))$ 
5: for all  $\langle \bar{A} \rangle \phi \in \text{ModSubf}_{\bar{A}A}(\psi)$  do
6:   for  $i = 1, \dots, |\tilde{\rho}|$  do
7:      $T[\langle \bar{A} \rangle \phi, i] \leftarrow V_{\bar{A}}(\phi, \text{fst}(\tilde{\rho}))$ 
8: for all subformulas  $\phi$  of  $\psi$ , not contained in (or equal to)
    $A\bar{A}$ -modal subformulas of  $\psi$ , by increasing length do
9:   if  $\phi = p$ , for  $p \in \mathcal{AP}$  then
10:     $T[p, 1] \leftarrow p \in \mu(\text{fst}(\tilde{\rho}))$ 
11:   for  $i = 2, \dots, |\tilde{\rho}|$  do
12:     $T[p, i] \leftarrow T[p, i-1]$  and  $p \in \mu(\tilde{\rho}(i))$ 
13:   else if  $\phi = \neg\phi_1$  then
14:     for  $i = 1, \dots, |\tilde{\rho}|$  do
15:        $T[\phi, i] \leftarrow \text{not } T[\phi_1, i]$ 
16:   else if  $\phi = \phi_1 \vee \phi_2$  then
17:     for  $i = 1, \dots, |\tilde{\rho}|$  do
18:        $T[\phi, i] \leftarrow T[\phi_1, i]$  or  $T[\phi_2, i]$ 
19:   else if  $\phi = \langle B \rangle \phi_1$  then
20:      $T[\phi, 1] \leftarrow \perp$ 
21:     for  $i = 2, \dots, |\tilde{\rho}|$  do
22:        $T[\phi, i] \leftarrow T[\phi, i-1]$  or  $T[\phi_1, i-1]$ 
23: return  $T[\psi, |\tilde{\rho}|]$ 

```

depth of the modalities $\langle A \rangle$ and $\langle \bar{A} \rangle$, the oracle answers for all the formulas ψ' such that $\langle A \rangle \psi'$ or $\langle \bar{A} \rangle \psi'$ is a subformula (be it maximal or not) of ψ .

Let us now consider the *non-deterministic polynomial time* procedure Oracle($\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}$) reported in Algorithm 2, which is used as the basic engine by the oracle in the aforementioned model checking Algorithm 1. The idea underlying Algorithm 2 is first to non-deterministically generate a track $\tilde{\rho}$ by unravelling the Kripke structure \mathcal{K} according to the parameter DIRECTION, and then to verify ψ over $\tilde{\rho}$. Such a procedure actually exploits a result proved in [2] (see, in particular, Theorem 10) stating a so-called “polynomial-size model-track property” for formulas of the fragment $A\bar{A}\bar{E}\bar{E}$: if ρ is a track of \mathcal{K} , ϕ is an $A\bar{A}\bar{E}\bar{E}$ formula, and $\mathcal{K}, \rho \models \phi$, then there exists $\rho' \in \text{Trk}_{\mathcal{K}}$ such that $|\rho'| \leq |W| \cdot (2|\phi| + 1)^2$, $\text{fst}(\rho) = \text{fst}(\rho')$, $\text{lst}(\rho) = \text{lst}(\rho')$, and $\mathcal{K}, \rho' \models \phi$. This property guarantees that, in order to check the satisfiability of a formula ϕ , it is enough to consider tracks having a length bounded by $|W| \cdot (2|\phi| + 1)^2$. Such a result holds by symmetry for formulas of the fragment $A\bar{A}B\bar{B}$ as well.

An execution of Oracle($\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}$) starts (line 1) by *non-deterministically* generating a track $\tilde{\rho}$ (having a length of at most $|W| \cdot (2|\psi| + 1)^2$), with v as its first (resp., last) state if the DIRECTION parameter is FORWARD (resp., BACKWARD). The track is generated by visiting the unravelling of \mathcal{K} (resp., of \mathcal{K} with transposed edges). The remaining part of the algorithm checks *deterministically* whether $\mathcal{K}, \tilde{\rho} \models \psi$. Such a verification is performed in a bottom-up way: for all the subformulas ϕ of ψ (starting from the minimal ones) and for all the prefixes $\tilde{\rho}(1, i)$ of $\tilde{\rho}$, with $1 \leq i \leq |\tilde{\rho}|$ (starting from the shorter ones), the procedure establishes whether $\mathcal{K}, \tilde{\rho}(1, i) \models \phi$ or not, and this result is stored in the entry $T[\phi, i]$ of a Boolean table T . Note that if the considered subformula of ψ is an element of $\text{ModSubf}_{A\bar{A}}(\psi)$, the algorithm does not need to perform any verification, since the result is already available in the Boolean vectors V_A and $V_{\bar{A}}$ (as a consequence of the previously completed calls to the procedure Oracle), and the table T is updated accordingly (lines 2–7). For the remaining subformulas, the entries of T are computed, as we already said, in a bottom-up fashion (lines 8–22). The result of the overall verification is stored in $T[\psi, |\tilde{\rho}|]$ and returned (line 23).

The algorithm presented here for checking formulas of $A\bar{A}B$ can trivially be adapted to check formulas of the symmetric fragment $A\bar{A}\bar{E}$.

The following lemma establishes the soundness and completeness of the procedure Oracle.

Lemma 1. *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, ψ be an $A\bar{A}B$ formula, and $V_A(\bullet, \bullet)$ and $V_{\bar{A}}(\bullet, \bullet)$ be two Boolean arrays. Let us assume that*

1. *for each $\langle A \rangle \phi \in \text{ModSubf}_{A\bar{A}}(\psi)$ and $v' \in W$, $V_A(\phi, v') = \top$ iff there exists $\rho \in \text{Trk}_{\mathcal{K}}$ such that*

- $\text{fst}(\rho) = v'$ and $\mathcal{X}, \rho \models \phi$, and
2. for each $\langle \bar{A} \rangle \phi \in \text{ModSubf}_{\bar{A}\bar{A}}(\psi)$ and $v' \in W$, $V_{\bar{A}}(\phi, v') = \top$ iff there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{lst}(\rho) = v'$ and $\mathcal{X}, \rho \models \phi$.

Then, $\text{Oracle}(\mathcal{X}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}})$ features a successful computation (returning \top) iff:

- there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$, in the case DIRECTION is FORWARD;
- there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{lst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$, in the case DIRECTION is BACKWARD.

Proof. It is easy to check that if $\tilde{\rho}$ is the track non-deterministically generated by A_track at line 1, then, for $i = 1, \dots, |\tilde{\rho}|$, it holds that $\mathcal{X}, \tilde{\rho}(1, i) \models \phi \iff T[\phi, i] = \top$, either by hypothesis, when ϕ occurs in $\text{ModSubf}_{\bar{A}\bar{A}}(\psi)$ (lines 2–7), or by construction, when ϕ does not occur in $\text{ModSubf}_{\bar{A}\bar{A}}(\psi)$ (lines 8–22).

Let us now assume that the value of the parameter DIRECTION is FORWARD (the proof for the other direction is analogous).

- (\Rightarrow) If $\text{Oracle}(\mathcal{X}, \psi, v, \text{FORWARD}, V_A \cup V_{\bar{A}})$ features a successful computation, it means that there exists a track $\tilde{\rho} \in \text{Trk}_{\mathcal{X}}$ (generated at line 1) such that $\text{fst}(\tilde{\rho}) = v$ and $T[\psi, |\tilde{\rho}|] = \top$. Hence $\mathcal{X}, \tilde{\rho} \models \psi$.
- (\Leftarrow) If there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$, as a result of Theorem 10 of [2], there exists $\tilde{\rho} \in \text{Trk}_{\mathcal{X}}$ such that $\mathcal{X}, \tilde{\rho} \models \psi$, $\text{fst}(\tilde{\rho}) = \text{fst}(\rho)$, and $|\tilde{\rho}| \leq |W| \cdot (2|\psi| + 1)^2$. It follows that in some non-deterministic instance of $\text{Oracle}(\mathcal{X}, \psi, v, \text{FORWARD}, V_A \cup V_{\bar{A}})$, $\text{A_track}(\mathcal{X}, v, |W| \cdot (2|\psi| + 1)^2, \text{FORWARD})$ returns such $\tilde{\rho}$ (at line 1). Finally, we have that $T[\psi, |\tilde{\rho}|] = \top$ as $\mathcal{X}, \tilde{\rho} \models \psi$, hence the considered instance of $\text{Oracle}(\mathcal{X}, \psi, v, \text{FORWARD}, V_A \cup V_{\bar{A}})$ is successful. \square

The following theorem states soundness and completeness of the model checking procedure MC.

Theorem 1. *Let $\mathcal{X} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, ψ be an $\text{A}\bar{\text{A}}\text{B}$ formula, and $V_A(\bullet, \bullet)$ and $V_{\bar{A}}(\bullet, \bullet)$ be two Boolean arrays. If $\text{MC}(\mathcal{X}, \psi, \text{DIRECTION})$ is executed, then for all $v \in W$:*

- if DIRECTION is FORWARD, $V_A(\psi, v) = \top$ iff there is $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$;
- if DIRECTION is BACKWARD, $V_{\bar{A}}(\psi, v) = \top$ iff there is $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{lst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$.

Proof. The proof is by induction on the number n of occurrences of $\langle \text{A} \rangle$ and $\langle \bar{\text{A}} \rangle$ modalities in ψ .

(Base case: $n = 0$) Since $\text{ModSubf}_{\bar{A}\bar{A}}(\psi) = \emptyset$, conditions 1 and 2 of Lemma 1 are satisfied and the thesis trivially holds.

(Inductive case: $n > 0$) The formula ψ contains at least an $\langle \text{A} \rangle$ or an $\langle \bar{\text{A}} \rangle$ modality. Hence $\text{ModSubf}_{\bar{A}\bar{A}}(\psi) \neq \emptyset$. Since each recursive call to MC (either at line 2 or 4) is performed on a formula ϕ featuring a number of occurrences of $\langle \text{A} \rangle$ and $\langle \bar{\text{A}} \rangle$ which is strictly less than the number of their occurrences in ψ , we can apply the inductive hypothesis. As a consequence, when the control flow reaches line 5, it holds that:

1. for each $\langle \text{A} \rangle \phi \in \text{ModSubf}_{\bar{A}\bar{A}}(\psi)$ and $v' \in W$, $V_A(\phi, v') = \top$ iff there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\rho) = v'$ and $\mathcal{X}, \rho \models \phi$;
2. for each $\langle \bar{\text{A}} \rangle \phi \in \text{ModSubf}_{\bar{A}\bar{A}}(\psi)$ and $v' \in W$, $V_{\bar{A}}(\phi, v') = \top$ iff there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{lst}(\rho) = v'$ and $\mathcal{X}, \rho \models \phi$.

This implies that conditions 1 and 2 of Lemma 1 are fulfilled. Hence (assuming that DIRECTION is FORWARD), it holds that, for $v \in W$, $V_A(\psi, v) = \top$ iff there exists $\rho \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{X}, \rho \models \psi$. The case for $\text{DIRECTION} = \text{BACKWARD}$ is symmetric, and thus omitted. \square

As an immediate consequence we have that the procedure MC solves the model checking problem for $\text{A}\bar{\text{A}}\text{B}$ with an algorithm belonging to the complexity class \mathbf{P}^{NP} .

Corollary 1. *Let $\mathcal{X} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure and ψ be an $\text{A}\bar{\text{A}}\text{B}$ formula. If $\text{MC}(\mathcal{X}, \neg\psi, \text{FORWARD})$ is executed, then $V_A(\neg\psi, w_0) = \perp \iff \mathcal{X} \models \psi$.*

Corollary 2. *The model checking problem for $\text{A}\bar{\text{A}}\text{B}$ formulas over finite Kripke structures is in \mathbf{P}^{NP} .*

Proof. Given a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an $\overline{A\overline{A}B}$ formula ψ , the number of recursive calls performed by $\text{MC}(\mathcal{K}, \neg\psi, \text{FORWARD})$ is at most $|\psi|$. Each one costs $O(|\psi| + |W| \cdot (|\mathcal{K}| + |\psi| + |\psi| \cdot |W|))$, where the first addend comes from searching ψ for its modal subformulas (lines 1–4), and the second one from the preparation of the input for the oracle call, for each $v \in W$ (lines 5–9). Therefore its (deterministic) complexity is $O(|\psi|^2 \cdot |\mathcal{K}|^2)$. As for $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup \overline{V_A})$, its (non-deterministic) complexity is $O(|\psi|^3 \cdot |\mathcal{K}|)$, where $|\psi|$ is a bound to the number of subformulas and $O(|\psi|^2 \cdot |\mathcal{K}|)$ is the number of steps necessary to generate and check $\tilde{\rho}$. \square

Symmetrically, by easily adapting the procedure Oracle , it is straightforward to prove that the model checking problem for $\overline{A\overline{A}E}$ formulas is in \mathbf{P}^{NP} as well.

4 \mathbf{P}^{NP} -hardness of model checking for AB formulas

In this section, we prove that model checking for AB (and $\overline{A\overline{A}E}$) formulas is hard for \mathbf{P}^{NP} by reducing the \mathbf{P}^{NP} -complete problem SNSAT (Sequentially Nested SATisfiability), a logical problem with nested satisfiability questions [11], to it. SNSAT is defined as follows.

Definition 7. An instance \mathcal{I} of SNSAT consists of a set of Boolean variables $X = \{x_1, \dots, x_n\}$ and a set of Boolean formulas $\{F_1(Z_1), F_2(x_1, Z_2), \dots, F_n(x_1, \dots, x_{n-1}, Z_n)\}$, where, for $i = 1, \dots, n$, $F_i(x_1, \dots, x_{i-1}, Z_i)$ features variables in $\{x_1, \dots, x_{i-1}\}$ and in $Z_i = \{z_i^1, \dots, z_i^{j_i}\}$, the latter being a set of variables local to F_i , that is, $Z_i \cap Z_j = \emptyset$, for $j \neq i$, and $X \cap Z_i = \emptyset$. We denote $|X| (= n)$ by $|\mathcal{I}|$. Let $v_{\mathcal{I}}$ be the valuation of the variables in X defined as follows: $v_{\mathcal{I}}(x_i) = \top \iff F_i(v_{\mathcal{I}}(x_1), \dots, v_{\mathcal{I}}(x_{i-1}), Z_i)$ is satisfiable (by assigning suitable values to the local variables $z_i^1, \dots, z_i^{j_i} \in Z_i$). SNSAT is the problem of deciding, given an instance \mathcal{I} , with $|\mathcal{I}| = n$, whether $v_{\mathcal{I}}(x_n) = \top$. In such a case, we say that \mathcal{I} is a positive instance of SNSAT.

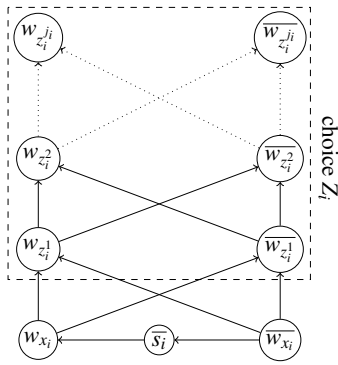
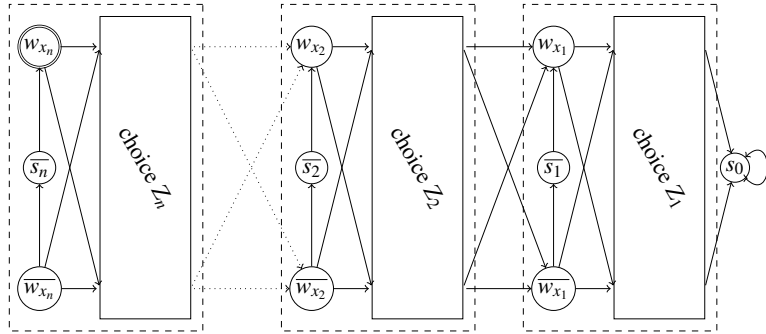
Given an SNSAT instance \mathcal{I} , with $|\mathcal{I}| = n$, the valuation $v_{\mathcal{I}}$ is unique and it can be easily computed by a \mathbf{P}^{NP} algorithm as follows. A first query to a SAT oracle determines whether $v_{\mathcal{I}}(x_1)$ is \top or \perp , since $v_{\mathcal{I}}(x_1) = \top$ iff $F_1(Z_1)$ is satisfiable. Then, we replace x_1 by the value $v_{\mathcal{I}}(x_1)$ in $F_2(x_1, Z_2)$ and another query to the SAT oracle is performed to determine whether $F_2(v_{\mathcal{I}}(x_1), Z_2)$ is satisfiable, gaining the value of $v_{\mathcal{I}}(x_2)$. This step is iterated other $n - 2$ times, until the value for $v_{\mathcal{I}}(x_n)$ is obtained.

Let \mathcal{I} be an instance of SNSAT, with $|\mathcal{I}| = n$. We now show how to build a finite Kripke structure $\mathcal{K}_{\mathcal{I}}$ and an AB formula $\Phi_{\mathcal{I}}$, by using *logarithmic working space*, such that \mathcal{I} is a positive instance of SNSAT if and only if $\mathcal{K}_{\mathcal{I}} \models \Phi_{\mathcal{I}}$. Such a reduction is inspired by similar constructions from [11].

Let $Z = \bigcup_{i=1}^n Z_i$ and let $R = \{r_i \mid i = 1, \dots, n\}$ and $R_i = R \setminus \{r_i\}$ be $n + 1$ sets of auxiliary variables. The Kripke structure $\mathcal{K}_{\mathcal{I}}$ consists of a suitable composition of n instances of a *gadget* (an instance for each variable $x_1, \dots, x_n \in X$). The structure of the gadget for x_i , with $1 \leq i \leq n$, is shown in Figure 4, assuming that the labeling of states (nodes) is defined as follows:

- $\mu(w_{x_i}) = X \cup Z \cup \{s, t\} \cup R_i$, and $\mu(\overline{w_{x_i}}) = (X \setminus \{x_i\}) \cup Z \cup \{s, t\} \cup R_i \cup \{p_{\overline{x_i}}\}$;
- for $u_i = 1, \dots, j_i$, $\mu(w_{z_i^{u_i}}) = X \cup Z \cup \{s, t\} \cup R_i$, and $\mu(\overline{w_{z_i^{u_i}}}) = X \cup (Z \setminus \{z_i^{u_i}\}) \cup \{s, t\} \cup R_i$;
- $\mu(\overline{s_i}) = X \cup Z \cup \{t\} \cup R_i$.

The Kripke structure $\mathcal{K}_{\mathcal{I}}$ is obtained by sequentializing (adding suitable arcs) the n instances of the gadget (in reverse order, from x_n to x_1), adding a collector terminal state s_0 , with labeling $\mu(s_0) = X \cup Z \cup \{s\} \cup R$, and setting w_{x_n} as the initial state. The overall construction is reported in Figure 5. Formally, $\mathcal{K}_{\mathcal{I}} = (X \cup Z \cup \{s, t\} \cup R \cup \{p_{\overline{x_i}} \mid i = 1, \dots, n\}, W, \delta, \mu, w_{x_n})$. $\mathcal{K}_{\mathcal{I}}$ enjoys the following properties: (i) any track satisfying s does not pass through any $\overline{s_i}$, for $1 \leq i \leq n$; (ii) any track *not* satisfying t has s_0 as its

Figure 4: The gadget for x_i .Figure 5: Kripke structure $\mathcal{K}_{\mathcal{S}}$ associated with an SNSAT instance \mathcal{S} , with $|\mathcal{S}| = n$. Notice that the states \bar{s}_n and \bar{w}_{x_n} are unreachable.

last state; (iii) any track *not* satisfying r_i passes through some state of the i -th gadget, for $1 \leq i \leq n$; (iv) the only track satisfying $p_{\bar{x}_i}$ is \bar{w}_{x_i} (notice that $|\bar{w}_{x_i}| = 1$), for $1 \leq i \leq n$.

A track $\rho \in \text{Trk}_{\mathcal{K}_{\mathcal{S}}}$ induces a truth assignment of all the proposition letters, denoted by ω_{ρ} , which is defined as $\omega_{\rho}(y) = \top \iff \mathcal{K}_{\mathcal{S}}, \rho \models y$, for any letter y . In the following, we will write $\omega_{\rho}(Z_i)$ for $\omega_{\rho}(z_i^1), \dots, \omega_{\rho}(z_i^j)$. In particular, if ρ starts from some state w_{x_i} or \bar{w}_{x_i} , and satisfies $s \wedge \neg t$ (that is, it reaches the collector state s_0 without visiting any node \bar{s}_j , for $1 \leq j \leq i$), ω_{ρ} fulfills the following conditions: for $1 \leq m \leq i$,

- if $w_{x_m} \in \text{states}(\rho)$, then $\omega_{\rho}(x_m) = \top$, and if $\bar{w}_{x_m} \in \text{states}(\rho)$, then $\omega_{\rho}(x_m) = \perp$;
- for $1 \leq u_m \leq j_m$, if $w_{z_m^{u_m}} \in \text{states}(\rho)$, then $\omega_{\rho}(z_m^{u_m}) = \top$, and if $\bar{w}_{z_m^{u_m}} \in \text{states}(\rho)$, then $\omega_{\rho}(z_m^{u_m}) = \perp$;

It immediately follows that $\mathcal{K}_{\mathcal{S}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$ iff $F_m(\omega_{\rho}(x_1), \dots, \omega_{\rho}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$. Finally, let $\mathcal{F}_{\mathcal{S}} = \{\psi_k \mid 0 \leq k \leq n+1\}$ be the set of formulas defined as: $\psi_0 = \perp$ and, for $k \geq 1$,

$$\psi_k = \langle \mathbf{A} \rangle \left[\begin{array}{c} (s \wedge \neg t) \wedge \bigwedge_{i=1}^n \left((x_i \wedge \neg r_i) \rightarrow F_i(x_1, \dots, x_{i-1}, Z_i) \right) \\ \wedge \\ [B] \left(\left(\bigvee_{i=1}^n \langle \mathbf{A} \rangle p_{\bar{x}_i} \right) \rightarrow \langle \mathbf{A} \rangle (\neg s \wedge \ell_{=2} \wedge \langle \mathbf{A} \rangle (\ell_{=2} \wedge \neg \psi_{k-1})) \right) \end{array} \right],$$

$\underbrace{\hspace{15em}}_{\varphi_k}$

where $\ell_{=2} = \langle \mathbf{B} \rangle \top \wedge [B][B]\perp$ is satisfied only by tracks of length 2. The first conjunct of φ_k ($s \wedge \neg t$) forces the track to reach the collector state s_0 , without visiting any state \bar{s}_j . The second conjunct checks that if the track assigns the truth value \top to x_m passing through w_{x_m} (with $1 \leq m \leq n$), then $F_m(x_1, \dots, x_{m-1}, Z_m)$ is satisfied by ω_{ρ} (which amounts to say that the SAT problem connected with Z_m has a positive answer, for the selected values of x_1, \dots, x_{m-1}). Conversely, the third conjunct ensures that if the track assigns the truth value \perp to some x_m by passing through \bar{w}_{x_m} , then, intuitively, the SAT problem connected with Z_m has no assignment satisfying $F_m(x_1, \dots, x_{m-1}, Z_m)$. As a matter of fact, if ρ satisfies φ_k for some $k \geq 2$, and assigns \perp to x_m , then there is a prefix $\tilde{\rho}$ of ρ ending in \bar{w}_{x_m} . Since $\bigvee_{i=1}^n \langle \mathbf{A} \rangle p_{\bar{x}_i}$ is satisfied by $\tilde{\rho}$, then $\langle \mathbf{A} \rangle (\neg s \wedge \ell_{=2} \wedge \langle \mathbf{A} \rangle (\ell_{=2} \wedge \neg \psi_{k-1}))$ must be satisfied as well. The only possibility is that the track $\bar{s}_m \cdot w_{x_m}$ does not model ψ_{k-1} (as $\bar{w}_{x_m} \cdot \bar{s}_m$ has to model $\langle \mathbf{A} \rangle (\ell_{=2} \wedge \neg \psi_{k-1})$). However, since $\psi_{k-1} = \langle \mathbf{A} \rangle \varphi_{k-1}$, this holds iff $\mathcal{K}, w_{x_m} \not\models \psi_{k-1}$.

The following theorem states the correctness of the construction.

Theorem 2. *Let \mathcal{S} be an instance of SNSAT with $|\mathcal{S}| = n$, and let $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{S}}$ be defined as above. For all $0 \leq k \leq n+1$ and all $r = 1, \dots, n$, it holds that:*

1. *if $k \geq r$, then $v_{\mathcal{S}}(x_r) = \top \iff \mathcal{K}_{\mathcal{S}}, w_{x_r} \models \psi_k$;*

2. if $k \geq r + 1$, then $v_{\mathcal{G}}(x_r) = \perp \iff \mathcal{K}_{\mathcal{G}}, \overline{w_{x_r}} \models \psi_k$.

Proof. The proof is by induction on $k \geq 0$.

(Base case: $k = 0$). The thesis trivially holds.

(Inductive case: $k \geq 1$). We first prove the (\Leftarrow) implication for both item 1 and item 2.

- (Item 1) Assume that $k \geq r$ and $\mathcal{K}_{\mathcal{G}}, w_{x_r} \models \psi_k$. Thus, there exists $\rho \in \text{Trk}_{\mathcal{K}_{\mathcal{G}}}$ such that $\rho = w_{x_r} \cdots s_0$ does not pass through any $\overline{s_m}$, $1 \leq m \leq r$ and $\mathcal{K}_{\mathcal{G}}, \rho \models \phi_k$. We show by induction on $1 \leq m \leq r$ that $\omega_{\rho}(x_m) = v_{\mathcal{G}}(x_m)$.
 - Let us consider first the case where ρ passes through w_{x_m} , implying that $\omega_{\rho}(x_m) = \top$; thus $\mathcal{K}_{\mathcal{G}}, \rho \models x_m \wedge \neg r_m$ and $\mathcal{K}_{\mathcal{G}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$. If $m = 1$ (base case), since F_1 is satisfiable, then $v_{\mathcal{G}}(x_1) = \top$. If $m \geq 2$ (inductive case), by the inductive hypothesis, it holds that $\omega_{\rho}(x_1) = v_{\mathcal{G}}(x_1), \dots, \omega_{\rho}(x_{m-1}) = v_{\mathcal{G}}(x_{m-1})$. Since $\mathcal{K}_{\mathcal{G}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$ or, equivalently, $F_m(\omega_{\rho}(x_1), \dots, \omega_{\rho}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$, it holds that $F_m(v_{\mathcal{G}}(x_1), \dots, v_{\mathcal{G}}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$ and, by definition of $v_{\mathcal{G}}$, $v_{\mathcal{G}}(x_m) = \top$.
 - Conversely, let us consider the case where ρ passes through $\overline{w_{x_m}}$, implying that $\omega_{\rho}(x_m) = \perp$ and $m < r$, as we are assuming $\text{fst}(\rho) = w_{x_r}$. In this case, the prefix $w_{x_r} \cdots \overline{w_{x_m}}$ of ρ satisfies both $\bigvee_{i=1}^n \langle A \rangle p_{\overline{x_i}}$ and $\langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\ell_{=2} \wedge \neg \psi_{k-1}))$. Therefore, $\mathcal{K}_{\mathcal{G}}, \overline{w_{x_m}} \cdot \overline{s_m} \models \langle A \rangle (\ell_{=2} \wedge \neg \psi_{k-1})$ and $\mathcal{K}_{\mathcal{G}}, \overline{s_m} \cdot w_{x_m} \not\models \psi_{k-1}$, with $\psi_{k-1} = \langle A \rangle \phi_{k-1}$. Hence $\mathcal{K}_{\mathcal{G}}, w_{x_m} \not\models \psi_{k-1}$. Since $1 \leq m < r$, we have $1 \leq m < r \leq k$, thus $k' = k - 1 \geq m \geq 1$. By the inductive hypothesis (on $k' = k - 1$), we get that $v_{\mathcal{G}}(x_m) = \perp$.

Therefore $v_{\mathcal{G}}(x_r) = \omega_{\rho}(x_r)$ and, since $w_{x_r} \in \text{states}(\rho)$, we have that $\omega_{\rho}(x_r) = \top$ and the thesis, that is, $v_{\mathcal{G}}(x_r) = \top$, follows.

- (Item 2) Assume that $k \geq r + 1$ and $\mathcal{K}_{\mathcal{G}}, \overline{w_{x_r}} \models \psi_k$. The proof follows the same steps as the previous case and it is thus only sketched: there exists $\rho \in \text{Trk}_{\mathcal{K}_{\mathcal{G}}}$ such that $\rho = \overline{w_{x_r}} \cdots s_0$ does not pass through any $\overline{s_m}$, for $1 \leq m \leq r$, and $\mathcal{K}_{\mathcal{G}}, \rho \models \phi_k$. The only thing which changes is that the prefix $\overline{w_{x_r}}$ satisfies $\bigvee_{i=1}^n \langle A \rangle p_{\overline{x_i}}$, thus as before we get $\mathcal{K}_{\mathcal{G}}, w_{x_r} \not\models \psi_{k-1}$. Now, $k' = k - 1 \geq r \geq 1$ and, by the inductive hypothesis (on $k' = k - 1$), it holds that $v_{\mathcal{G}}(x_r) = \perp$.

We prove now the converse implication (\Rightarrow) for both item 1 and item 2.

- (Item 1) Assume that $k \geq r$ and $v_{\mathcal{G}}(x_r) = \top$. Let us consider the track $\rho \in \text{Trk}_{\mathcal{K}_{\mathcal{G}}}$, $\rho = w_{x_r} \cdots s_0$ never passing through any $\overline{s_m}$, for $1 \leq m \leq r$, such that $w_{x_m} \in \text{states}(\rho)$ if $v_{\mathcal{G}}(x_m) = \top$, and $\overline{w_{x_m}} \in \text{states}(\rho)$ if $v_{\mathcal{G}}(x_m) = \perp$, for $1 \leq m \leq r$. Such a choice of ρ ensures that $v_{\mathcal{G}}(x_m) = \omega_{\rho}(x_m)$. In addition, the choice of ρ has to induce also the proper evaluation of local variables, that is, if $v_{\mathcal{G}}(x_m) = \top$, then for $1 \leq u_m \leq j_m$, $w_{z_m^{u_m}} \in \text{states}(\rho)$ if $F_m(v_{\mathcal{G}}(x_1), \dots, v_{\mathcal{G}}(x_{m-1}), Z_m)$ is satisfied for $z_m^{u_m} = \top$, $\overline{w_{z_m^{u_m}}} \in \text{states}(\rho)$ otherwise. Notice that such a choice of ρ is always possible. We have to show that $\mathcal{K}_{\mathcal{G}}, \rho \models \phi_k$, hence $\mathcal{K}_{\mathcal{G}}, w_{x_r} \models \psi_k$.
 - For all $1 \leq m \leq r$ such that $v_{\mathcal{G}}(x_m) = \top$, it holds that $F_m(v_{\mathcal{G}}(x_1), \dots, v_{\mathcal{G}}(x_{m-1}), Z_m)$ is satisfiable. Hence, by our choice of ρ , $F_m(\omega_{\rho}(x_1), \dots, \omega_{\rho}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$, or, equivalently, $\mathcal{K}_{\mathcal{G}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$. Therefore, $\mathcal{K}_{\mathcal{G}}, \rho \models \bigwedge_{i=1}^n ((x_i \wedge \neg r_i) \rightarrow F_i(x_1, \dots, x_{i-1}, Z_i))$.
 - Conversely, for all $1 \leq m < r$ such that $v_{\mathcal{G}}(x_m) = \perp$ ($m \neq r$ as, by hypothesis, $v_{\mathcal{G}}(x_r) = \top$), it holds that $\overline{w_{x_m}} \in \text{states}(\rho)$. Since $m < r$, it holds that $k \geq r > m$ and $k - 1 \geq m \geq 1$. By the inductive hypothesis, we have that $\mathcal{K}_{\mathcal{G}}, w_{x_m} \not\models \psi_{k-1}$. It follows that $\mathcal{K}_{\mathcal{G}}, \overline{s_m} \cdot w_{x_m} \models \neg \psi_{k-1} \wedge \ell_{=2}$, $\mathcal{K}_{\mathcal{G}}, \overline{w_{x_m}} \cdot \overline{s_m} \models \neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2})$ and $\mathcal{K}_{\mathcal{G}}, \overline{w_{x_m}} \models \langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2}))$. Hence, $\mathcal{K}_{\mathcal{G}}, \rho \models [B]((\bigvee_{i=1}^n \langle A \rangle p_{\overline{x_i}}) \rightarrow \langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2})))$.

Combining the two cases, we can conclude that $\mathcal{K}_{\mathcal{G}}, \rho \models \phi_k$.

- (Item 2) Assume that $k \geq r + 1$ and $v_{\mathcal{G}}(x_r) = \perp$. The proof is as before and it is sketched. In

this case, we choose a track $\rho = \overline{w_{x_r}} \cdots s_0$. Since $k' = k - 1 \geq r$, by the inductive hypothesis, $\mathcal{K}_{\mathcal{S}}, w_{x_r} \not\models \psi_{k-1}$, and we can prove that $\mathcal{K}_{\mathcal{S}}, \overline{w_{x_r}} \models \langle \mathbf{A} \rangle (\neg s \wedge \ell_{=2} \wedge \langle \mathbf{A} \rangle (\neg \psi_{k-1} \wedge \ell_{=2}))$. \square

The correctness of the reduction from SNSAT to model checking for AB follows as a corollary.

Corollary 3. *Let \mathcal{S} be an instance of SNSAT, with $|\mathcal{S}| = n$, and let $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{S}}$ be defined as above. Then, $v_{\mathcal{S}}(x_n) = \top \iff \mathcal{K}_{\mathcal{S}} \models [B]_{\perp} \rightarrow \psi_n$.*

Proof. By Theorem 2, $v_{\mathcal{S}}(x_n) = \top \iff \mathcal{K}_{\mathcal{S}}, w_{x_n} \models \psi_n$. If $v_{\mathcal{S}}(x_n) = \top$, then $\mathcal{K}_{\mathcal{S}}, w_{x_n} \models \psi_n$ and, since w_{x_n} is the only initial track satisfying $[B]_{\perp}$ (only satisfiable by tracks of length 1), $\mathcal{K}_{\mathcal{S}} \models [B]_{\perp} \rightarrow \psi_n$. Conversely, if $\mathcal{K}_{\mathcal{S}} \models [B]_{\perp} \rightarrow \psi_n$, then $\mathcal{K}_{\mathcal{S}}, w_{x_n} \models \psi_n$, allowing us to conclude that $v_{\mathcal{S}}(x_n) = \top$. \square

Eventually we can state the complexity of the problem.

Corollary 4. *The model checking problem for AB formulas over finite Kripke structures is $\mathbf{P}^{\mathbf{NP}}$ -hard (under $\mathbf{LOGSPACE}$ reductions).*

Proof. The result follows from Corollary 3 considering that, for an instance of SNSAT \mathcal{S} , with $|\mathcal{S}| = n$, $\mathcal{K}_{\mathcal{S}}$ and $\psi_n \in \mathcal{F}_{\mathcal{S}}$ have a size polynomial in n and in the length of the formulas of \mathcal{S} . Moreover, their structures are repetitive, therefore they can be built by using logarithmic working space. \square

We can prove the same complexity result for the symmetric fragment $\overline{\mathbf{A}\mathbf{E}}$, just by transposing the edges of $\mathcal{K}_{\mathcal{S}}$, and by replacing $[B]$ with $[E]$ and $\langle \mathbf{A} \rangle$ with $\langle \overline{\mathbf{A}} \rangle$ in the definition of ψ_n .

We summarize all the $\mathbf{P}^{\mathbf{NP}}$ -completeness results achieved in the following statement.

Corollary 5. *The model checking problem for AB, $\overline{\mathbf{A}\mathbf{E}}$, $\mathbf{A}\overline{\mathbf{A}\mathbf{B}}$, and $\mathbf{A}\overline{\mathbf{A}\mathbf{E}}$ formulas over finite Kripke structures is $\mathbf{P}^{\mathbf{NP}}$ -complete.*

We conclude the paper by providing a complexity upper and lower bound for $\overline{\mathbf{A}\mathbf{B}}$ and the symmetric fragment $\mathbf{A}\mathbf{E}$. A $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ model checking algorithm for $\overline{\mathbf{A}\mathbf{B}}$ formulas can be obtained by a suitable adaptation of the one for $\mathbf{A}\overline{\mathbf{A}\mathbf{B}}$ we devised in [19]. Due to the lack of space, we outline the construction in the appendix of [3], and state here only the result. As for the hardness, we can observe that the $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hardness of $\overline{\mathbf{A}}$ and \mathbf{A} , proved in [19], immediately propagates to $\overline{\mathbf{A}\mathbf{B}}$ and $\mathbf{A}\mathbf{E}$, respectively.

Theorem 3. *The model checking problem for $\overline{\mathbf{A}\mathbf{B}}$ and $\mathbf{A}\mathbf{E}$ formulas over finite Kripke structures is in $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ and it is hard for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$.*

5 Conclusions and future work

In this paper, we have proved that the model checking problem for the HS fragments AB, $\overline{\mathbf{A}\mathbf{E}}$, $\mathbf{A}\overline{\mathbf{A}\mathbf{B}}$, and $\mathbf{A}\overline{\mathbf{A}\mathbf{E}}$ is $\mathbf{P}^{\mathbf{NP}}$ -complete. They are thus somehow “halfway” between $\mathbf{A}\overline{\mathbf{A}\mathbf{B}\mathbf{B}}$, $\mathbf{A}\overline{\mathbf{A}\mathbf{E}\mathbf{E}}$, and $\mathbf{A}\overline{\mathbf{A}\mathbf{B}\mathbf{E}}$, which are \mathbf{PSPACE} -complete [2, 17, 18], and Prop, B, and E, which are $\mathbf{co-NP}$ -complete [2, 17], and \mathbf{A} , $\overline{\mathbf{A}}$, and $\mathbf{A}\overline{\mathbf{A}}$, whose model checking is in $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ [19]. In addition, we have shown that model checking for the HS fragments $\overline{\mathbf{A}\mathbf{B}}$ and $\mathbf{A}\mathbf{E}$ has a lower complexity (it is in between $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$) [3].

As for future work, we are looking for possible improvements to known complexity results for (full) HS model checking. We know that it is $\mathbf{EXSPACE}$ -hard (we proved $\mathbf{EXSPACE}$ -hardness of its fragment BE), while the only available decision procedure is nonelementary. We also started a comparative study of the expressiveness of HS fragments (with the current semantics as well as with some variants of it, which limit past/future branching) and of standard temporal logics, such as LTL, CTL, and CTL*.

Acknowledgments. The work by Alberto Molinari, Angelo Montanari, and Pietro Sala has been supported by the GNCS project *Logic, Automata, and Games for Auto-Adaptive Systems*.

References

- [1] J. F. Allen (1983): *Maintaining Knowledge about Temporal Intervals*. *Communications of the ACM* 26(11), pp. 832–843, doi:10.1145/182.358434.
- [2] L. Bozzelli, A. Molinari, A. Montanari, A. Peron & P. Sala (2016): *Interval Temporal Logic Model Checking: the Border Between Good and Bad HS Fragments*. In: *IJCAR*, LNAI 9706, Springer, pp. 389–405, doi:10.1007/978-3-319-40229-1_27. Available at <https://www.dimi.uniud.it/la-ricerca/publicazioni/preprints/1.2016/>.
- [3] L. Bozzelli, A. Molinari, A. Montanari, A. Peron & P. Sala (2016): *Model Checking the Logic of Allen’s Relations Meets and Started-by is P^{NP} -Complete*. Technical Report, University of Udine, Udine, Italy. Available at <https://www.dimi.uniud.it/la-ricerca/publicazioni/preprints/3488/>.
- [4] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari & G. Sciavicco (2014): *The dark side of interval temporal logic: marking the undecidability border*. *Annals of Mathematics and Artificial Intelligence* 71(1-3), pp. 41–83, doi:10.1007/s10472-013-9376-4.
- [5] D. Bresolin, V. Goranko, A. Montanari & P. Sala (2010): *Tableaux for Logics of Subinterval Structures over Dense Orderings*. *Journal of Logic and Computation* 20(1), pp. 133–166, doi:10.1093/logcom/exn063.
- [6] D. Bresolin, V. Goranko, A. Montanari & G. Sciavicco (2009): *Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions*. *Annals of Pure and Applied Logic* 161(3), pp. 289–304, doi:10.1016/j.apal.2009.07.003.
- [7] E. A. Emerson & J. Y. Halpern (1986): “Sometimes” and “not never” revisited: on branching versus linear time temporal logic. *Journal of the ACM* 33(1), pp. 151–178, doi:10.1145/4904.4999.
- [8] F. Giunchiglia & P. Traverso (1999): *Planning as Model Checking*. In: *ECP*, LNCS 1809, Springer, pp. 1–20, doi:10.1007/10720246_1.
- [9] G. Gottlob (1995): *NP Trees and Carnap’s Modal Logic*. *Journal of the ACM* 42(2), pp. 421–457, doi:10.1145/201019.201031.
- [10] J. Y. Halpern & Y. Shoham (1991): *A Propositional Modal Logic of Time Intervals*. *Journal of the ACM* 38(4), pp. 935–962, doi:10.1145/115234.115351.
- [11] F. Laroussinie, N. Markey & P. Schnoebelen (2001): *Model Checking CTL^+ and $FCTL$ is Hard*. In: *FOS-SACS*, pp. 318–331, doi:10.1007/3-540-45315-6_21.
- [12] A. Lomuscio & J. Michaliszyn (2013): *An Epistemic Halpern-Shoham Logic*. In: *IJCAI*, pp. 1010–1016.
- [13] A. Lomuscio & J. Michaliszyn (2014): *Decidability of model checking multi-agent systems against a class of EHS specifications*. In: *ECAI*, pp. 543–548, doi:10.3233/978-1-61499-419-0-543.
- [14] A. Lomuscio & J. Michaliszyn (2016): *Model Checking Multi-Agent Systems against Epistemic HS Specifications with Regular Expressions*. In: *KR*, pp. 298–308.
- [15] A. Lomuscio, H. Qu & F. Raimondi (2009): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. In: *CAV*, Springer, pp. 682–688, doi:10.1007/978-3-642-02658-4_55.
- [16] A. Molinari, A. Montanari, A. Murano, G. Perelli & A. Peron (2016): *Checking interval properties of computations*. *Acta Informatica*, doi:10.1007/s00236-015-0250-1. Available at <http://arxiv.org/abs/1601.03195>. Accepted for publication.
- [17] A. Molinari, A. Montanari & A. Peron (2015): *Complexity of ITL model checking: some well-behaved fragments of the interval logic HS*. In: *TIME*, pp. 90–100, doi:10.1109/TIME.2015.12. Available at <http://arxiv.org/abs/1601.03202>.
- [18] A. Molinari, A. Montanari & A. Peron (2015): *A Model Checking Procedure for Interval Temporal Logics based on Track Representatives*. In: *CSL*, pp. 193–210, doi:10.4230/LIPIcs.CSL.2015.193.
- [19] A. Molinari, A. Montanari, A. Peron & P. Sala (2016): *Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture*. In: *KR*, pp. 473–483.
- [20] B. Moszkowski (1983): *Reasoning About Digital Circuits*. Ph.D. thesis, Stanford University, Stanford, CA.

- [21] A. Pnueli (1977): *The temporal logic of programs*. In: *FOCS*, IEEE Computer Society, pp. 46–57, doi:10.1109/SFCS.1977.32.
- [22] I. Pratt-Hartmann (2005): *Temporal prepositions and their logic*. *Artificial Intelligence* 166(1-2), pp. 1–36, doi:10.1016/j.artint.2005.04.003.
- [23] P. Roeper (1980): *Intervals and Tenses*. *J. of Philosophical Logic* 9, pp. 451–469, doi:10.1007/BF00262866.
- [24] P. Schnoebelen (2003): *Oracle Circuits for Branching-Time Model Checking*. In: *ICALP*, LNCS 2719, Springer, pp. 790–801, doi:10.1007/3-540-45061-0.62.
- [25] C. Zhou & M. R. Hansen (2004): *Duration Calculus - A Formal Approach to Real-Time Systems*. Monographs in Theoretical Computer Science. An EATCS Series, Springer.