

Model Checking for Fragments of the Interval Temporal Logic HS at the Low Levels of the Polynomial Time Hierarchy[☆]

Laura Bozzelli^a, Alberto Molinari^b, Angelo Montanari^b, Adriano Peron^a, Pietro Sala^c

^a*Department of Electronic Engineering and Information Technologies, University of Napoli “Federico II”, Italy*

^b*Department of Mathematics, Computer Science, and Physics, University of Udine, Italy*

^c*Department of Computer Science, University of Verona, Italy*

Abstract

Some temporal properties of reactive systems, such as actions with duration, accomplishments, and temporal aggregations, which are inherently interval-based, can not be properly dealt with by the standard, point-based temporal logics LTL, CTL and CTL*, as they give a state-by-state account of system evolution. Conversely, interval temporal logics—which feature intervals, instead of points, as their primitive entities—are highly expressive formalisms for temporal representation and reasoning that naturally allow one to deal with them.

In this paper, we study the model checking (MC) problem for Halpern and Shoham’s modal logic of time intervals (HS), interpreted on Kripke structures, under the homogeneity assumption, according to which a proposition letter holds over a finite computation path (interval) if and only if it holds at all of its states. HS is the best known interval-based temporal logic, which has one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen’s relations), apart from equality. We focus on the MC problem for some HS fragments featuring modalities for (a subset of) Allen’s relations meet, met-by, started-by, and finished-by, showing that it is in $\mathbf{P}^{\mathbf{NP}}$, a class to which other point-based logics (e.g., CTL+ and FCTL) are known to belong. Additionally, we provide some complexity lower bounds to the problem. All the algorithms we propose can be efficiently implemented by means of a polynomial-time procedure which iteratively invokes a SAT-solver, enabling us to directly exploit the great speed of SAT-solvers.

Keywords: Interval Temporal Logic, Model Checking, Computational Complexity

2010 MSC: 03B70, 68Q60

1. Introduction

Our dependence on hardware and software systems is continuously increasing under many aspects of our everyday life. Embedded systems are employed for critical applications, e.g.,

[☆]This paper is an extended and revised version of [40] and [10].

Email addresses: lr.bozzelli@gmail.com (Laura Bozzelli), molinari.alberto@gmail.com (Alberto Molinari), angelo.montanari@uniud.it (Angelo Montanari), adrperon@unina.it (Adriano Peron), pietro.sala@univr.it (Pietro Sala)

air traffic and railway control systems, telephone networks, and nuclear plants monitoring. Security protocols are at the basis of e-commerce websites and services, and are exploited in all applications which have to ensure user privacy. Biomedical instruments and equipment are endowed with automatic or proactive functionalities, and are supposed to help humans and to prevent human error. Thus, the essential requirements of safety, reliability, and correctness for these systems suggest to support their development steps, namely, design, implementation, verification, and testing, by structured methodologies possibly founded on *formal methods*—some of them are even becoming integral part of standards—as well as by suitable specification languages and automatic verification techniques and tools.

A well known technique in this setting is *model checking*. Model checking (MC) allows one to verify the desired properties of a system against a model of its behaviour [19]. Properties are usually specified by means of temporal logics, such as LTL and CTL, and systems are represented as labelled state-transition graphs (Kripke structures). MC algorithms perform, in a fully automatic way, an (implicit or explicit) exhaustive enumeration of all the states reachable by the system, and either terminate positively—proving that all properties are met—or produce a counterexample—witnessing that some behavior falsifies a property, which is extremely useful for debugging purposes.

MC can be applied during the early stages of the development cycle, allowing one to analyze even partial specifications, in such a way that it is not necessary to completely describe a system before information can be obtained regarding its correctness. MC has been applied in a variety of practical scenarios, including, for instance, communication and security protocols [2, 3], embedded reactive systems [18], computer device drivers [57], database-backed web applications [24], concurrency control and transaction atomicity [35], automated verification of UML design of applications [20], testing of railway control systems [5, 44], and verification of clinical guidelines [22].

The MC problem has been investigated for a long time only in the context of *point-based temporal logics*, like LTL, CTL, and CTL*, which predicate over single system/computation states [21, 46, 47, 51]. For instance, LTL allows one to reason about changes in the truth value of formulas in a Kripke structure over a linearly-ordered temporal domain, where each moment in time has a unique possible future. More precisely, one has to consider all possible paths in a Kripke structure and to analyse, for each of them, how proposition letters, labelling the states, change from one state to the next one along the path. A systematic investigation of MC for *interval temporal logics* has been initiated very recently.

Interval temporal logics (ITLs) [27, 53, 54] feature intervals, instead of points, as their primitive entities. This makes them a highly expressive formalism for temporal representation and reasoning, with the ability of easily “mastering” advanced temporal features, such as actions with duration, accomplishments, and temporal aggregations, which can not be properly dealt with by standard, point-based temporal logics. ITLs have been applied in a variety of computer science fields, including artificial intelligence (reasoning about action and change, qualitative reasoning, planning, multi-agent systems, and computational linguistics), theoretical computer science (formal verification), and databases (temporal and spatio-temporal databases) [6, 23, 25, 34, 43, 48, 58]. However, the great expressiveness of ITLs is a double-edged sword: in most cases, the satisfiability problem for ITLs turns out to

be undecidable, and, in the few decidable ones, the standard proof machinery, like Rabin’s theorem, is usually not applicable.

The best known ITL is *Halpern and Shoham’s modal logic of time intervals* (HS, for short) [27], which has one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen’s relations [1]), apart from equality. In [27], the authors prove that the satisfiability problem for HS, interpreted over all relevant (classes of) linear orders, is undecidable. The investigation has been later extended to many HS fragments, leading to the conclusion that undecidability prevails over them as well (see [12] for an up-to-date account of undecidable fragments). However, meaningful exceptions exist, including the interval logic of temporal neighbourhood \overline{AA} and the interval logic of sub-intervals D [13, 14, 15, 41, 42].

In this paper, we focus on the *MC problem for HS*, which has entered the research agenda only recently [31, 32, 33, 36, 37, 38, 39] (it is worth pointing out that, in contrast to the case of point-based, linear temporal logics, there is no easy reduction from the MC problem to validity/satisfiability for ITL). In interval-based MC, in order to verify interval properties of computations, one needs to collect information about the states of a system into computation stretches. To this aim, we interpret each finite path of a Kripke structure (a trace) as an interval, and we define the labelling of an interval on the basis of the proposition letters which hold on the sequence of states that compose it. Different ways of defining interval labeling have been proposed in the literature. A short account of them is given in the related work section below.

1.1. Related work

In [36], Molinari et al. gave a first characterization of MC for full HS interpreted over finite Kripke structures, under the *homogeneity assumption* [49], according to which a proposition letter holds over an interval if and only if it holds at all its states. In that paper, the authors showed that finite Kripke structures can be suitably mapped into interval-based structures, called abstract interval models, over which HS formulas can be interpreted. Then, they proved a small model theorem showing (with a non-elementary procedure) the decidability of MC for full HS, which was later proved to be **EXPSpace**-hard by Bozzelli et al. in [8].¹

The MC problem for some large fragments of HS was studied in [8, 37, 38]. In [38], Molinari et al. devised an **EXPSpace** MC algorithm for the HS fragment $\overline{AABB\overline{E}}$ (resp., $\overline{AAE\overline{BE}}$) of Allen’s relations *meets*, *met-by*, *starts*, *finishes*, and *started-by* (resp., *finished-by*), which exploits the possibility of finding, for each trace (of unbounded length), an equivalent bounded-length trace representative in such a way that, while checking a property, the

¹It is worth pointing out that the homogeneity assumption, which allows us to interpret HS formulas on Kripke structures in a fairly natural way, changes the status of the satisfiability problem for HS and its fragments. In particular, in [9] Bozzelli et al. showed that, when interpreted over the (infinite) fullpaths of a finite Kripke structure (which is not the way we interpret it here), LTL and HS have the same expressive power, but the latter is provably at least exponentially more succinct. As a byproduct, the satisfiability problem for full HS, under such a trace-based semantics, turns out to be decidable. Thus, under the homogeneity assumption, the relevant issue for the satisfiability problem of HS and its fragments becomes its complexity, rather than its decidability. We addressed it for the interval logic of sub-intervals D in [11].

algorithm only needs to consider traces whose length does not exceed the given bound. The **PSPACE**-hardness of the problem was proved in [37]. In [8], Bozzelli et al. showed that MC for the fragment $A\bar{A}B\bar{B}$ (resp., $A\bar{A}E\bar{E}$) obtained from $A\bar{A}B\bar{B}E$ (resp., $A\bar{A}E\bar{B}E$) by removing the modality for the Allen relation *finished-by* (resp., *started-by*) is **PSPACE**-complete.

In [31, 32, 33], Lomuscio and Michaliszyn addressed the MC problem for some fragments of HS extended with epistemic modalities. Their semantic assumptions are different from those made in [36], making a systematic comparison of the two research lines quite difficult. In both cases, formulas of HS are evaluated over finite paths/intervals of a Kripke structure; however, while in [36] homogeneity is assumed, in [31, 32] truth of proposition letters over an interval depends only on its endpoints.

In [31], the authors focused on the HS fragment **BED** of Allen’s relations *started-by*, *finished-by*, and *contains* (since modality $\langle D \rangle$ is definable in terms of modalities $\langle B \rangle$ and $\langle E \rangle$, **BED** is actually as expressive as **BE**), extended with epistemic modalities. They considered a restricted form of MC, which verifies a given specification against a single (finite) initial computation interval. Their goal was indeed to reason about a given computation of a multi-agent system, rather than on all its admissible computations. They proved that the considered MC problem is **PSPACE**-complete; furthermore, they showed that the same problem restricted to the pure temporal fragment **BED**, that is, the one obtained by removing epistemic modalities, is in **P**. These results do not come as a surprise: modalities $\langle B \rangle$ and $\langle E \rangle$ allow one to access only sub-intervals of the initial one, whose number is quadratic in the length (number of states) of the initial interval.

In [32], they showed that the picture drastically changes with other HS fragments that allow one to access infinitely many intervals. In particular, they proved that the MC problem for the fragment $A\bar{B}L$ of Allen’s relations *meets*, *starts*, and *before* (since modality $\langle L \rangle$ is definable in terms of modality $\langle A \rangle$, $A\bar{B}L$ is actually as expressive as $A\bar{B}$), extended with epistemic modalities, is decidable with a non-elementary upper bound. Note that, thanks to modalities $\langle A \rangle$ and $\langle \bar{B} \rangle$, formulas of $A\bar{B}L$ can possibly refer to infinitely many (future) intervals.

Finally, in [33], Lomuscio and Michaliszyn showed how to use regular expressions in order to specify the way in which intervals of a Kripke structure get labelled. Such an extension leads to a significant increase in expressiveness, as the labelling of an interval is no more determined by that of its endpoints only, but it depends on the ordered sequence of states the interval consists of. They also proved that there is no corresponding increase in computational complexity, as the bounds given in [31, 32] still hold with the new semantic variant: MC for **BED** is still in **PSPACE**, and it is non-elementarily decidable for $A\bar{B}L$.

1.2. Main contributions

In this paper, we study the MC problem for some of the HS fragments featuring (a subset of) Allen’s relations *meet*, *met-by*, *started-by*, and *finished-by*, namely, **A**, \bar{A} , $A\bar{A}$, **AB**, $\bar{A}\bar{B}$, **AE**, $\bar{A}E$, $A\bar{A}B$, and $A\bar{A}E$. All these fragments have a similar computational complexity, as their MC problem settles in one of the lowest levels of the *polynomial-time hierarchy*, $\mathbf{P}^{\mathbf{NP}}$, or below. Such a class consists of the set of problems decided by a deterministic polynomial-time-bounded Turing machine, with the “support” of an oracle for the class **NP**, that is, a

tool which decides, in one computation step, whether an instance of a problem belonging to \mathbf{NP} is positive or not. $\mathbf{P}^{\mathbf{NP}}$ is also referred to as \mathbf{P} *relative to* \mathbf{NP} (relativization).

Though the fragments in the considered set are similar, some differences can be marked. In particular, the fragments A , \bar{A} , $A\bar{A}$, $\bar{A}B$, and AE are actually “easier” than the other ones, since they require the \mathbf{P} Turing machine to perform just $O(\log^2 n)$ queries to the \mathbf{NP} oracle, for an input size n , instead of $O(n^k)$ queries, for some constant $k \geq 0$, as it is required in the general case for a polynomial running time machine. The MC problem for the considered subset of fragments witnesses a “non-standard” complexity class in the polynomial-time hierarchy, called *bounded-query* class, that will be presented in more detail below.

More formally, we first prove that MC for AB , $\bar{A}E$, $A\bar{A}B$, and $A\bar{A}E$ is a $\mathbf{P}^{\mathbf{NP}}$ -complete problem. To this end, we design an $\mathbf{P}^{\mathbf{NP}}$ algorithm exploiting a *small-model* theorem proved in [8] and we prove a matching complexity lower bound.²

Next, we devise a second MC algorithm for all the remaining fragments, that is, A , \bar{A} , $A\bar{A}$, $\bar{A}B$, and AE , via a reduction *to* the problem $\text{TB}(\text{SAT})_{1 \times M}$ [50] (a problem complete for the above-mentioned bounded-query class), whose instance is a complex circuit in which some of the gates are endowed with \mathbf{NP} oracles.

Finally, we identify a lower bound, which shows that at least $\log n$ queries are needed to solve the problem. Notice that, unfortunately, such a lower bound does not exactly match the upper bound leaving open the question whether the problem can be solved by $o(\log^2 n)$ (i.e., strictly less than $O(\log^2 n)$) queries to an \mathbf{NP} oracle, or a tighter lower bound can be proved (or both).

It is worth noting that, no matter what the precise number of \mathbf{NP} queries to be performed is, both the MC algorithms we propose can be efficiently implemented in practice by means of a polynomial-time procedure which iteratively invokes a SAT-solver. Such a procedure just generates some suitable Boolean formulas, feeds the SAT-solver, and stores the results. The modular and repetitive structure of the required Boolean formulas allows us both to efficiently generate them and to exploit the *warm-restart* technique of SAT-solvers to quickly solve formulas following a common structural pattern.

1.3. Organization of the paper

In Section 2, we provide some background knowledge. More precisely, we introduce HS (Subsection 2.1), abstract interval models, and the MC problem for HS (Subsection 2.2). Then, we recall some basic notions about the complexity classes that come into play in the paper (Subsection 2.3). Finally, we sketch the picture of known and new complexity results for the MC problem for HS (Subsection 2.4).

In Section 3, we describe a $\mathbf{P}^{\mathbf{NP}}$ MC algorithm for $A\bar{A}B$ (and $A\bar{A}E$). In Section 4, we provide a different MC algorithm for the fragments $A\bar{A}$, $\bar{A}B$, and AE , whose computational complexity is lower, as it requires only $O(\log^2 n)$ queries to a \mathbf{NP} oracle.

In Section 5, we prove a $\mathbf{P}^{\mathbf{NP}}$ lower bound for AB (resp., $\bar{A}E$) MC. The bound immediately propagates to $A\bar{A}B$ (resp., $A\bar{A}E$), closing the complexity gap and proving that the MC

²In this paper, we use **LOGSPACE** many-one reductions for the proofs of hardness.

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

problem for AB , \overline{AE} , $A\overline{AB}$, and $A\overline{AE}$ is $\mathbf{P}^{\mathbf{NP}}$ -complete. In Section 6, we prove that MC for A and \overline{A} formulas requires at least $\log n$ queries to a \mathbf{NP} oracle: this bound propagates to $A\overline{A}$, \overline{AB} , and AE .

Conclusions summarize the work done and outline some directions for future research.

2. Preliminaries

2.1. The interval temporal logic HS

An interval algebra to reason about intervals and their relative order was originally proposed by Allen in [1]. Some years later, Halpern and Shoham began a systematic logical study of interval representation and reasoning by introducing the interval temporal logic now known as HS, which has one modality for each Allen interval relation but equality [27]. 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 and equality. Given a binary relation \mathcal{R} , the inverse $\overline{\mathcal{R}}$ is such that $b\overline{\mathcal{R}}a$ if and only if $a\mathcal{R}b$. Notationally, if $\langle X \rangle$ is the modality for \mathcal{R} , we denote by $\langle \overline{X} \rangle$ the modality for $\overline{\mathcal{R}}$.

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, that is, $\langle A \rangle$, $\langle L \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle D \rangle$, $\langle O \rangle$, $\langle \overline{A} \rangle$, $\langle \overline{L} \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$, $\langle \overline{D} \rangle$, and $\langle \overline{O} \rangle$.

HS formulas are defined by the grammar:

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle\psi \mid \langle \overline{X} \rangle\psi,$$

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 equivalence \leftrightarrow . Furthermore, for any existential modality $\langle X \rangle\psi$ (resp., $\langle \overline{X} \rangle\psi$), the dual universal modality $[X]\psi$ (resp., $[\overline{X}]\psi$) is defined as $\neg\langle X \rangle\neg\psi$ (resp., $\neg\langle \overline{X} \rangle\neg\psi$). 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 point.³ Under such an assumption, all HS modalities can be expressed in terms

³All the results we prove in the paper hold for the strict semantics as well.

of modalities $\langle B \rangle$, $\langle E \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$ [27]. HS can thus be seen as a multi-modal logic with these four 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 shall focus on some HS fragments not including $\langle \bar{B} \rangle$ and $\langle \bar{E} \rangle$, we add both $\langle A \rangle$ and $\langle \bar{A} \rangle$ to the set of considered HS modalities.

Definition 1 ([36]). 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 the 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 at 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:

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

2.2. Kripke structures, abstract interval models, and model checking

In the context of model checking, finite state systems are usually modelled as Kripke structures. Following the approach in [36], we 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 relation⁴ between pairs of states, $\mu : W \mapsto 2^{\mathcal{AP}}$ is a total labelling function, and $w_0 \in W$ is the initial state.

⁴A relation $\delta \subseteq W \times W$ is left-total if, for all $w \in W$, there exists at least one $w' \in W$ such that $(w, w') \in \delta$.

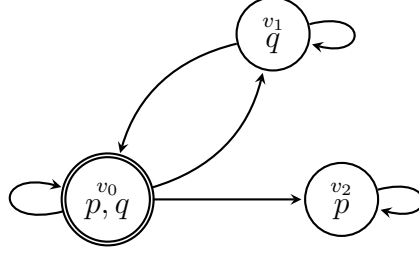


Figure 1: The Kripke structure \mathcal{K}_3 .

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

Example 1. Figure 1 depicts the finite Kripke structure $\mathcal{K}_3 = (\{p, q\}, \{v_0, v_1, v_2\}, \delta, \mu, v_0)$, where $\delta = \{(v_0, v_0), (v_0, v_1), (v_0, v_2), (v_1, v_0), (v_1, v_1), (v_2, v_2)\}$, $\mu(v_0) = \{p, q\}$, $\mu(v_1) = \{q\}$ and $\mu(v_2) = \{p\}$. The initial state v_0 is marked by a double circle.

A *trace* 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 $\text{Trc}_{\mathcal{K}}$ be the (infinite) set of all traces over a finite Kripke structure \mathcal{K} . For any trace $\rho = v_1 \cdots v_n \in \text{Trc}_{\mathcal{K}}$, we define:

- $|\rho| = n$, and for $1 \leq i \leq |\rho|$, $\rho(i) = v_i$;
- $\text{fst}(\rho) = v_1$, $\text{lst}(\rho) = v_n$;
- $\text{states}(\rho) = \{v_1, \dots, v_n\} \subseteq W$;
- $\rho(i, j) = v_i \cdots v_j$, with $1 \leq i \leq j \leq |\rho|$, is the subtrace 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.

Given $\rho, \rho' \in \text{Trc}_{\mathcal{K}}$, we denote by $\rho \cdot \rho'$ the concatenation of the traces ρ and ρ' . Finally, if $\text{fst}(\rho) = w_0$ (the initial state of \mathcal{K}), ρ is called an *initial trace*.

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

Definition 3. 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

- $\mathbb{I} = \text{Trc}_{\mathcal{K}}$,
- $A_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \text{lst}(\rho) = \text{fst}(\rho')\}$,
- $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Pref}(\rho)\}$,

- $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Suff}(\rho)\}$,
- $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is such that, for all $\rho \in \mathbb{I}$, $\sigma(\rho) = \bigcap_{w \in \text{states}(\rho)} \mu(w)$.

Relations $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as the Allen’s relations A , B , and E , respectively. Furthermore, according to the definition of σ , a proposition letter $p \in \mathcal{AP}$ holds over $\rho = v_1 \cdots v_n$ if and only if 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 [49]. Satisfiability of an HS formula over a Kripke structure can be given in terms of induced abstract interval models.

Definition 4. Let \mathcal{K} be a finite Kripke structure and ψ be an HS formula. We say that a trace $\rho \in \text{Trc}_{\mathcal{K}}$ satisfies ψ , denoted as $\mathcal{K}, \rho \models \psi$, if and only if it holds that $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

Furthermore, we say that \mathcal{K} models ψ , denoted as $\mathcal{K} \models \psi$, if and only if for all *initial* traces $\rho' \in \text{Trc}_{\mathcal{K}}$ it holds that $\mathcal{K}, \rho' \models \psi$. The *MC problem* for HS over finite Kripke structures is the problem of deciding whether $\mathcal{K} \models \psi$.

It is worth pointing out that every Kripke structure \mathcal{K} induces an abstract interval model, and that only interval models arising from Kripke structures are considered in the MC problem. Such a problem is not trivially decidable, as the set $\text{Trc}_{\mathcal{K}}$ of traces of \mathcal{K} is infinite.

Remark 1. We would like to draw attention to the *branching* semantics of modalities $\langle A \rangle$ and $\langle \bar{A} \rangle$: $\langle A \rangle$ (resp., $\langle \bar{A} \rangle$) allows one to “move” to *any* trace branching on the right (resp., left) of the considered one. As an example, with reference to the Kripke structure \mathcal{K}_3 in Figure 1, given the trace $\rho = v_0 v_0 v_0$, it holds that $\rho A_{\mathbb{I}} v_0 v_1$, $\rho A_{\mathbb{I}} v_0 v_2$, $\rho A_{\mathbb{I}} v_0 v_1 v_1$, $\rho A_{\mathbb{I}} v_0 v_2 v_2$, and so on. Thus, $\mathcal{K}_3, \rho \models \langle A \rangle q \wedge \langle A \rangle p \wedge \langle A \rangle \neg q \wedge \langle A \rangle \neg p$, since $\mathcal{K}_3, v_0 v_1 \models q$, but $\mathcal{K}_3, v_0 v_1 \not\models p$, and $\mathcal{K}_3, v_0 v_2 \models p$, but $\mathcal{K}_3, v_0 v_2 \not\models q$.

We conclude the section with a simple example (a simplified version of the one given in [36]), showing that the fragments investigated in this paper can express meaningful properties of state transition systems in a very compact way, compared to their formulation in standard point-based temporal logics.⁵

Example 2. In Figure 2, we give 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 (denoted by a double circle) 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

⁵We systematically compared the expressive power of interval temporal logic model checking with that of point-based temporal logic one in [9]. We refer the interested reader to such a publication.

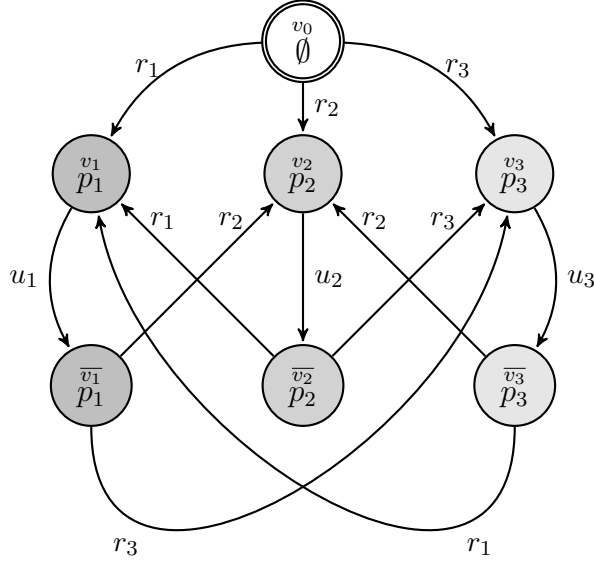


Figure 2: The Kripke structure \mathcal{K}_{Sched} .

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.

We now show how some meaningful properties to be checked over \mathcal{K}_{Sched} can be expressed in the HS fragment $\bar{\text{AE}}$. In all the following formulas, we state the validity of the considered properties over all legal computation sub-intervals by using the modality $[E]$ (as all computation sub-intervals are suffixes of at least one initial trace of the Kripke structure):

- $\mathcal{K}_{Sched} \models [E] (\langle E \rangle^3 \top \rightarrow (\chi(p_1, p_2) \vee \chi(p_1, p_3) \vee \chi(p_2, p_3)))$,
where $\chi(p, q) = \langle E \rangle \langle \bar{A} \rangle p \wedge \langle E \rangle \langle \bar{A} \rangle q$;
- $\mathcal{K}_{Sched} \not\models [E] (\langle E \rangle^{10} \top \rightarrow \langle E \rangle \langle \bar{A} \rangle p_3)$;
- $\mathcal{K}_{Sched} \not\models [E] (\langle E \rangle^5 \rightarrow (\langle E \rangle \langle \bar{A} \rangle p_1 \wedge \langle E \rangle \langle \bar{A} \rangle p_2 \wedge \langle E \rangle \langle \bar{A} \rangle p_3))$.

The first formula requires that at least 2 proposition letters are witnessed in any suffix with length at least 4 of an initial trace. Since a process cannot be executed twice in a row, the formula is satisfied by \mathcal{K}_{Sched} .

The second formula requires that, in any suffix with length at least 11 of an initial trace, process 3 is executed at least once in some internal states (*non starvation*). \mathcal{K}_{Sched} does not satisfy the formula, because the scheduler can defer the execution of a process ad libitum.

The third formula requires that, in any suffix having length at least 6 of an initial trace, 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.

Before summarizing known complexity results about MC for HS and its fragments (under the homogeneity assumption), we recall some relevant notions of complexity theory and we give a short account of some not that well known complexity classes.

2.3. Some complexity classes in the polynomial-time hierarchy

For the sake of completeness, we briefly recall here some notions concerning the polynomial-time hierarchy exploited in the paper.

The *polynomial-time hierarchy*, denoted by **PH**, was introduced by Stockmeyer in [52], and is defined as

$$\mathbf{PH} = \bigcup_{k \in \mathbb{N}} \Delta_k^p,$$

where $\Delta_0^p = \Sigma_0^p = \Pi_0^p = \mathbf{P}$, and, for all $k \geq 1$, $\Delta_k^p = \mathbf{P}^{\Sigma_{k-1}^p}$, $\Sigma_k^p = \mathbf{NP}^{\Sigma_{k-1}^p}$, $\Pi_k^p = \mathbf{co}\text{-}\Sigma_k^p$.

In particular, we have that $\Delta_1^p = \mathbf{P}$, $\Sigma_1^p = \mathbf{NP}$, and $\Delta_2^p = \mathbf{P}^{\mathbf{NP}}$. A well-known example of complete problem for Σ_k^p (resp., Π_k^p) is to decide the truth of fully-quantified formulas of the form $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \phi(x_1, x_2, \dots, x_n)$, where $\phi(x_1, x_2, \dots, x_n)$ is a quantifier-free Boolean formula, whose variables range in the set $\{x_1, x_2, \dots, x_n\}$, $Q_i \in \{\exists, \forall\}$, for all $2 \leq i \leq n$, $Q_1 = \exists$ (resp., $Q_1 = \forall$), and there are $k - 1$ quantifier alternations, that is, $k - 1$ different indexes $j > 1$ such that $Q_j \neq Q_{j-1}$. On the contrary, Δ_k^p does not feature very popular complete problems. As an example, for each $k \geq 1$, a Δ_{k+1}^p -complete problem is to decide whether, given a true quantified Boolean formula of the form

$$\exists x_1 \dots \exists x_r \forall x_{r+1} Q_{r+2} x_{r+2} \dots Q_n x_n \phi(x_1, \dots, x_n),$$

with $k - 1$ quantifier alternations, the lexicographically maximum truth assignment v to the variables $\langle x_1, \dots, x_r \rangle$ such that

$$\forall x_{r+1} Q_{r+2} x_{r+2} \dots Q_n x_n \phi(v(x_1), \dots, v(x_r), x_{r+1}, \dots, x_n)$$

is true assigns 1 to x_r [26].

As a particular case, given a satisfiable Boolean formula $\phi(x_1, \dots, x_n)$, the problem of deciding whether the lexicographically maximum truth assignment to $\langle x_1, \dots, x_n \rangle$ satisfying ϕ assigns 1 to x_n is complete for $\Delta_2^p = \mathbf{P}^{\mathbf{NP}}$. For other examples of $\mathbf{P}^{\mathbf{NP}}$ -complete problems (many of them are related to MC) we refer the reader to [4, 28, 29, 30].

Above **NP** and **co-NP**, but below $\mathbf{P}^{\mathbf{NP}}$, is the class $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$, introduced by Papadimitriou and Zachos in [45], which is the set of problems decided by a deterministic **P** algorithm (Turing machine) which requires only $O(\log n)$ queries to an **NP** oracle (being n the input size). Analogously, $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ is the set of problems decided by a **P** algorithm requiring $O(\log^2 n)$ queries to an **NP** oracle.⁶ These complexity classes (and all others

⁶Here and in the following, we assume that the polynomial hierarchy **PH** is not collapsing, and that $\mathbf{P}^{\mathbf{NP}}$, $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$, and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ are *distinct*, as it is widely conjectured.

which set a bound on the number of allowed queries) are called *bounded query classes*. Note that $\mathbf{P}^{\mathbf{NP}}$, $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ are closed under complementation, as well as under **LOGSPACE** (many-one) reductions.

As for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$, it has been proved (see [16, 56]) that $\mathbf{P}^{\mathbf{NP}[O(\log n)]} = \mathbf{LOGSPACE}^{\mathbf{NP}} = \mathbf{P}_{\parallel}^{\mathbf{NP}}$, where $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ is the class of problems decided by a deterministic \mathbf{P} algorithm which performs a single round (or a *constant* number of rounds) of parallel queries to an \mathbf{NP} oracle. By *parallel queries*, it is intended that each query is independent of the outcome of any other or, equivalently, that all queries have to be formulated before the oracle is consulted. Obviously, the constraint of parallelism is not necessarily fulfilled in the class $\mathbf{P}^{\mathbf{NP}}$, where a query to the oracle may be *adaptive*, that is, it may depend on the results of previously performed queries. An example of complete problem for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ is **PARITY(SAT)**: given a set of Boolean formulas $\Gamma = \{\phi_1, \dots, \phi_n\}$, the problem is to decide if the number of *satisfiable* formulas in Γ is odd or even [55].

As for $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$, it has been proved in [17] that $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]} = \mathbf{P}_{\parallel O(\log n)}^{\mathbf{NP}}$, (in $\mathbf{P}_{\parallel O(\log n)}^{\mathbf{NP}}$ a succession of $O(\log n)$ parallel query rounds are allowed). To the best of our knowledge, the first complete problems for this class were introduced in [50]. Among these complete problems, a detailed account of the problem $\mathbf{TB(SAT)}_{1 \times M}$ will be given in Section 4.

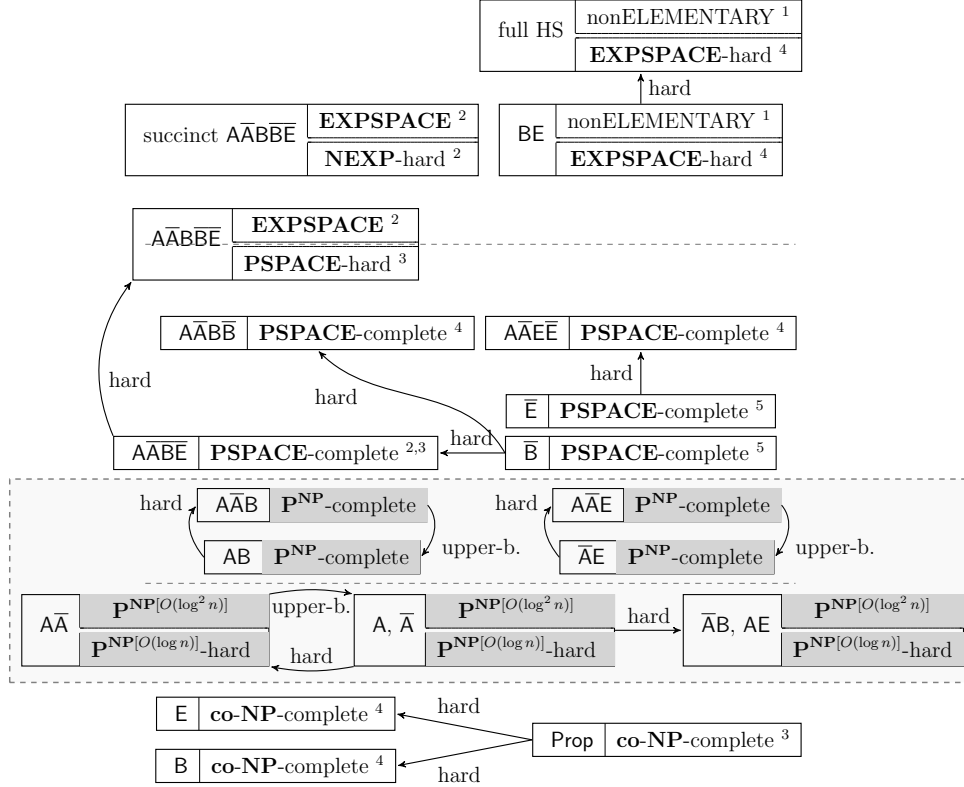
2.4. The general picture

In this section, we give an overview of both known and new results about the complexity of the MC problem for full HS and its proper fragments (under the homogeneity assumption). The results are summarized in Figure 3.

In [36], Molinari et al. proved that, given a Kripke structure \mathcal{K} and a bound k on the structural complexity of HS formulas, that is, on the nesting depth of $\langle \mathbf{E} \rangle$ and $\langle \mathbf{B} \rangle$ 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 MC problem for (full) HS is decidable, providing an algorithm with non-elementary complexity. In [8], **EXSPACE**-hardness of the fragment **BE**, and thus of full HS, has been shown.

The fragments **AABBE** and **AAEBE** have been investigated in [38]. For each of them, an **EXSPACE** MC algorithm has been devised that, for any trace of the Kripke structure, finds a satisfiability-preserving trace of bounded length (*trace representative*). In this way, the MC algorithm needs to check only traces with a given maximum length. **PSPACE**-hardness of MC for **AABBE** and **AAEBE** has been proved in [37] (if a succinct encoding of formulas is exploited, the algorithm remains in **EXSPACE**, but a **NEXP** lower bound can be given [38]). In addition, it has been shown that formulas satisfying a constant bound on the nesting depth of $\langle \mathbf{B} \rangle$ (resp., $\langle \mathbf{E} \rangle$) can be checked in polynomial working space [38].

Finally, the MC problem has been shown to be **PSPACE**-complete for the HS fragments **AABE** [37], **AABB** and **AAEE** [8], and $\bar{\mathbf{B}}$ and $\bar{\mathbf{E}}$ [40], and **co-NP**-complete for the HS fragments **B** and **E** (the same complexity as MC for the purely propositional HS fragment **Prop** [37]). In all these cases, the computational complexity turns out to be comparable with or lower than that of LTL MC, which is known to be **PSPACE**-complete [51]. A comparison of different



¹ [36], ² [38], ³ [37], ⁴ [8], ⁵ [40]

Figure 3: The computational complexity of MC for HS and its fragments: known results are depicted in white boxes, new ones in gray boxes.

semantics for HS, that is, state-based (the one adopted here), linear, and computation-tree-based, together with an expressiveness comparison with the point-based temporal logics LTL, CTL, and CTL*, can be found in [9].

In this paper, we complete the analysis of the sub-fragments of $\overline{A}\overline{A}\overline{B}\overline{B}$ (respectively, $\overline{A}\overline{A}\overline{E}\overline{E}$), which are still expressive enough to capture meaningful interval properties of state transition systems and whose MC problem has a computational complexity markedly lower than that of full HS (see Figure 4). More precisely,

- in Section 3, we devise a \mathbf{P}^{NP} MC algorithm for $\overline{A}\overline{A}\overline{B}$ (resp., $\overline{A}\overline{A}\overline{E}$);
- in Section 4, we provide a $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$ MC algorithm for $\overline{A}\overline{A}$, $\overline{A}\overline{B}$, and $\overline{A}\overline{E}$;
- in Section 5, we prove that MC for $\overline{A}\overline{B}$ (resp., $\overline{A}\overline{E}$) is \mathbf{P}^{NP} -hard, from which it follows that MC for $\overline{A}\overline{B}$, $\overline{A}\overline{E}$, and $\overline{A}\overline{A}\overline{B}$ is \mathbf{P}^{NP} -complete;
- in Section 6, we show that MC for \overline{A} and \overline{A} is $\mathbf{P}^{\text{NP}[O(\log n)]}$ -hard; such a lower bound immediately propagates to $\overline{A}\overline{A}$, $\overline{A}\overline{B}$ and $\overline{A}\overline{E}$ showing that MC for these five fragments is in between $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$ and $\mathbf{P}^{\text{NP}[O(\log n)]}$.

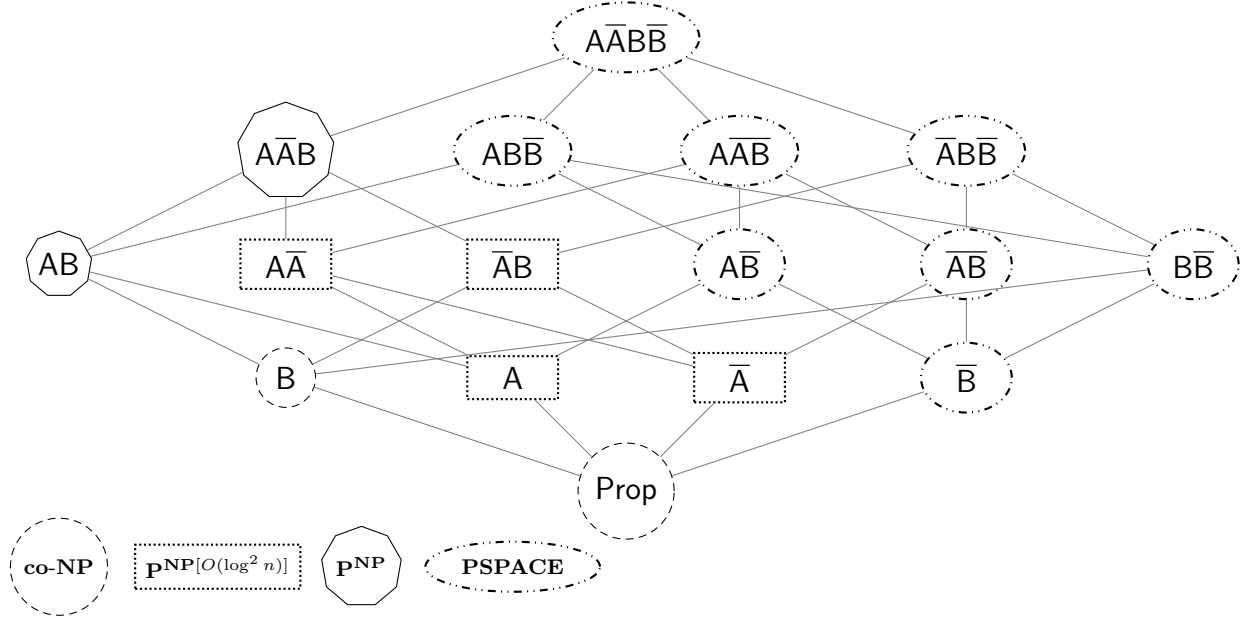


Figure 4: The computational complexity of MC for the sub-fragments of $A\bar{A}\bar{B}\bar{B}$.

It is worth pointing out that the fragment $\bar{A}\bar{B}$ belongs to a lower complexity class than the fragment AB (the same holds for the symmetric fragments $\bar{A}E$ and $\bar{A}\bar{E}$). Such a difference can be intuitively explained by the different expressiveness of the two fragments. Let us consider an AB formula of the form $\langle B \rangle \langle A \rangle \theta$. A trace ρ satisfies $\langle B \rangle \langle A \rangle \theta$ if there exists a prefix $\tilde{\rho}$ of ρ from which a branch, i.e., a trace starting from $\text{lst}(\tilde{\rho})$, satisfying θ departs. Hence, AB allows one to state specific properties of the branches departing from a state occurring in a given path. Such an ability will be exploited in Section 5 to prove the \mathbf{P}^{NP} -hardness of AB . This kind of properties cannot be expressed in the fragment $\bar{A}\bar{B}$. Indeed, for any given trace ρ , modality $\langle \bar{A} \rangle$ only allows one to “select” traces leading to the first state of ρ , and modality $\langle B \rangle$ is of no help: if we consider a prefix $\tilde{\rho}$ of ρ , the set of traces 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 $\langle \bar{A} \rangle$ and $\langle B \rangle$ does not give any advantage in terms of expressiveness. Finally, since A , \bar{A} , and AA are devoid of modalities for prefixes (and suffixes), they analogously belong to $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$.

3. A \mathbf{P}^{NP} MC algorithm for the HS fragments $A\bar{A}\bar{B}$ and $A\bar{A}\bar{E}$

In this section, we present an MC algorithm for $A\bar{A}\bar{B}$ formulas (see the procedure MC reported in Algorithm 1) with complexity in the class \mathbf{P}^{NP} . W.l.o.g., we restrict our attention to $A\bar{A}\bar{B}$ formulas devoid of occurrences of conjunctions and universal modalities (definable, as usual, by means of disjunctions, negations, and existential modalities).

The MC procedure MC for a formula ψ against a Kripke structure \mathcal{K} 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{K} . The function $V_A(\phi, v)$ (resp.,

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

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

$V_{\bar{A}}(\phi, v)$) returns \top if and only if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ starting from the state v (resp., leading to the state v) such that $\mathcal{K}, \rho \models \phi$. MC is initially invoked with parameters $(\mathcal{K}, \neg\psi, \text{FORWARD})$. During the execution, it instantiates the entries of V_A and $V_{\bar{A}}$, which are exploited in order to answer the MC problem $\mathcal{K} \models \psi$. In the end, this is equivalent to checking whether $V_A(\neg\psi, w_0) = \perp$, where w_0 is the initial state of \mathcal{K} .

Let us consider MC in more detail. Along with the Kripke structure \mathcal{K} 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{K} . MC is applied recursively (lines 1–4) 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 $\text{Success}(\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}))$, and it returns \top whenever there exists a computation of the non-deterministic algorithm $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}})$ returning \top , namely, whenever there is a suitable trace starting from, or leading to v (depending on the value of the parameter DIRECTION), and satisfying ψ .

We define now the set of $A\bar{A}$ -modal subformulas of ψ ($\text{ModSubf}_{A\bar{A}}(\psi)$) used to “direct” the recursive calls of MC (lines 1–4).

Definition 5. The set $\text{ModSubf}_{A\bar{A}}(\psi)$ of $A\bar{A}$ -modal subformulas of an $A\bar{A}B$ formula ψ is the set of subformulas of ψ either of the form $\langle A \rangle \psi'$ or of 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*.

As an example, it holds that

- $\text{ModSubf}_{A\bar{A}}(\langle A \rangle \langle \bar{A} \rangle q) = \{\langle A \rangle \langle \bar{A} \rangle q\}$, 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 the set $\text{ModSubf}_{A\bar{A}}(\psi)$ (lines 1–4). In this way, we can recursively gather in the Boolean vectors V_A and $V_{\bar{A}}$, by increasing nesting 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 .

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

```
1:  $\tilde{\rho} \leftarrow \text{A\_trace}(\mathcal{K}, v, |W| \cdot (2|\psi| + 1)^2, \text{DIRECTION}) \triangleleft$  a trace of  $\mathcal{K}$  from/to  $v$  having 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  $\langle \bar{A} \rangle \phi \in \text{ModSubf}_{A\bar{A}}(\psi)$  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  $\varphi$  of  $\psi$ , not contained in (or equal to)  $A\bar{A}$ -modal subformulas of  $\psi$ ,
   by increasing length do
9:   if  $\varphi = p$ , 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  $\varphi = \neg\varphi_1$  then
14:     for  $i = 1, \dots, |\tilde{\rho}|$  do
15:        $T[\varphi, i] \leftarrow \text{not } T[\varphi_1, i]$ 
16:   else if  $\varphi = \varphi_1 \vee \varphi_2$  then
17:     for  $i = 1, \dots, |\tilde{\rho}|$  do
18:        $T[\varphi, i] \leftarrow T[\varphi_1, i]$  or  $T[\varphi_2, i]$ 
19:   else if  $\varphi = \langle B \rangle \varphi_1$  then
20:      $T[\varphi, 1] \leftarrow \perp$ 
21:     for  $i = 2, \dots, |\tilde{\rho}|$  do
22:        $T[\varphi, i] \leftarrow T[\varphi, i - 1]$  or  $T[\varphi_1, i - 1]$ 
23: return  $T[\psi, |\tilde{\rho}|]$ 
```

Let us now consider the *non-deterministic polynomial time* procedure Oracle($\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}}$) (see Algorithm 2), which is used as the “basic engine” by the oracle in the aforementioned MC Algorithm 1. The idea underlying Algorithm 2 is to first non-deterministically generate a trace $\tilde{\rho}$ 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 [8], which states a “polynomial-size model-trace property” for formulas of the fragment $A\bar{A}\bar{B}\bar{B}$:

Proposition 1 ([8], consequence of Theorem 10). *Let ρ be a trace of a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and ϕ be an $A\bar{A}\bar{B}\bar{B}$ formula such that $\mathcal{K}, \rho \models \phi$. Then, there exists $\rho' \in \text{Trc}_{\mathcal{K}}$ 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 traces whose length is bounded by $|W| \cdot (2|\phi| + 1)^2$.

An execution of $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_A \cup V_{\bar{A}})$ starts (line 1) by *non-deterministically* generating a trace $\tilde{\rho}$ (having length 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 trace is generated by visiting the unravelling of \mathcal{K} (resp., of \mathcal{K} with transposed edges). The remaining part of the algorithm *deterministically* checks whether $\mathcal{K}, \tilde{\rho} \models \psi$ or not. 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 if $\mathcal{K}, \tilde{\rho}(1, i) \models \phi$, 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 sub-formulas, 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).

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

The next lemma states soundness and completeness of the procedure Oracle , see Algorithm 2 (its proof is in the Appendix A.1).

Lemma 2. *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, ψ be an $A\bar{A}B$ formula, and $V_A(\cdot, \cdot)$ and $V_{\bar{A}}(\cdot, \cdot)$ be two Boolean arrays. Let us assume that*

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

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

- *there is $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$, when DIRECTION is FORWARD ;*
- *there is $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$, when DIRECTION is BACKWARD .*

The following theorem states soundness and completeness of the model checking procedure MC (Algorithm 1).

Theorem 3. *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, ψ be an $A\bar{A}B$ formula, and $V_A(\cdot, \cdot)$ and $V_{\bar{A}}(\cdot, \cdot)$ be two Boolean arrays. If $\text{MC}(\mathcal{K}, \psi, \text{DIRECTION})$ is executed, then for all $v \in W$:*

- *if DIRECTION is FORWARD , $V_A(\psi, v) = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$;*

- if DIRECTION is BACKWARD, $V_{\bar{A}}(\psi, v) = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$.

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

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

Otherwise, $n > 0$ and the formula ψ contains at least an $\langle A \rangle$ or an $\langle \bar{A} \rangle$ modality, and thus $\text{ModSubf}_{\text{A}\bar{\text{A}}}(\psi) \neq \emptyset$. Since each recursive call to MC (either at line 2 or 4) is performed on a formula ϕ featuring a number of occurrences of $\langle A \rangle$ and $\langle \bar{A} \rangle$ 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 $\langle A \rangle \phi \in \text{ModSubf}_{\text{A}\bar{\text{A}}}(\psi)$ and $v' \in W$, $V_{\text{A}}(\phi, v') = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v'$ and $\mathcal{K}, \rho \models \phi$;
2. for each $\langle \bar{A} \rangle \phi \in \text{ModSubf}_{\text{A}\bar{\text{A}}}(\psi)$ and $v' \in W$, $V_{\bar{\text{A}}}(\phi, v') = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v'$ and $\mathcal{K}, \rho \models \phi$.

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

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

Corollary 4. *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure and ψ be an $\text{A}\bar{\text{A}}\text{B}$ formula. If $\text{MC}(\mathcal{K}, \neg\psi, \text{FORWARD})$ is executed, then $V_{\text{A}}(\neg\psi, w_0) = \perp$ if and only if $\mathcal{K} \models \psi$.*

Corollary 5. *The MC 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 $\text{A}\bar{\text{A}}\text{B}$ 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 it is due to the search of ψ for its modal subformulas (lines 1–4), and the second one to 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 $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\text{A}} \cup V_{\bar{\text{A}}})$, 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

By a straightforward adaptation of the procedure Oracle, it is easy to prove that also the MC problem for the symmetric fragment $\text{A}\bar{\text{A}}\text{E}$ is in \mathbf{P}^{NP} . As we will show in Section 5, both problems are actually *complete for \mathbf{P}^{NP}* .

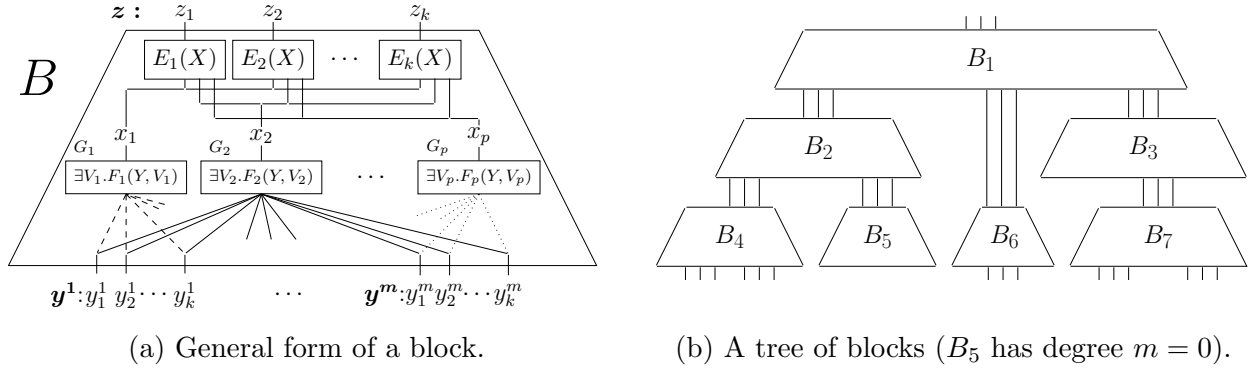


Figure 5: A block (a) and a tree of blocks (b).

4. A $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$ MC algorithm for the HS fragments $\overline{\mathbf{AA}}$, $\overline{\mathbf{AB}}$, and $\overline{\mathbf{AE}}$

In this section, we first propose an MC algorithm for the fragment $\overline{\mathbf{AA}}$ with complexity $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$, thus lower than the complexity of the one described in the previous section for $\overline{\mathbf{AB}}$. As a matter of fact, we do not directly devise an MC algorithm; we proceed instead *via a reduction to* a $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$ -complete problem, namely, $\text{TB}(\text{SAT})_{1 \times M}$ (a restriction of $\text{TB}(\text{SAT})$, see [50]), whose instances are complex circuits where some of the gates are endowed with \mathbf{NP} oracles.

4.1. The problem $\text{TB}(\text{SAT})_{1 \times M}$

In order to introduce $\text{TB}(\text{SAT})$, we need to preliminarily describe its basic component, which is the *block*. A block B (see Figure 5a) is a circuit whose input lines are organized in m bit vectors $\mathbf{y}^1, \dots, \mathbf{y}^m$, each of which has k entries, namely $\mathbf{y}^i = (y_1^i, \dots, y_k^i)$. The values m and k are respectively called the *degree* and the *width* of B . The input lines are connected to p internal gates G_1, \dots, G_p . Each gate G_i features a Boolean formula $F_i(Y, V_i)$ associated with it, where $Y = \{y_s^j \mid j = 1, \dots, m, s = 1, \dots, k\}$ and V_i is a set of private variables of F_i , not occurring in any other F_j , with $j \neq i$, that is, $V_i \cap V_j = \emptyset$ for $j \neq i$. The gate G_i queries a SAT oracle in order to decide whether the associated Boolean formula is satisfiable. The output of G_i is denoted by x_i , and it evaluates to \top if and only if $F_i(Y, V_i)$ is satisfiable. Finally, k classic circuits (without oracles) E_1, \dots, E_k compute, from $X = \{x_1, \dots, x_p\}$, their outputs z_1, \dots, z_k , which are also the final k outputs of the block B .

The size of B is defined as the total number of gates, plus the lengths of all the associated Boolean formulas. In the following, to make clear that a gate G_i (respectively, input y_i , block output z_i , gate output x_i) is an element of a block B , we write $B(G_i)$ (respectively, $B(y_i)$, $B(z_i)$, $B(x_i)$).

Given the $k \cdot m$ input bits, determining the output value of any z_i is a $\mathbf{P}_{\parallel}^{\text{NP}}$ problem: the p queries associated with the oracle gates—which determine the outputs x_j 's—can be performed in parallel (they are independent of each other) and then the value of the block output z_i can be calculated in deterministic polynomial time.

Blocks of *the same width* can be combined together to form a tree-structured complex circuit, called a *tree of blocks*. See Figure 5b for an example. Every block in the tree-

structure has a level: blocks which are leaves of the tree are at level 1; a block B_i whose inputs depend on (at least) a block B_j at level $d - 1$ and possibly on other blocks at levels less than d , is at level d . In Figure 5b, B_4, B_5, B_6, B_7 are at level 1, B_2 and B_3 at level 2, and B_1 at level 3. If the root of the tree-structure T is at level d , the k outputs of T can be determined by d rounds of parallel queries: all the queries relative to blocks placed at the same level d' can be answered in parallel once all those at level $d' - 1$ have been answered.

TB(SAT) is the problem of deciding whether a specific output z_i of (the root of) a tree-structure of blocks T is \top or \perp , given the values for the inputs (of the leaf blocks) of T . As proved in [50], the problem TB(SAT) is $\mathbf{P}^{\mathbf{NP}}$ -complete.

The problem $\text{TB(SAT)}_{1 \times M}$ is a constrained version of TB(SAT): any Boolean formula (SAT query) associated with a block B of the tree-structure must have the following form:

$$\exists \ell_1, \dots, \ell_m \in \{1, \dots, k\} \exists V'_i . F_i(y_{\ell_1}^1, \dots, y_{\ell_m}^m, \ell_1, \dots, \ell_m, V'_i),$$

where m and k are respectively the degree and the width of B . This amounts to say that F_i can use *only one bit from each input vector* of B (no matter which), hence “ $1 \times M$ ”. The existential quantification over the indexes ℓ_1, \dots, ℓ_m is an abuse of notation borrowed from [50]: $\exists \ell_j \in \{1, \dots, k\}$ is just a shorthand for k bits (belonging to the set of private variables) “ $\ell_j = 1$ ”, \dots , “ $\ell_j = k$ ”, among which exactly one is \top . In the formula above, V'_i is V_i deprived of such bits.

In [50], it is proved that $\text{TB(SAT)}_{1 \times M}$ is a $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ -complete problem. In particular, the proof of membership to $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ exploits the *squeeze* technique of [26] applied to TREE(SAT) instances. The particular form “ $1 \times M$ ” of the queries allows us to “reshape” the tree-structure of blocks, in such a way that the height becomes logarithmic in the number of blocks. Therefore, only $O(\log n)$ rounds of parallel queries are needed, allowing us to prove the membership of the problem to $\mathbf{P}_{||O(\log n)}^{\mathbf{NP}} = \mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$.

4.2. The reduction of the MC problem for $\overline{\mathbf{A}\mathbf{A}}$ formulas to $\text{TB(SAT)}_{1 \times M}$

Let us show now how to reduce the MC problem for $\overline{\mathbf{A}\mathbf{A}}$ formulas to $\text{TB(SAT)}_{1 \times M}$. As in the previous section, w.l.o.g., we assume that only *existential* modalities occur in the $\overline{\mathbf{A}\mathbf{A}}$ formula ψ to be checked over a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$, with $W = \{w_1, \dots, w_{|W|}\}$.⁷ We consider its negation $\neg\psi$ and build from it a tree-structure of blocks $T_{\mathcal{K}, \neg\psi}$. Each block of $T_{\mathcal{K}, \neg\psi}$ has a type, FORWARD or BACKWARD, and it is associated with a subformula of $\neg\psi$. The root block, B_{root} , is always of type FORWARD and it is associated with $\neg\psi$. Each block B has an output line z_i for each state $w_i \in W$, thus the *width* of all blocks is $k = |W|$.

Starting from B_{root} , $T_{\mathcal{K}, \neg\psi}$ is built by recursive applications of the following *basic step*, which are guided by the $\overline{\mathbf{A}\mathbf{A}}$ -modal subformulas (recall Definition 5): if some (generic) block B is associated with a formula φ , then

⁷Here, for technical reasons, we assume an arbitrary order of the states of the Kripke structure, $w_1, \dots, w_{|W|}$, where w_1 is the initial state.

- for every $\phi \in \text{ModSubf}_{\overline{\text{AA}}}(\varphi)$, where $\phi = \langle \text{A} \rangle \xi$, we create a FORWARD child B' of B associated with ξ , and
- for every $\phi' \in \text{ModSubf}_{\overline{\text{AA}}}(\varphi)$, where $\phi' = \langle \overline{\text{A}} \rangle \xi'$, we create a BACKWARD child B'' of B associated with ξ' .

Then, the basic step is recursively applied to all the generated children of B , terminating when $\text{ModSubf}_{\overline{\text{AA}}}(\varphi) = \emptyset$. Note that a block B associated with a formula φ has *degree* $m = |\text{ModSubf}_{\overline{\text{AA}}}(\varphi)|$.

In such a way, we determine the tree-structure of blocks $T_{\mathcal{X}, \neg\psi}$. We now describe the internal structure of blocks.

As a preliminary step, we suitably transform $\overline{\text{AA}}$ formulas φ into Boolean ones by replacing all the occurrences of proposition letters and modal subformulas in φ by Boolean variables, as described by the next definition.

Definition 6. Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$ be a finite Kripke structure and let χ be an $\overline{\text{AA}}$ formula. We define $\overline{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})$, where $V_{\mathcal{AP}} = \{v_p \mid p \in \mathcal{AP}\}$ and $V_{\text{modSubf}} = \{v_{\chi'} \mid \chi' \in \text{ModSubf}_{\overline{\text{AA}}}(\chi)\}$ are sets of Boolean variables, as the *Boolean* formula obtained from χ by replacing

- each (occurrence of a) $\overline{\text{AA}}$ -modal subformula $\chi' \in \text{ModSubf}_{\overline{\text{AA}}}(\chi)$ by the variable $v_{\chi'}$,
- and then each (occurrence of a) proposition letter $p \in \mathcal{AP}$ by the variable v_p .

Given a trace $\rho \in \text{Trc}_{\mathcal{K}}$ and an $\overline{\text{AA}}$ formula χ , it is easy to prove (by induction on the complexity of $\overline{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})$) that if ω is an interpretation of the variables of $V_{\mathcal{AP}} \cup V_{\text{modSubf}}$ such that $\omega(v_p) = \top \iff \mathcal{K}, \rho \models p$, for all $p \in \mathcal{AP}$, and $\omega(v_{\chi'}) = \top \iff \mathcal{K}, \rho \models \chi'$, for all $\chi' \in \text{ModSubf}_{\overline{\text{AA}}}(\chi)$, then it holds that $\mathcal{K}, \rho \models \chi \iff \omega(\overline{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$.

We are now ready to describe the internal structure of a block B for a formula φ in $T_{\mathcal{X}, \neg\psi}$. Let us assume that B has type FORWARD and let us refer to the block depicted in Figure 5a for the description. The block features a gate G_i , with $1 \leq i \leq |W|$, for each state of \mathcal{K} . Each output line z_i of B is directly linked to the output x_i of the oracle gate G_i , avoiding the use of circuits E_1, \dots, E_k .

Now, let $F_i(Y, V)$ be the Boolean formula for the gate G_i , with $1 \leq i \leq |W|$ (for the sake of simplicity, we write V instead of V_i). The basic idea is that $F_i(Y, V)$ is satisfiable if and only if there is a trace having length at most $|W|^2 + 2$, starting from the i -th state of W , which satisfies $\overline{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}})$, where φ is the formula associated with the block B . To check the existence of such a witness trace, we need a set of private variables $V_{\text{trace}} = \{v_1^1, \dots, v_{|W|}^1, v_1^2, \dots, v_{|W|}^2, \dots, v_1^{|W|^2+2}, \dots, v_{|W|}^{|W|^2+2}\}$. In particular, the subset of variables $v_1^j, \dots, v_{|W|}^j$, with $1 \leq j \leq |W|^2 + 2$, is used to “encode” the state in the j -th position of the trace. The encoding requires that exactly one variable v_k^j of the subset is assigned to \top , for $1 \leq k \leq |W|$, meaning that the k -th state of \mathcal{K} occurs in the j -th position of the sequence. Moreover, we use a set of private variables $V_{\text{last}} = \{v_1, v_2, \dots, v_{|W|}\}$ which are

used to encode the last state of the witness trace (note that the length of the witness trace can be actually less than the bound $|W|^2 + 2$).

In detail, the Boolean formula $trace(V_{trace}, V_{last}, V_{\mathcal{AP}})$, which ensures that a truth assignment of the private variables V_{trace} properly encodes a trace ρ of \mathcal{X} of length ℓ , for $1 \leq \ell \leq |W|^2 + 2$, is as follows.

$$\begin{aligned}
trace(V_{trace}, V_{last}, V_{\mathcal{AP}}) = & \\
\bigvee_{\ell=1}^{|W|^2+2} & \left[\bigwedge_{t=1}^{\ell} one_t(v_1^t, v_2^t, \dots, v_{|W|}^t) \wedge \bigwedge_{t=1}^{\ell-1} edge_t(v_1^t, \dots, v_{|W|}^t, v_1^{t+1}, \dots, v_{|W|}^{t+1}) \wedge \underbrace{\bigwedge_{t=1}^{|W|} (v_t^\ell \leftrightarrow v_t)}_{(1)} \wedge \right. \\
& \left. \underbrace{\bigwedge_{p \in \mathcal{AP}} \left((v_p \rightarrow \bigwedge_{t=1}^{\ell} \bigwedge_{j=1}^{|W|} (v_j^t \rightarrow VAL(w_j, p))) \wedge (\neg v_p \rightarrow \bigvee_{t=1}^{\ell} \bigwedge_{j=1}^{|W|} (v_j^t \rightarrow \neg VAL(w_j, p))) \right)}_{(2)} \right].
\end{aligned}$$

For any $1 \leq t \leq \ell$, being ℓ the length of the witness trace, the Boolean formula $one_t(v_1^t, v_2^t, \dots, v_{|W|}^t)$ “checks” that the variables $v_1^t, v_2^t, \dots, v_{|W|}^t$ encode (exactly) one state for the t -th element of the trace:

$$one_t(v_1^t, v_2^t, \dots, v_{|W|}^t) = \left(\bigvee_{j=1}^{|W|} v_j^t \right) \wedge \left(\bigwedge_{j=1}^{|W|} \bigwedge_{k=j+1}^{|W|} \neg(v_j^t \wedge v_k^t) \right).$$

Then, for any $1 \leq t \leq \ell - 1$, the formula $edge_t(v_1^t, \dots, v_{|W|}^t, v_1^{t+1}, \dots, v_{|W|}^{t+1})$ checks that if w_k and w_j are states which occur consecutively in the encoded trace (v_k^t and v_j^{t+1} are set to \top), then $(w_k, w_j) \in \delta$:

$$edge_t(v_1^t, \dots, v_{|W|}^t, v_1^{t+1}, \dots, v_{|W|}^{t+1}) = \bigvee_{(w_k, w_j) \in \delta} (v_k^t \wedge v_j^{t+1}).$$

Then, conjunct (1) ensures that the private variables in V_{last} encode the last state of the witness trace, that is, the ℓ -th state. Finally, conjunct (2) enforces the homogeneity assumption, ensuring that a variable $v_p \in V_{\mathcal{AP}}$ evaluates to \top if and only if p holds over all the states of the witness trace ($VAL(w_j, p)$ is just a shorthand for \top if $p \in \mu(w_j)$, and \perp otherwise).

Now, taking the set of private variables $V = V_{last} \cup V_{trace} \cup V_{\mathcal{AP}} \cup V_{modSubf}$, the Boolean formula $F_i(Y, V)$ for the gate G_i is formally defined as:

$$\begin{aligned}
F_i(Y, V) = & v_i^1 \wedge \bar{\varphi}(V_{\mathcal{AP}}, V_{modSubf}) \wedge trace(V_{trace}, V_{last}, V_{\mathcal{AP}}) \wedge \\
& \bigwedge_{\langle A \rangle \xi \in \text{ModSubf}_{A\bar{A}}(\varphi)} (v_{\langle A \rangle \xi} \leftrightarrow \bigvee_{j=1}^{|W|} (v_j \wedge y_j^\xi)) \quad \wedge \quad \bigwedge_{\langle \bar{A} \rangle \xi' \in \text{ModSubf}_{A\bar{A}}(\varphi)} (v_{\langle \bar{A} \rangle \xi'} \leftrightarrow \bigvee_{j=1}^{|W|} (v_j^1 \wedge y_j^{\xi'}))
\end{aligned}$$

The first conjunct of $F_i(Y, V)$ requires that the witness trace starts with the i -th state of \mathcal{K} . The fourth one requires that each private variable $v_{\langle A \rangle \xi}$, for $\langle A \rangle \xi \in \text{ModSubf}_{\overline{\text{AA}}}(\varphi)$ has exactly the same truth assignment as the j -th output, y_j^ξ , of the block for ξ (which is a child of B)—provided that the final state of the trace is the j -th state of \mathcal{K} . Since exactly one among the variables of $V_{\text{last}} = \{v_1, \dots, v_{|W|}\}$ is set to \top , it is guaranteed that at most one bit for every child-block is considered by B , thus fulfilling the restriction of $\text{TB}(\text{SAT})_{1 \times M}$. The last conjunct of $F_i(Y, V)$ forces the symmetric constraint for modal subformulas of the form $\langle \overline{A} \rangle \xi'$.

The formula $F_i(Y, V)$ for a gate G_i in a BACKWARD block is very similar: we just need to replace the first conjunct v_i^1 by v_i .

The following proposition states the correctness of the encoding for traces.

Proposition 6. *Given a trace $\rho \in \text{Trc}_{\mathcal{K}}$, with $|\rho| \leq |W|^2 + 2$, there exists a truth assignment ω to the variables in V which satisfies the formula $\text{trace}(V_{\text{trace}}, V_{\text{last}}, V_{\mathcal{AP}})$, and*

- for any $1 \leq t \leq |\rho|$ and $1 \leq j \leq |W|$, $\rho(t) = w_j \iff \omega(v_j^t) = \top$ and $\omega(v_j^{|\rho|}) = \omega(v_j)$;
- for any $p \in \mathcal{AP}$, $\omega(v_p) = \top \iff \mathcal{K}, \rho \models p$.

Conversely, if a truth assignment ω to the variables in V satisfies the s -th disjunct of $\text{trace}(V_{\text{trace}}, V_{\text{last}}, V_{\mathcal{AP}})$, then there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$, with $|\rho| = s$, such that

- for any $1 \leq t \leq |\rho|$ and $1 \leq j \leq |W|$, $\rho(t) = w_j \iff \omega(v_j^t) = \top$;
- for any $p \in \mathcal{AP}$, $\mathcal{K}, \rho \models p \iff \omega(v_p) = \top$.

The following theorem states the correctness of the construction of $T_{\mathcal{K}, \neg\psi}$ (the proof is given in Appendix A.2).

Theorem 7. *Let ψ be an $\overline{\text{AA}}$ formula and $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$ be a finite Kripke structure. For every block B of $T_{\mathcal{K}, \neg\psi}$, if B is associated with an $\overline{\text{AA}}$ formula φ , then*

- if B is a FORWARD block, for all $i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ if and only if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$;
- if B is a BACKWARD block, for all $i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ if and only if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$.

The two next corollaries immediately follow.

Corollary 8. *Let ψ be an $\overline{\text{AA}}$ formula, \mathcal{K} be a finite Kripke structure, and B_{root} be the root block of $T_{\mathcal{K}, \neg\psi}$. Then, it holds that $B_{\text{root}}(z_1) = \perp \iff \mathcal{K} \models \psi$.*

Corollary 9. *The MC problem for $\overline{\text{AA}}$ formulas over finite Kripke structures belongs to $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$.*

PROOF. The result follows from Corollary 8 and the fact that the instance of $\text{TB}(\text{SAT})_{1 \times M}$ generated from an $\overline{\text{AA}}$ formula ψ and a Kripke structure \mathcal{K} is polynomial in $|\psi|$ and $|\mathcal{K}|$. \square

4.3. The reduction of the MC problem for $\overline{\text{AB}}$ (resp., AE) formulas to $\text{TB}(\text{SAT})_{1 \times M}$

We conclude the section by showing that it is possible to adapt the above-described reduction to the fragment $\overline{\text{AB}}$ and the symmetric fragment AE . Let us focus on $\overline{\text{AB}}$ (the case for AE can be dealt with in an analogous way).

Having in mind that $\overline{\text{AB}}$ is a fragment of AAB , by removing the case for the modality $\langle \text{A} \rangle$ in Algorithm 2, we get a procedure for which Lemma 2 still holds. Since Algorithm 2 is in NP , there must exist a reduction to SAT, that is, given an instance $(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})$ for Oracle , there exists a Boolean formula $\Psi_{(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})}$, which depends on $(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})$, that is satisfiable if and only if $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})$ admits a successful computation on the given input. By Lemma 2, this is the case if and only if:

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

provided that for each $\langle \overline{\text{A}} \rangle \phi \in \text{ModSubf}_{\text{A}\overline{\text{A}}}(\psi)$ and $v' \in W$, $V_{\overline{\text{A}}}(\phi, v') = \top$ if and only if there exists $\rho' \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho') = v'$ and $\mathcal{K}, \rho' \models \phi$.

The idea is that such a formula $\Psi_{(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})}$ can be used as a SAT query for an oracle gate G_i in a generic block associated with the formula ψ . The role of the global Boolean vector $V_{\overline{\text{A}}}$ is instead played by the local communication among blocks in the tree-structure; $\Psi_{(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})}$ basically has the same structure as the Boolean formula $F_i(Y, V)$, with some minor differences which are outlined below.

First of all, we observe that Algorithm 2 works with traces whose length is bounded by $|W| \cdot (2|\psi| + 1)^2$ (instead of $|W|^2 + 2$ as in $F_i(Y, V)$). A trace is then encoded exactly as in $F_i(Y, V)$ by using a set of private variables $V_{\text{trace}} = \{v_j^t \mid j = 1, \dots, |W|, t = 1, \dots, |W| \cdot (2|\psi| + 1)^2\}$; $\Psi_{(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})}$ has also to encode the Boolean table T of Oracle with entries $T[\varphi, i]$, where φ is a subformula of ψ , and $1 \leq i \leq |W| \cdot (2|\psi| + 1)^2$ is the length of a prefix of the considered trace. Therefore, there is a variable $x_{\varphi, i}$ in $\Psi_{(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\overline{\text{A}}})}$ for each entry $T[\varphi, i]$, with the intuitive meaning that $x_{\varphi, i}$ is assigned \top if and only if $T[\varphi, i] = \top$. Actually, in this encoding we do not need any entry for a modal subformula $\langle \overline{\text{A}} \rangle \xi$, whose truth value is conveyed by y_j^ξ , namely, the j -th input connected to the child block for the subformula ξ (assuming that the starting state of the trace is the j -th state of \mathcal{K}). It is worth noting that this construction is possible since all the prefixes of the trace ρ encoded by the assignment to V_{trace} , and ρ itself, share the same starting point, and thus agree on the truth value of any $\overline{\text{A}}$ -modal subformula. The most relevant consequence of this property is that the constraint of $\text{TB}(\text{SAT})_{1 \times M}$ on the form of SAT queries is respected.

As for the construction of $T_{\mathcal{K}, \neg\psi}$, it is exactly as before where, in particular, the root block B_{root} has type FORWARD, and all the other blocks have type BACKWARD. The following result can be stated.

Theorem 10. *The MC problem for $\overline{\text{AB}}$ (resp., AE) formulas over finite Kripke structures belongs to $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$.*

The construction we have sketched cannot be adapted to the fragment **AB**. This is due to the fact that the *right* endpoints of the prefixes of a trace differ in general, and thus they do not necessarily agree on the truth value of **A**-modal subformulas, hence the restriction of $\text{TB}(\text{SAT})_{1 \times M}$ on the form of SAT queries cannot be respected. In the next section, we will prove that MC for **AB** formulas is indeed *inherently* more difficult than MC for $\overline{\text{AB}}$.

5. \mathbf{P}^{NP} -hardness of MC for the HS fragments **AB** and $\overline{\text{AE}}$

In this section, we show that the \mathbf{P}^{NP} -complete problem of Sequentially Nested SATisfiability (SNSAT, [28]) can be reduced to the MC problem for formulas of the fragment **AB** (and similarly $\overline{\text{AE}}$) thus proving that the problem is hard for the class \mathbf{P}^{NP} . SNSAT is a logical problem with nested satisfiability questions defined as follows.

Definition 7. An instance \mathcal{I} of SNSAT consists of a set of n Boolean variables $X = \{x_1, \dots, x_n\}$ and a set of n 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 the set of private variables $Z_i = \{z_i^1, \dots, z_i^{j_i}\}$, that is, $Z_i \cap Z_j = \emptyset$, for $j \neq i$, and $X \cap Z_i = \emptyset$. We denote by $|\mathcal{I}|$ the size $|X| = n$.

Let $v_{\mathcal{I}}$ be a truth-assignment of the variables in X defined as follows:

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

(by a suitable truth-assignment to the private 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 truth-assignment $v_{\mathcal{I}}$ 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 $v_{\mathcal{I}}(x_1) = \top$ if and only if $F_1(Z_1)$ 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, yielding the value of $v_{\mathcal{I}}(x_2)$. This step is iterated other $n - 2$ times, finally obtaining the value of $v_{\mathcal{I}}(x_n)$.

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 [28].

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 6a, assuming that the labeling of states (nodes) is defined as follows:

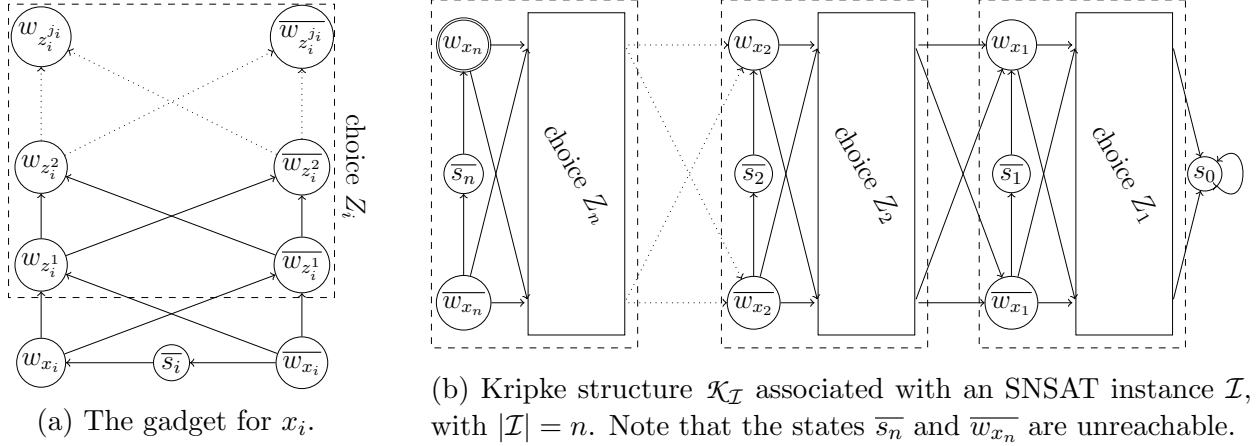


Figure 6: A gadget (a) and the resulting Kripke structure (b).

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

The Kripke structure $\mathcal{K}_{\mathcal{I}}$ is obtained by sequentializing (adding suitable edges) 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 6b. Formally, we have

$$\mathcal{K}_{\mathcal{I}} = (X \cup Z \cup \{s, t\} \cup R \cup \{p_{\bar{x}_i} \mid i = 1, \dots, n\}, W, \delta, \mu, w_{x_n}).$$

The Kripke structure $\mathcal{K}_{\mathcal{I}}$ features the following properties:

- any trace satisfying s does not pass through any \bar{s}_i , for $1 \leq i \leq n$;
- any trace *not* satisfying t has s_0 as its last state;
- any trace *not* satisfying r_i passes through some state of the i -th gadget, for $1 \leq i \leq n$;
- the only trace satisfying $p_{\bar{x}_i}$ is \bar{w}_{x_i} (note that $|\bar{w}_{x_i}| = 1$), for $1 \leq i \leq n$.

A trace $\rho \in \text{Trc}_{\mathcal{K}_{\mathcal{I}}}$ induces a truth assignment of all the proposition letters, denoted by ω_ρ , which is defined as $\omega_\rho(y) = \top \iff \mathcal{K}_{\mathcal{I}}, \rho \models y$, for any letter y . In the following, we shall write $\omega_\rho(Z_i)$ for $\omega_\rho(z_i^1), \dots, \omega_\rho(z_i^{j_i})$. 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 $\overline{w_{z_m^{u_m}}} \in \text{states}(\rho)$, then $\omega_\rho(z_m^{u_m}) = \perp$.

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

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

where $\ell_{=2} = \langle B \rangle \top \wedge [B] \perp$ is satisfied only by traces of length 2. The first conjunct of φ_k , i.e, $s \wedge \neg t$, forces the trace to reach the collector state s_0 , without visiting any state $\overline{s_j}$. The second conjunct checks that if the trace 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 trace assigns the truth value \perp to some x_m by passing through $\overline{w_{x_m}}$, 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 $\overline{w_{x_m}}$. Since $\bigvee_{i=1}^n \langle A \rangle p_{x_i}$ is satisfied by $\tilde{\rho}$, $\langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\ell_{=2} \wedge \neg \psi_{k-1}))$ must be satisfied as well. The only possibility is that the trace $\overline{s_m} \cdot w_{x_m}$ does not model ψ_{k-1} , as $\overline{w_{x_m}} \cdot \overline{s_m}$ has to model $\langle A \rangle (\ell_{=2} \wedge \neg \psi_{k-1})$. However, since $\psi_{k-1} = \langle A \rangle \varphi_{k-1}$, this holds if and only if $\mathcal{K}, w_{x_m} \not\models \psi_{k-1}$.

The following theorem states the correctness of the construction. Its proof can be found in Appendix A.3.

Theorem 11. *Let \mathcal{I} be an instance of SNSAT, with $|\mathcal{I}| = n$, and let $\mathcal{K}_{\mathcal{I}}$ and $\mathcal{F}_{\mathcal{I}}$ 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 $v_{\mathcal{I}}(x_r) = \top \iff \mathcal{K}_{\mathcal{I}}, w_{x_r} \models \psi_k$;
2. if $k \geq r+1$, then $v_{\mathcal{I}}(x_r) = \perp \iff \mathcal{K}_{\mathcal{I}}, \overline{w_{x_r}} \models \psi_k$.

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

Corollary 12. *Let \mathcal{I} be an instance of SNSAT, with $|\mathcal{I}| = n$, and let $\mathcal{K}_{\mathcal{I}}$ and $\mathcal{F}_{\mathcal{I}}$ be defined as above. Then,*

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

PROOF. By Theorem 11, $v_{\mathcal{I}}(x_n) = \top \iff \mathcal{K}_{\mathcal{I}}, w_{x_n} \models \psi_n$. If $v_{\mathcal{I}}(x_n) = \top$ then $\mathcal{K}_{\mathcal{I}}, w_{x_n} \models \psi_n$ and, since w_{x_n} is the only initial trace satisfying $[B] \perp$ (this formula is satisfied by traces having length equal to 1 only), $\mathcal{K}_{\mathcal{I}} \models [B] \perp \rightarrow \psi_n$. Conversely, if $\mathcal{K}_{\mathcal{I}} \models [B] \perp \rightarrow \psi_n$, then $\mathcal{K}_{\mathcal{I}}, w_{x_n} \models \psi_n$, allowing us to conclude that $v_{\mathcal{I}}(x_n) = \top$. \square

Eventually we can state the complexity of the problem.

Corollary 13. *The MC problem for AB formulas over finite Kripke structures is $\mathbf{P}^{\mathbf{NP}}$ -hard (under LOGSPACE reductions).*

PROOF. The result follows from Corollary 12 considering that, for an instance of SNSAT \mathcal{I} , with $|\mathcal{I}| = n$, $\mathcal{K}_{\mathcal{I}}$ and $\psi_n \in \mathcal{F}_{\mathcal{I}}$ have a size polynomial in n and in the length of the formulas of \mathcal{I} . Moreover, their structures are repetitive, hence they can be built by using logarithmic working space. \square

We can prove the same complexity lower bound for the symmetric fragment $\overline{\text{AE}}$, just by transposing the edges of $\mathcal{K}_{\mathcal{I}}$, and by replacing [B] with [E] and $\langle A \rangle$ with $\langle \overline{A} \rangle$ in the definition of ψ_n . This hardness result immediately propagates to the bigger fragments $\overline{\text{AAB}}$ and $\overline{\text{AAE}}$.

Finally, we summarize the $\mathbf{P}^{\mathbf{NP}}$ -completeness results that can be obtained by putting together Corollary 5 in Section 3 and Corollary 13.

Corollary 14. *The MC problem for AB, $\overline{\text{AE}}$, $\overline{\text{AAB}}$, and $\overline{\text{AAE}}$ formulas over finite Kripke structures is $\mathbf{P}^{\mathbf{NP}}$ -complete.*

In the next section, we shall prove a different complexity lower bound for the fragments A, $\overline{\text{A}}$, $\overline{\text{AA}}$, $\overline{\text{AB}}$, and AE, to which the present one does not apply.

6. $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hardness of MC for the HS fragments A and $\overline{\text{A}}$

In this section, we prove that the MC problem for formulas of the fragment A (and of $\overline{\text{A}}$, respectively) over finite Kripke structures is $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hard, by reducing to it the problem PARITY(SAT) [55], a problem complete for $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$. The problem PARITY(SAT) is to decide, for a set of Boolean formulas Γ , if the number of *satisfiable* formulas in Γ is odd or even. The hardness of the MC problem for A and $\overline{\text{A}}$, immediately propagates to $\overline{\text{AA}}$, AE, $\overline{\text{AA}}$ and $\overline{\text{AB}}$.

Let Γ be a set of n Boolean formulas $\{\phi_i(x_1^i, \dots, x_{m_i}^i) \mid 1 \leq i \leq n, m_i \in \mathbb{N}\}$. We provide a Kripke structure $\mathcal{K}_{\text{PAR}}^{\Gamma}$ and an A-formula Φ_{Γ} such that $\mathcal{K}_{\text{PAR}}^{\Gamma} \models \Phi_{\Gamma}$ if and only if the number of satisfiable Boolean formulas in Γ is *odd*.

We start by defining a Boolean formula, $\text{parity}(F, Z)$, over two sets of Boolean variables, $F = \{f_1, \dots, f_n\}$ and $Z = \{z_1, \dots, z_t\}$, with $t = 3 \cdot (n - 1) + 1$. Such a formula allows one to decide the parity of the number of variables in F that evaluate to \top . Z is a set of auxiliary variables, whose truth values are *functionally determined* by those assigned to the variables in F . Given a truth assignment, the number of variables in F set to \top is even if $\text{parity}(F, Z)$ evaluates to \top , and, in particular, its last variable z_t evaluates to \top . The formula $\text{parity}(F, Z)$ is defined as follows:

$$\text{parity}(f_1, \dots, f_n, z_1, \dots, z_t) = z_t \wedge \text{par}_n(f_1, \dots, f_n, z_1, \dots, z_t),$$

where $t = 3 \cdot (n - 1) + 1$ and, for $i \geq 1$, $\text{par}_i(f_1, f_2, \dots, f_i, z_1, \dots, z_{3(i-1)+1})$ is inductively defined as:

$$\text{par}_1(f_1, z_1) = \neg f_1 \leftrightarrow z_1, \text{ and, for all } i \geq 2,$$

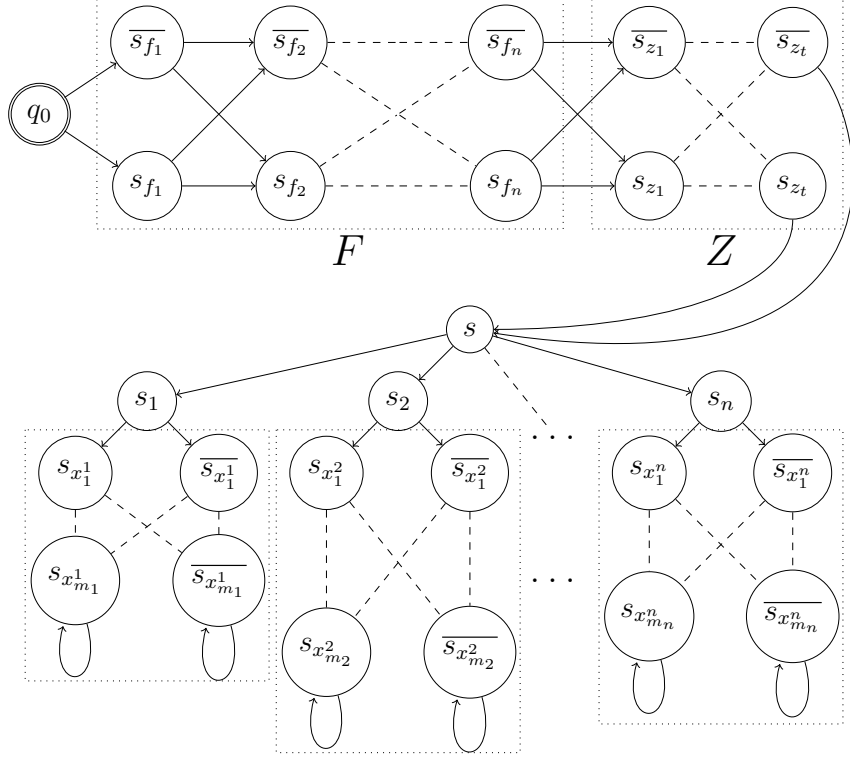


Figure 7: The Kripke structure \mathcal{K}_{PAR}^Γ .

$$\begin{aligned} \text{par}_i(f_1, f_2, \dots, f_i, z_1, \dots, z_{\alpha+3}) = \\ (z_{\alpha+1} \leftrightarrow (f_i \wedge \neg z_\alpha)) \wedge (z_{\alpha+2} \leftrightarrow (\neg f_i \wedge z_\alpha)) \wedge (z_{\alpha+3} \leftrightarrow (z_{\alpha+2} \vee z_{\alpha+1})) \wedge \\ \text{par}_{i-1}(f_1, f_2, \dots, f_{i-1}, z_1, \dots, z_\alpha), \end{aligned}$$

with $\alpha = 3 \cdot (i - 2) + 1$.

Each assignment satisfying par_i has to set z_α to the parity value for the set of Boolean variables f_1, f_2, \dots, f_{i-1} . Such a value is then possibly changed according to the truth of f_i and “assigned” to $z_{\alpha+3}$. Note that the length of $\text{parity}(F, Z)$ is polynomial in n .

We now show how to build the Kripke structure \mathcal{K}_{PAR}^Γ depicted in Figure 7, such that a subset of its traces encode all the possible truth assignments to the variables of $F \cup Z$ and to all the variables occurring in formulas of Γ . \mathcal{K}_{PAR}^Γ features a pair of states for each Boolean variable in $F \cup Z$ as well as for all the variables of formulas in Γ (one state for each truth value). Each path from the initial state q_0 to the state s represents a truth assignment to the variables in $F \cup Z$. Then, the structure branches into n substructures, each one modeling the possible truth assignments to the variables of a formula in Γ .

Formally, $\mathcal{K}_{PAR}^\Gamma = (\mathcal{AP}, W, \delta, \mu, q_0)$, where

- $\mathcal{AP} = \{p, q\} \cup F \cup Z \cup \{aux_i \mid 1 \leq i \leq n\} \cup \{x_{j_i}^i \mid 1 \leq i \leq n, 1 \leq j_i \leq m_i\}$,
- $W = \{q_0\} \cup \{s_{f_i}, \overline{s_{f_i}} \mid 1 \leq i \leq n\} \cup \{s_z, \overline{s_z} \mid z \in Z\} \cup \{s\} \cup \{s_i \mid 1 \leq i \leq n\} \cup \{s_{x_{j_i}^i}, \overline{s_{x_{j_i}^i}} \mid 1 \leq i \leq n, 1 \leq j_i \leq m_i\}$,

- $\delta = \{(q_0, s_{f_1}), (q_0, \overline{s_{f_1}})\} \cup \{(s_{f_i}, s_{f_{i+1}}), (s_{f_i}, \overline{s_{f_{i+1}}}), (\overline{s_{f_i}}, s_{f_{i+1}}), (\overline{s_{f_i}}, \overline{s_{f_{i+1}}}) \mid 1 \leq i < n\} \cup \{(s_{f_n}, s_{z_1}), (\overline{s_{f_n}}, s_{z_1}), (s_{f_n}, \overline{s_{z_1}}), (\overline{s_{f_n}}, \overline{s_{z_1}})\} \cup \{(s_{z_i}, s_{z_{i+1}}), (s_{z_i}, \overline{s_{z_{i+1}}}), (\overline{s_{z_i}}, s_{z_{i+1}}), (\overline{s_{z_i}}, \overline{s_{z_{i+1}}}) \mid 1 \leq i < t\} \cup \{(s_{z_t}, s), (\overline{s_{z_t}}, s)\} \cup \{(s, s_i), (s_i, s_{x_1^i}), (s_i, \overline{s_{x_1^i}}) \mid 1 \leq i \leq n\} \cup \{(s_{x_{j_i}^i}, s_{x_{j_i+1}^i}), (\overline{s_{x_{j_i}^i}}, s_{x_{j_i+1}^i}), (\overline{s_{x_{j_i}^i}}, s_{x_{j_i+1}^i}), (s_{x_{j_i}^i}, \overline{s_{x_{j_i+1}^i}}) \mid 1 \leq i \leq n, 1 \leq j_i < m_i\} \cup \{(s_{x_{m_i}^i}, s_{x_{m_i}^i}), (\overline{s_{x_{m_i}^i}}, \overline{s_{x_{m_i}^i}}) \mid 1 \leq i \leq n\},$
- and the labeling function μ is defined as follows:
 - $\mu(q_0) = \{p, q\} \cup F \cup Z$;
 - for all $1 \leq i \leq n$, $\mu(s_{f_i}) = \{p, q\} \cup F \cup Z$; $\mu(\overline{s_{f_i}}) = \{p, q\} \cup (F \setminus \{f_i\}) \cup Z$;
 - for all $z \in Z$, $\mu(s_z) = \{p, q\} \cup F \cup Z$; $\mu(\overline{s_z}) = \{p, q\} \cup F \cup (Z \setminus \{z\})$;
 - $\mu(s) = \{q\} \cup F \cup Z \cup \{aux_i \mid 1 \leq i \leq n\} \cup \{x_{j_i}^i \mid 1 \leq i \leq n, 1 \leq j_i \leq m_i\}$;
 - for all $1 \leq i \leq n$, $\mu(s_i) = \{aux_i\} \cup \{x_{j_i}^i \mid 1 \leq j_i \leq m_i\}$;
 - for all $1 \leq i \leq n$, $1 \leq k_i \leq m_i$, $\mu(s_{x_{k_i}^i}) = \{aux_i\} \cup \{x_{j_i}^i \mid 1 \leq j_i \leq m_i\}$, and $\mu(\overline{s_{x_{k_i}^i}}) = \{aux_i\} \cup \{x_{j_i}^i \mid 1 \leq j_i \leq m_i\} \setminus \{x_{k_i}^i\}$.

According to the definition of $\mathfrak{X}_{PAR}^\Gamma$, it holds that:

1. Each trace ρ from q_0 to s encodes a truth assignment to the proposition letters in $F \cup Z$ (for all $y \in F \cup Z$, y is \top in ρ if and only if $y \in \bigcap_{w \in \text{states}(\rho)} \mu(w)$). Conversely, for each truth assignment to the proposition letters in $F \cup Z$, there exists an initial trace ρ , reaching the state s , encoding such an assignment. Note that, among the initial traces, the ones leading to s are exactly those satisfying $q \wedge \neg p$.
2. An initial trace leading to s satisfies $\text{parity}(F, Z)$ if the induced assignment sets an even number of f_i 's to \top , and every $z \in Z$ to the truth value which is functionally implied by the values of the f_i 's.
3. A trace $\tilde{\rho}$ starting from s and ending in a state s , s_i , $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, with $1 \leq i \leq n$ and $1 \leq j \leq m_i$, encodes a truth assignment to the proposition letters x_1, \dots, x_{m_i} (if the trace ends in s or s_i , all the variables are assigned to \top ; if it ends in $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, in particular all the variables $x_{j+1}^i, \dots, x_{m_i}^i$ are assigned to \top , by homogeneity).
4. A Boolean formula $\phi_i(x_1^i, \dots, x_{m_i}^i) \in \Gamma$ is satisfiable if and only if there exists a trace $\tilde{\rho}$ starting from s and ending in a state s , s_i , $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, for some $j = 1, \dots, m_i$, such that $\mathfrak{X}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$.

Finally, let us consider the **A** formula

$$\psi = q \wedge \neg p \wedge \text{parity}(F, Z) \wedge \bigwedge_{i=1}^n (f_i \leftrightarrow \langle A \rangle (aux_i \wedge \phi_i(x_1^i, \dots, x_{m_i}^i))).$$

In view of the above observations, ψ is satisfied by an initial trace $\bar{\rho}$ if (and only if) (i) $\bar{\rho}$ leads to s , (ii) $\bar{\rho}$ induces an assignment which sets an even number of f_i 's to \top and all $z \in Z$

accordingly, and (iii) for all $1 \leq i \leq n$, f_i is \top if and only if there exists a trace $\tilde{\rho}$ starting from s and ending in a state $s, s_i, s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{K}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$. Note that the length of ψ is polynomial in the input size.

Let us now assume we are given an instance of PARITY(SAT) Γ with an *even* number of satisfiable Boolean formulas. Then, there exists an initial trace ρ ending in s such that, for all i , $s_{f_i} \in \text{states}(\rho)$ if $\phi_i(x_1^i, \dots, x_{m_i}^i)$ is satisfiable, and $\overline{s_{f_i}} \in \text{states}(\rho)$ otherwise. Moreover, ρ can be chosen in such a way that $\mathcal{K}_{PAR}^\Gamma, \rho \models \text{parity}(F, Z)$. It immediately follows that, for all i , $\mathcal{K}_{PAR}^\Gamma, \rho \models f_i$ if and only if $\mathcal{K}_{PAR}^\Gamma, \rho \models \langle A \rangle (\text{aux}_i \wedge \phi_i(x_1^i, \dots, x_{m_i}^i))$, concluding that $\mathcal{K}_{PAR}^\Gamma, \rho \models \psi$.

Conversely, let ρ be an initial trace such that $\mathcal{K}_{PAR}^\Gamma, \rho \models \psi$. It holds that ρ ends in s and sets an even number of f_i 's to \top . Furthermore, if $\mathcal{K}_{PAR}^\Gamma, \rho \models f_i$, then there exists $\tilde{\rho}$ starting from s and ending in $s, s_i, s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{K}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$, hence $\phi_i(x_1^i, \dots, x_{m_i}^i)$ is satisfiable. If $\mathcal{K}_{PAR}^\Gamma, \rho \models \neg f_i$, then there exists no $\tilde{\rho}$ starting from s and ending in $s, s_i, s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{K}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$. Thus $\phi_i(x_1^i, \dots, x_{m_i}^i)$ is unsatisfiable. Hence, Γ contains an even number of satisfiable formulas.

Therefore we have proved that the number of *satisfiable* Boolean formulas of Γ is *even* if and only if there exists an initial trace ρ such that $\mathcal{K}_{PAR}^\Gamma, \rho \models \psi$. This amounts to say that Γ contains an *odd* number of satisfiable Boolean formulas (the PARITY(SAT) problem) if and only if $\mathcal{K}_{PAR}^\Gamma \models \Phi_\Gamma$, where $\Phi_\Gamma = \neg\psi$ (the MC problem). The next theorem immediately follows.

Theorem 15. *The MC problem for A formulas over finite Kripke structures is $\mathbf{P}^{\text{NP}[O(\log n)]}$ -hard (under LOGSPACE reductions).*

A similar proof can be given for \overline{A} (roughly speaking, we replace all the occurrences of $\langle A \rangle$ in Φ_Γ by $\langle \overline{A} \rangle$, and we stick the n substructures of \mathcal{K}_{PAR}^Γ —after transposing all their edges—on q_0 , instead of s).

Finally, we observe that the $\mathbf{P}^{\text{NP}[O(\log n)]}$ -hardness of \overline{A} and A immediately propagates to $A\overline{A}$, $\overline{A}B$ and AE , yielding, together with Corollary 9 and Theorem 10, the following result.

Theorem 16. *The MC problem for A, \overline{A} , $A\overline{A}$, $\overline{A}B$ and AE formulas over finite Kripke structures is in $\mathbf{P}^{\text{NP}[O(\log^2 n)]}$ and it is hard for $\mathbf{P}^{\text{NP}[O(\log n)]}$.*

This result leaves open the question whether MC for the above fragments can be solved by $o(\log^2 n)$ (i.e., strictly less than $O(\log^2 n)$) queries to an **NP** oracle, or a tighter lower bound can be proved, or both (e.g., the problem may be complete for $\mathbf{P}^{\text{NP}[O(\log n \log \log n)]}$). As a matter of fact, any attempt to reduce $\text{TB(SAT)}_{1 \times M}$ to MC for A, AE , or $A\overline{A}$ failed, because in such “reduction” we need an HS formula of length $\Theta(n^{\log n})$, which clearly cannot be generated in polynomial time.

7. Conclusions and future work

In this paper, we have studied the complexity of MC for some fragments of the Halpern and Shoham’s interval temporal logic HS (under the homogeneity assumption), interpreted over finite Kripke structures.

First, we have proved that the fragments AB , \overline{AE} , $A\overline{AB}$, and $A\overline{AE}$ are complete for $\mathbf{P}^{\mathbf{NP}}$, thus joining other (point-based) temporal logics, e.g., CTL^+ , FCTL , and ECTL^+ , whose MC problem is complete for that class [28]. Among these fragments, AB turns out to be quite significant. In particular, as far as expressiveness is concerned, when interpreted over discrete *linear* orders, it captures LTL [9].

In addition, we have shown that MC for A , \overline{A} , $A\overline{A}$, \overline{AB} , and AE has a lower complexity placed in between $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ and $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$. This result has been proved by reducing MC to $\text{TB}(\text{SAT})_{1 \times M}$, the problem of deciding the output value of a complex circuit, where some gates are endowed with \mathbf{NP} oracles.

The fragments we have considered are somehow “halfway” between $A\overline{AB\overline{B}}$, $A\overline{AE\overline{E}}$, and $A\overline{AB\overline{E}}$, which are \mathbf{PSPACE} -complete [8, 37, 38], and Prop , B , and E , that are co-NP -complete [8, 37]. All MC procedures we propose can be easily implemented with the support of SAT-solvers, whose extreme efficiency may be “imported” in this context in a straightforward way.

Even though the homogeneity assumption is a natural choice in a number of application domains, we have recently explored alternative semantics. In particular, inspired by the work in [33], we have managed to relax it by defining interval labelling via regular expressions that allow one to specify the behaviour of proposition letters on each interval on the basis of its component states [7]. We have studied how complexity changes under such a semantic variant of HS, and it turns out that MC for full HS is still (nonelementarily) decidable, but all sub-fragments of $A\overline{AB\overline{B}}$ and $A\overline{AE\overline{E}}$ become complete for \mathbf{PSPACE} (even the complexity of the simple fragments Prop , B , and E is “pushed” to \mathbf{PSPACE} as a result of the introduction of regular expressions).

As for future work, we are looking for possible improvements to known complexity results for MC of full HS. We know that it is $\mathbf{EXSPACE}$ -hard (we proved the $\mathbf{EXSPACE}$ -hardness of its fragment BE [8]), while the only available decision procedure is nonelementary.

In addition, we want to study the MC problem for HS over *visibly pushdown systems* (VPS), which have the ability of modelling recursive programs and infinite state systems.

Last but not least, we are thinking of inherently *interval-based models of systems*. Kripke structures, being based on states, are naturally oriented to the description of point-based properties of systems, and of how they evolve state-by-state. We want to come up with suitable (and practical) description paradigms for systems, which allow us to directly model them on the basis of their interval behavior/properties. Only after devising these models (something that seems to be extremely challenging), a really general interval-based MC will be possible.

Acknowledgements

We would like to thank the reviewers for their useful comments and suggestions. The work by Alberto Molinari and Angelo Montanari has been supported by the GNCS project *Logics and Automata for Interval Model Checking*.

References

- [1] Allen, J. F., 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26 (11), 832–843.
- [2] Armando, A., Carbone, R., Compagna, L., 2007. LTL Model Checking for Security Protocols. In: *CSF*. pp. 385–396.
- [3] Basin, D., Cremers, C., Meadows, C., 2015. Model checking security protocols. URL <http://www-oldurl5.inf.ethz.ch/personal/basin/pubs/security-modelchecking.pdf>
- [4] Batzold, T., Morin, G., Pecoraro, J., 2009. Completeness in the Σ_2^P Hierarchy—A Compendium. URL [http://act.buaa.edu.cn/caoyang/files/notes/completeness in Delta_2_p hierarchy_A compendium.pdf](http://act.buaa.edu.cn/caoyang/files/notes/completeness%20in%20Delta_2_p_hierarchy_A_compendium.pdf)
- [5] Benerecetti, M., Guglielmo, R. D., Gentile, U., Marrone, S., Mazzocca, N., Nardone, R., Peron, A., Velardi, L., Vittorini, V., 2017. Dynamic state machines for modelling railway control systems. *Science of Computer Programming* 133, 116–153.
- [6] Bowman, H., Thompson, S. J., 2003. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation* 13 (2), 195–239.
- [7] Bozzelli, L., Molinari, A., Montanari, A., Peron, A., 2017. An in-Depth Investigation of Interval Temporal Logic Model Checking with Regular Expressions. In: *SEFM*. pp. 104–119.
- [8] Bozzelli, L., Molinari, A., Montanari, A., Peron, A., Sala, P., 2016. Interval Temporal Logic Model Checking: the Border Between Good and Bad HS Fragments. In: *IJCAR*. LNAI 9706. Springer, pp. 389–405.
- [9] Bozzelli, L., Molinari, A., Montanari, A., Peron, A., Sala, P., 2016. Interval vs. Point Temporal Logic Model Checking: an Expressiveness Comparison. In: *FSTTCS*. pp. 26:1–26:14.
- [10] Bozzelli, L., Molinari, A., Montanari, A., Peron, A., Sala, P., 2016. Model Checking the Logic of Allen’s Relations Meets and Started-by is $\mathbf{P}^{\mathbf{NP}}$ -Complete. In: *GandALF*. pp. 76–90.
- [11] Bozzelli, L., Molinari, A., Montanari, A., Peron, A., Sala, P., 2017. Satisfiability and model checking for the logic of sub-intervals under the homogeneity assumption. In: *ICALP*. LIPIcs 80. Schloss Dagstuhl, pp. 120:1–120:14.
- [12] Bresolin, D., Della Monica, D., Goranko, V., Montanari, A., Sciavicco, G., 2014. The dark side of interval temporal logic: marking the undecidability border. *Annals of Mathematics and Artificial Intelligence* 71 (1–3), 41–83.
- [13] Bresolin, D., Goranko, V., Montanari, A., Sala, P., 2010. Tableaux for logics of subinterval structures over dense orderings. *Journal of Logic and Computation* 20 (1), 133–166.
- [14] Bresolin, D., Goranko, V., Montanari, A., Sciavicco, G., 2009. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic* 161 (3), 289–304.
- [15] Bresolin, D., Montanari, A., Sala, P., Sciavicco, G., 2011. What’s decidable about Halpern and Shoham’s interval logic? The maximal fragment $\mathbf{ABB\bar{L}}$. In: *LICS*. pp. 387–396.
- [16] Buss, S., Hay, L., 1991. On truth-table reducibility to SAT. *Information and Computation* 102, 86–102.
- [17] Castro, J., Seara, C., 1992. Characterizations of some complexity classes between Θ_2^P and Δ_2^P . In: *STACS*. Springer, pp. 303–317.
- [18] Cimatti, A., 2001. *Industrial Applications of Model Checking*. Springer, Ch. 6, pp. 153–168.
- [19] Clarke, E. M., Grumberg, O., Peled, D. A., 2002. *Model Checking*. MIT Press.
- [20] Donini, F., Mongiello, M., Ruta, M., Totaro, R., 2006. A model checking-based method for verifying web application design. *Electronic Notes in Theoretical Computer Science* 151 (2), 19–32.
- [21] Emerson, E. A., Halpern, J. Y., 1986. “Sometimes” and “not never” revisited: on branching versus linear time temporal logic. *Journal of the ACM* 33 (1), 151–178.
- [22] Giordano, L., Terenziani, P., Bottrighi, A., Montani, S., Donzella, L., 2006. Model checking for clinical guidelines: an agent-based approach. In: *AMIA*. pp. 289–293.
- [23] Giunchiglia, F., Traverso, P., 1999. Planning as model checking. In: *ECP*. LNCS 1809. Springer, pp. 1–20.

- [24] Gligoric, M., Majumdar, R., 2013. Model Checking Database Applications. Springer, Ch. 1, pp. 549–564.
- [25] Goranko, V., Montanari, A., Sciavicco, G., 2004. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics* 14 (1–2), 9–54.
- [26] Gottlob, G., 1995. NP Trees and Carnap’s Modal Logic. *Journal of the ACM* 42 (2), 421–457.
- [27] Halpern, J. Y., Shoham, Y., 1991. A propositional modal logic of time intervals. *Journal of the ACM* 38 (4), 935–962.
- [28] Laroussinie, F., Markey, N., Schnoebelen, P., 2001. Model checking CTL⁺ and FCTL is hard. In: FOSSACS. pp. 318–331.
- [29] Laroussinie, F., Markey, N., Schnoebelen, P., 2002. On model checking durational Kripke structures. In: FOSSACS. Springer, pp. 264–279.
- [30] Laroussinie, F., Meyer, A., Petonnet, E., 2010. Counting CTL. Springer, Ch. 1, pp. 206–220.
- [31] Lomuscio, A., Michaliszyn, J., 2013. An epistemic Halpern-Shoham logic. In: IJCAI. pp. 1010–1016.
- [32] Lomuscio, A., Michaliszyn, J., 2014. Decidability of model checking multi-agent systems against a class of EHS specifications. In: ECAI. pp. 543–548.
- [33] Lomuscio, A., Michaliszyn, J., 2016. Model checking multi-agent systems against epistemic HS specifications with regular expressions. In: KR. pp. 298–308.
- [34] Lomuscio, A., Qu, H., Raimondi, F., 2009. MCMAS: A model checker for the verification of multi-agent systems. In: CAV. Springer, pp. 682–688.
- [35] Mentis, A., Katsaros, P., 2012. Model checking and code generation for transaction processing software. *Concurrency and Computation: Practice and Experience* 24 (7), 711–722.
- [36] Molinari, A., Montanari, A., Murano, A., Perelli, G., Peron, A., 2016. Checking interval properties of computations. *Acta Informatica* 53 (6–8), 587–619.
- [37] Molinari, A., Montanari, A., Peron, A., 2015. Complexity of ITL model checking: some well-behaved fragments of the interval logic HS. In: TIME. pp. 90–100.
- [38] Molinari, A., Montanari, A., Peron, A., 2015. A model checking procedure for interval temporal logics based on track representatives. In: CSL. pp. 193–210.
- [39] Molinari, A., Montanari, A., Peron, A., 2018. Model checking for fragments of Halpern and Shoham’s interval temporal logic based on track representatives. *Information and Computation*. In Press.
- [40] Molinari, A., Montanari, A., Peron, A., Sala, P., 2016. Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture. In: KR. pp. 473–483.
- [41] Montanari, A., Puppis, G., Sala, P., 2010. Maximal decidable fragments of Halpern and Shoham’s modal logic of intervals. In: ICALP. LNCS 6199. Springer, pp. 345–356.
- [42] Montanari, A., Puppis, G., Sala, P., 2015. A decidable weakening of compass logic based on cone-shaped cardinal directions. *Logical Methods in Computer Science* 11 (4).
- [43] Moszkowski, B., 1983. Reasoning about digital circuits. Ph.D. thesis, Stanford University, Stanford, CA.
- [44] Nardone, R., Gentile, U., Benerecetti, M., Peron, A., Vittorini, V., Marrone, S., Mazzocca, N., 2016. Modeling Railway Control Systems in Promela. Springer, Ch. 1, pp. 121–136.
- [45] Papadimitriou, C. H., Zachos, S. K., 1982. Two remarks on the power of counting. *Theoretical Computer Science: 6th GI-Conference*, 269–275.
- [46] Pnueli, A., 1977. The temporal logic of programs. In: FOCS. IEEE Computer Society, pp. 46–57.
- [47] Pnueli, A., 1981. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science* 13, 45–60.
- [48] Pratt-Hartmann, I., 2005. Temporal prepositions and their logic. *Artificial Intelligence* 166 (1–2), 1–36.
- [49] Roeper, P., 1980. Intervals and tenses. *Journal of Philosophical Logic* 9, 451–469.
- [50] Schnoebelen, P., 2003. Oracle circuits for branching-time model checking. In: ICALP. LNCS 2719. Springer, pp. 790–801.
- [51] Sistla, A. P., Clarke, E. M., 1985. The complexity of propositional linear temporal logics. *Journal of the ACM* 32 (3), 733–749.
- [52] Stockmeyer, L. J., 1976. The polynomial-time hierarchy. *Theoretical Computer Science* 3 (1), 1–22.

- [53] Venema, Y., 1990. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic* 31 (4), 529–547.
- [54] Venema, Y., 1991. A modal logic for chopping intervals. *Journal of Logic and Computation* 1 (4), 453–476.
- [55] Wagner, K. W., 1987. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science* 51 (1), 53–80.
- [56] Wagner, K. W., 1990. Bounded query classes. *SIAM Journal of Computation* 19 (5), 833–846.
- [57] Witkowski, T., Blanc, N., Kroening, D., Weissenbacher, G., 2007. Model checking concurrent Linux device drivers. In: ASE. pp. 501–504.
- [58] Zhou, C., Hansen, M. R., 2004. Duration Calculus—A Formal Approach to Real-Time Systems. *Monographs in Theoretical Computer Science*. Springer.

Appendix A. Appendix

Appendix A.1. Proof of Lemma 2

Lemma 2. Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, ψ be an $\mathbf{A}\bar{\mathbf{A}}\mathbf{B}$ formula, and $V_{\mathbf{A}}(\cdot, \cdot)$ and $V_{\bar{\mathbf{A}}}(\cdot, \cdot)$ be two Boolean arrays. Let us assume that

1. for each $\langle \mathbf{A} \rangle \phi \in \text{ModSubf}_{\mathbf{A}\bar{\mathbf{A}}}(\psi)$ and $v' \in W$, $V_{\mathbf{A}}(\phi, v') = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v'$ and $\mathcal{K}, \rho \models \phi$, and
2. for each $\langle \bar{\mathbf{A}} \rangle \phi \in \text{ModSubf}_{\mathbf{A}\bar{\mathbf{A}}}(\psi)$ and $v' \in W$, $V_{\bar{\mathbf{A}}}(\phi, v') = \top$ if and only if there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho) = v'$ and $\mathcal{K}, \rho \models \phi$.

Then, $\text{Oracle}(\mathcal{K}, \psi, v, \text{DIRECTION}, V_{\mathbf{A}} \cup V_{\bar{\mathbf{A}}})$ features a successful computation (returning \top) if and only if:

- there is $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$, when DIRECTION is FORWARD;
- there is $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$, when DIRECTION is BACKWARD.

PROOF. It is easy to check that if $\tilde{\rho}$ is the trace non-deterministically generated by $\mathbf{A_trace}$ at line 1, then, for $i = 1, \dots, |\tilde{\rho}|$, it holds that $\mathcal{K}, \tilde{\rho}(1, i) \models \phi \iff T[\phi, i] = \top$, either by hypothesis, when ϕ occurs in $\text{ModSubf}_{\mathbf{A}\bar{\mathbf{A}}}(\psi)$ (lines 2–7), or by construction, when ϕ does not occur in $\text{ModSubf}_{\mathbf{A}\bar{\mathbf{A}}}(\psi)$ (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 $\text{Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_{\mathbf{A}} \cup V_{\bar{\mathbf{A}}})$ features a successful computation, it means that there exists a trace $\tilde{\rho} \in \text{Trc}_{\mathcal{K}}$ (generated at line 1) such that $\text{fst}(\tilde{\rho}) = v$ and $T[\psi, |\tilde{\rho}|] = \top$ implying that $\mathcal{K}, \tilde{\rho} \models \psi$.

(\Leftarrow) If there exists $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = v$ and $\mathcal{K}, \rho \models \psi$, by Proposition 1 there exists $\tilde{\rho} \in \text{Trc}_{\mathcal{K}}$ such that $\mathcal{K}, \tilde{\rho} \models \psi$, $\text{fst}(\tilde{\rho}) = \text{fst}(\rho)$, and $|\tilde{\rho}| \leq |W| \cdot (2|\psi| + 1)^2$. It follows that in some non-deterministic instance of $\text{Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_{\mathbf{A}} \cup V_{\bar{\mathbf{A}}})$, $\mathbf{A_trace}(\mathcal{K}, v, |W| \cdot (2|\psi| + 1)^2, \text{FORWARD})$ returns such $\tilde{\rho}$ (at line 1). Finally, we have that $T[\psi, |\tilde{\rho}|] = \top$ as $\mathcal{K}, \tilde{\rho} \models \psi$, and hence the considered instance of $\text{Oracle}(\mathcal{K}, \psi, v, \text{FORWARD}, V_{\mathbf{A}} \cup V_{\bar{\mathbf{A}}})$ is successful. \square

Appendix A.2. Proof of Theorem 7

Theorem 7. Let ψ be an AA formula and $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$ be a finite Kripke structure. For every block B of $T_{\mathcal{K}, \neg\psi}$, if B is associated with an AA formula φ , then

- if B is a FORWARD block, for all $i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ if and only if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$;
- if B is a BACKWARD block, for all $i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ if and only if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{lst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$.

PROOF. The proof is by induction on the level $L \geq 1$ of the block B . The proof of the base case for $L = 1$ is a simpler version of the inductive step and it is therefore omitted.

Assume that B is a FORWARD block at level $L \geq 2$ associated with a formula φ (the BACKWARD case is symmetric).

We first prove the implication (\Leftarrow). We have to show that if there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = w_i$ (for some $i \in \{1, \dots, |W|\}$) and $\mathcal{K}, \rho \models \varphi$, then $B(z_i) = \top$ that is, there exists a truth assignment ω to the variables in V satisfying the formula $F_i(Y, V)$ of G_i . In [38], it is proved that if φ is an AA formula and $\mathcal{K}, \rho \models \varphi$ (as in this case), there exists a trace $\rho' \in \text{Trc}_{\mathcal{K}}$, with $|\rho'| \leq |W|^2 + 2$, such that $\text{fst}(\rho) = \text{fst}(\rho') = w_i$, $\text{lst}(\rho) = \text{lst}(\rho')$, and $\mathcal{K}, \rho' \models \varphi$. Thus, by Proposition 6, there exists a truth assignment ω to the variables in V , that satisfies $\text{trace}(V_{\text{trace}}, V_{\text{last}}, V_{\mathcal{AP}})$, such that for all $1 \leq r \leq |\rho'|$ and $1 \leq j \leq |W|$, $\rho(r) = w_j \iff \omega(v_j^r) = \top$ and $\omega(v_j^{|\rho|}) = \omega(v_j)$, and for all $p \in \mathcal{AP}$, $\omega(v_p) = \top \iff \mathcal{K}, \rho' \models p$ (\star).

Since $L \geq 2$, it holds that $\text{ModSubf}_{\text{AA}}(\varphi) \neq \emptyset$. Let us consider a FORWARD child B' of B (if any), at a level lower than L , associated with some formula ξ such that $\langle A \rangle \xi \in \text{ModSubf}_{\text{AA}}(\varphi)$. By the inductive hypothesis, for all j , $B'(z_j) = \top$ if and only if there exists a trace $\bar{\rho} \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\bar{\rho}) = w_j$ and $\mathcal{K}, \bar{\rho} \models \xi$. Thus, $\mathcal{K}, \rho' \models \langle A \rangle \xi$ if and only if there exists $\tilde{\rho} \in \text{Trc}_{\mathcal{K}}$, with $\text{fst}(\tilde{\rho}) = \text{lst}(\rho') = w_j$, for some j , and $\mathcal{K}, \tilde{\rho} \models \xi$ if and only if $B'(z_j) (= y_j^\xi) = \top$. So if $\mathcal{K}, \rho' \models \langle A \rangle \xi$, then $y_j^\xi = \top$, and $\omega(v_j) \wedge y_j^\xi = \top$. Now, to satisfy $F_i(Y, V)$, the truth assignment ω has to be such that $\omega(v_{\langle A \rangle \xi}) = \top$. If $\mathcal{K}, \rho' \not\models \langle A \rangle \xi$, then $y_j^\xi = \perp$, thus $\bigvee_{s=1}^{|W|} (\omega(v_s) \wedge y_s^\xi)$ is false, and ω must be such that $\omega(v_{\langle A \rangle \xi}) = \perp$. To conclude, $\mathcal{K}, \rho' \models \langle A \rangle \xi$ if and only if $\omega(v_{\langle A \rangle \xi}) = \top$ ($\star\star$). The symmetric reasoning can be applied to BACKWARD children of B . Since $\mathcal{K}, \rho' \models \varphi$, by (\star) and ($\star\star$), we have $\omega(\bar{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$.

We prove now the implication (\Rightarrow). If $B(z_i) = \top$, then there exists a truth assignment ω of V satisfying $F_i(Y, V)$. In particular, ω satisfies $\text{trace}(V_{\text{trace}}, V_{\text{last}}, V_{\mathcal{AP}})$ and v_i^1 , thus, by Proposition 6, there exists a trace $\rho \in \text{Trc}_{\mathcal{K}}$ such that $\text{fst}(\rho) = w_i$, $\text{lst}(\rho) = w_j$, for some j , and $\mathcal{K}, \rho \models p \iff \omega(v_p) = \top$, for any $p \in \mathcal{AP}$. By inductive hypothesis, for all the formulas $\langle A \rangle \xi \in \text{ModSubf}_{\text{AA}}(\varphi)$, $\mathcal{K}, \rho \models \langle A \rangle \xi$ if and only if $\omega(v_{\langle A \rangle \xi}) = \top$, and symmetrically, for all $\langle \bar{A} \rangle \xi' \in \text{ModSubf}_{\text{AA}}(\varphi)$, $\mathcal{K}, \rho \models \langle \bar{A} \rangle \xi'$ if and only if $\omega(v_{\langle \bar{A} \rangle \xi'}) = \top$. Since $\omega(\bar{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$, then $\mathcal{K}, \rho \models \varphi$. \square

Appendix A.3. Proof of Theorem 11

Theorem 11. Let \mathcal{I} be an instance of SNSAT with $|\mathcal{I}| = n$, and let $\mathcal{K}_{\mathcal{I}}$ and $\mathcal{F}_{\mathcal{I}}$ 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 $v_{\mathcal{I}}(x_r) = \top \iff \mathcal{K}_{\mathcal{I}}, w_{x_r} \models \psi_k$;
2. if $k \geq r + 1$, then $v_{\mathcal{I}}(x_r) = \perp \iff \mathcal{K}_{\mathcal{I}}, \overline{w_{x_r}} \models \psi_k$.

PROOF. The proof is by induction on $k \geq 0$. If $k = 0$, the thesis trivially holds. Therefore, let us assume that $k \geq 1$. We first prove the (\Leftarrow) implication for both item 1 and item 2.

- (Item 1) Assume that $k \geq r$ and $\mathcal{K}_{\mathcal{I}}, w_{x_r} \models \psi_k$. Thus, there exists $\rho \in \text{Trc}_{\mathcal{K}_{\mathcal{I}}}$ such that $\rho = w_{x_r} \cdots s_0$ does not pass through any $\overline{s_m}$, for $1 \leq m \leq r$, and $\mathcal{K}_{\mathcal{I}}, \rho \models \varphi_k$. We show by induction on $1 \leq m \leq r$ that $\omega_{\rho}(x_m) = v_{\mathcal{I}}(x_m)$.
 - Let us consider first the case where ρ passes through w_{x_m} , implying that $\omega_{\rho}(x_m) = \top$; thus $\mathcal{K}_{\mathcal{I}}, \rho \models x_m \wedge \neg r_m$ and $\mathcal{K}_{\mathcal{I}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$. If $m = 1$ (base case), since F_1 is satisfiable, then $v_{\mathcal{I}}(x_1) = \top$. If $m \geq 2$ (inductive case), by the inductive hypothesis $\omega_{\rho}(x_1) = v_{\mathcal{I}}(x_1), \dots, \omega_{\rho}(x_{m-1}) = v_{\mathcal{I}}(x_{m-1})$. Since $\mathcal{K}_{\mathcal{I}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$ or, equivalently, $F_m(\omega_{\rho}(x_1), \dots, \omega_{\rho}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$, it holds that $F_m(v_{\mathcal{I}}(x_1), \dots, v_{\mathcal{I}}(x_{m-1}), \omega_{\rho}(Z_m)) = \top$ and, by definition of $v_{\mathcal{I}}$, $v_{\mathcal{I}}(x_m) = \top$.
 - 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, $\mathcal{K}_{\mathcal{I}}, \overline{w_{x_m}} \cdot \overline{s_m} \models \langle A \rangle (\ell_{=2} \wedge \neg \psi_{k-1})$ and $\mathcal{K}_{\mathcal{I}}, \overline{s_m} \cdot w_{x_m} \not\models \psi_{k-1}$, with $\psi_{k-1} = \langle A \rangle \varphi_{k-1}$. Hence $\mathcal{K}_{\mathcal{I}}, w_{x_m} \not\models \psi_{k-1}$. 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 $v_{\mathcal{I}}(x_m) = \perp$.

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

- (Item 2) Assume that $k \geq r + 1$ and $\mathcal{K}_{\mathcal{I}}, \overline{w_{x_r}} \models \psi_k$. The proof follows the same steps as the previous case and it is thus only sketched: there exists $\rho \in \text{Trc}_{\mathcal{K}_{\mathcal{I}}}$ 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 $\mathcal{K}_{\mathcal{I}}, \rho \models \varphi_k$. The only difference 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 $\mathcal{K}_{\mathcal{I}}, w_{x_r} \not\models \psi_{k-1}$. Now, $k' = k - 1 \geq r \geq 1$ and, by the inductive hypothesis (on $k' = k - 1$), $v_{\mathcal{I}}(x_r) = \perp$.

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

- (Item 1) Assume that $k \geq r$ and $v_{\mathcal{I}}(x_r) = \top$. Let us consider the trace $\rho \in \text{Trc}_{\mathcal{K}_{\mathcal{I}}}$, $\rho = w_{x_r} \cdots s_0$ never passing through any $\overline{s_m}$, for $1 \leq m \leq r$, such that $w_{x_m} \in \text{states}(\rho)$ if $v_{\mathcal{I}}(x_m) = \top$, and $\overline{w_{x_m}} \in \text{states}(\rho)$ if $v_{\mathcal{I}}(x_m) = \perp$, for $1 \leq m \leq r$. Such a choice of ρ ensures that $v_{\mathcal{I}}(x_m) = \omega_{\rho}(x_m)$. In addition, the choice of ρ has to induce also

the proper truth-assignment of private variables, that is, if $v_{\mathcal{I}}(x_m) = \top$, then for $1 \leq u_m \leq j_m$, $w_{z_m^{u_m}} \in \text{states}(\rho)$ if $F_m(v_{\mathcal{I}}(x_1), \dots, v_{\mathcal{I}}(x_{m-1}), Z_m)$ is satisfied for $z_m^{u_m} = \top$, and $\overline{w_{z_m^{u_m}}} \in \text{states}(\rho)$ otherwise. Note that such a choice of ρ is always possible. We have to show that $\mathcal{K}_{\mathcal{I}}, \rho \models \varphi_k$, hence $\mathcal{K}_{\mathcal{I}}, w_{x_r} \models \psi_k$.

- For all $1 \leq m \leq r$ such that $v_{\mathcal{I}}(x_m) = \top$, it holds that $F_m(v_{\mathcal{I}}(x_1), \dots, v_{\mathcal{I}}(x_{m-1}), Z_m)$ 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, $\mathcal{K}_{\mathcal{I}}, \rho \models F_m(x_1, \dots, x_{m-1}, Z_m)$. Thus, $\mathcal{K}_{\mathcal{I}}, \rho \models \bigwedge_{i=1}^n \left((x_i \wedge \neg r_i) \rightarrow F_i(x_1, \dots, x_{i-1}, Z_i) \right)$.
- Conversely, for all $1 \leq m < r$ such that $v_{\mathcal{I}}(x_m) = \perp$ ($m \neq r$ as, by hypothesis, $v_{\mathcal{I}}(x_r) = \top$), it holds that $\overline{w_{x_m}} \in \text{states}(\rho)$. Since $m < r$, we have $k \geq r > m$ and $k - 1 \geq m \geq 1$. By the inductive hypothesis, $\mathcal{K}_{\mathcal{I}}, w_{x_m} \not\models \psi_{k-1}$. It follows that $\mathcal{K}_{\mathcal{I}}, \overline{s_m} \cdot w_{x_m} \models \neg \psi_{k-1} \wedge \ell_{=2}$, $\mathcal{K}_{\mathcal{I}}, \overline{w_{x_m}} \cdot \overline{s_m} \models \neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2})$ and $\mathcal{K}_{\mathcal{I}}, \overline{w_{x_m}} \models \langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2}))$. Hence, $\mathcal{K}_{\mathcal{I}}, \rho \models [\text{B}] \left(\left(\bigvee_{i=1}^n \langle A \rangle p_{\overline{x_i}} \right) \rightarrow \langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2})) \right)$.

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

- (Item 2) Assume that $k \geq r + 1$ and $v_{\mathcal{I}}(x_r) = \perp$. The proof is as before and it is only sketched. In this case, we choose a trace $\rho = \overline{w_{x_r}} \cdot \dots \cdot s_0$. Since $k' = k - 1 \geq r$, by the inductive hypothesis, $\mathcal{K}_{\mathcal{I}}, w_{x_r} \not\models \psi_{k-1}$, and we can prove that $\mathcal{K}_{\mathcal{I}}, \overline{w_{x_r}} \models \langle A \rangle (\neg s \wedge \ell_{=2} \wedge \langle A \rangle (\neg \psi_{k-1} \wedge \ell_{=2}))$. \square