

Model Checking Well-Behaved Fragments of HS: The (Almost) Final Picture

Alberto Molinari and **Angelo Montanari**

Department of Mathematics, Computer Science, and Physics
University of Udine, Italy
molinari.alberto@gmail.com, angelo.montanari@uniud.it

Adriano Peron

Department of Electronic Engineering and IT
University of Napoli “Federico II”, Italy
adrperon@unina.it

Pietro Sala

Department of Computer Science
University of Verona, Italy
pietro.sala@univr.it

Abstract

Model checking is one of the most powerful and widespread tools for system verification with applications in many areas of computer science and artificial intelligence. The large majority of model checkers deal with properties expressed in point-based temporal logics, such as LTL and CTL. However, there exist relevant properties of systems which are inherently interval-based. Model checking algorithms for interval temporal logics (ITLs) have recently been proposed to check interval properties of computations. As the model checking problem for full Halpern and Shoham’s ITL (HS for short) turns out to be decidable, but computationally heavy, research has focused on its well-behaved fragments. In this paper, we provide an almost final picture of the computational complexity of model checking for HS fragments with modalities for (a subset of) Allen’s relations *meets*, *met by*, *starts*, and *ends*.

Introduction

Model checking is a significant and extensively studied problem in computer science and artificial intelligence. It has proved itself to be extremely useful in formal verification (Clarke, Grumberg, and Peled 2002), but it has also been successfully exploited in various areas of AI, ranging from planning and plan validation to configuration and multi-agent systems, e.g., (Giunchiglia and Traverso 1999; Lomuscio and Raimondi 2006).

Model checking allows one to specify the desired properties of a system in a temporal logic and to automatically check them against a suitable model of the system. Point-based temporal logics, such as LTL, CTL, and the like, which allow one to predicate over computation states/worlds, are usually adopted as the specification language, while system models are generally labelled state-transition graphs (Kripke structures). In a large number of application domains, point-based temporal logics turn out to be suitable for practical purposes. However, there are some relevant temporal properties that involve actions with duration, accomplishments, and temporal aggregations, which are inherently “interval-based” and thus cannot be expressed by point-based logics. In this paper, we focus on model checking algorithms for interval temporal logic (ITL).

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ITLs take intervals, instead of points, as their primitive entities, providing an alternative setting for reasoning about time (Halpern and Shoham 1991; Moszkowski 1983; Venema 1990; 1991). They have been applied in various areas of computer science and AI, including formal verification, computational linguistics, planning, and multi-agent systems (Lomuscio and Michaliszyn 2013; Moszkowski 1983; Pratt-Hartmann 2005; Zhou and Hansen 2004). To check interval properties of computations, one needs to collect information about states into computation stretches. This amounts to interpret each finite path of a Kripke structure as an interval, and to suitably define its labelling on the basis of the labelling of the states that compose it.

The most famous among ITLs is Halpern and Shoham’s modal logic of time intervals, abbreviated HS (Halpern and Shoham 1991). HS features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen’s relations (Allen 1983)), apart from equality. The satisfiability problem for HS turns out to be undecidable over all relevant (classes of) linear orders (Halpern and Shoham 1991). The same holds for most fragments of it (Bresolin et al. 2014; Lodaya 2000; Marcinkowski and Michaliszyn 2014). However, some meaningful exceptions exist, including the logic of temporal neighbourhood \overline{AA} and the logic of sub-intervals D (Bresolin et al. 2010; 2009; 2011; Montanari, Puppis, and Sala 2010).

In this paper, we address some open issues in HS model checking, which only recently entered the research agenda. In (Montanari et al. 2014), Montanari et al. deal with the model checking problem for full HS over finite Kripke structures (under the homogeneity assumption (Roeper 1980)). They introduce the fundamental ingredients of the problem, namely, the interpretation of HS formulas over (abstract) interval models, the mapping of finite Kripke structures into (abstract) interval models, and the notion of track descriptor, and they prove a small model theorem showing its non-elementary decidability. In (Molinari et al. 2015), Molinari et al. provide a lower bound to the complexity of the problem, showing that it is **EXPSpace**-hard, if a succinct encoding of formulas is used, **PSPACE**-hard otherwise. In (Molinari, Montanari, and Peron 2015b), the authors show that model checking for the HS fragment $\overline{AABB\overline{E}}$ (resp., $\overline{AA\overline{E}\overline{E}\overline{B}}$), whose modalities allow one to access intervals

which are met by/meet the current one, or are prefixes (resp., suffixes) or right/left-extensions of it, is in **EXSPACE**. Moreover, they prove that the problem is **NEXPTIME**-hard, if a succinct encoding of formulas is used, **NP**-hard otherwise. Finally, they showed that formulas which satisfy a (constant) bound on the nesting depth of $\langle B \rangle$ (resp., $\langle E \rangle$) modalities can be checked in polynomial working space. In (Molinari, Montanari, and Peron 2015a), the authors identify some well-behaved HS fragments, which are expressive enough to capture meaningful interval properties and computationally much better: the universal fragment of $\overline{A\overline{A}BE}$ and the fragments $\overline{A\overline{A}BE}$ and $\overline{A\overline{B}}$, whose model checking is complete for, respectively, **co-NP** and **PSPACE**.

In (Lomuscio and Michaliszyn 2013; 2014; 2015), Lomuscio and Michaliszyn investigate the model checking problem for epistemic extensions of some HS fragments. A detailed account of their results can be found in (Molinari et al. 2015). However, their semantic assumptions considerably differ from those made in (Montanari et al. 2014), thus making it difficult to compare the two research lines.

In the present work, we almost completely sort out the complexity of the sub-fragments of $\overline{A\overline{A}BE}$. We already know that $\overline{A\overline{B}}$ is **PSPACE**-complete. We first show that the model checking problem for A (and for \overline{A}) is $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hard. Then, we prove that the problem for $\overline{A\overline{A}}$ is in $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$. We conclude the paper by showing that it is already **PSPACE**-hard for \overline{B} (and for \overline{E}). It is worth pointing out that in all cases the complexity of the model checking problem turns out to be comparable to or lower than that of LTL (which is known to be **PSPACE**-complete).

The rest of the paper is organized as follows. First, we give some background knowledge about HS, Kripke structures and abstract interval models. Then, we provide an overview of the main results and relate them to known ones. In the next three sections, a hardness result for A (and \overline{A}), a model checking algorithm for $\overline{A\overline{A}}$, and a hardness result for \overline{B} (and \overline{E}) are presented. Conclusions provide a short assessment of the work and outline future research directions.

Preliminaries

The interval temporal logic HS

An interval algebra to reason about intervals and their relative order was first proposed by Allen in (Allen 1983). A systematic logical study of interval representation and reasoning was then presented by Halpern and Shoham, who introduced the interval temporal logic HS featuring one modality for each Allen’s relation, with the exception of equality (Halpern and Shoham 1991). Table 1 depicts 6 of the 13 Allen’s relations, together with the corresponding HS (existential) modalities. The other 7 relations are the 6 inverse relations (given a binary relation \mathcal{R} , the inverse relation $\overline{\mathcal{R}}$ is such that $b\overline{\mathcal{R}}a$ if and only if $a\mathcal{R}b$) and equality.

The HS language consists of a set of proposition letters \mathcal{AP} , the Boolean connectives \neg and \wedge , and a temporal modality for each of the (non trivial) Allen’s relations, i.e., $\langle A \rangle$, $\langle L \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle D \rangle$, $\langle O \rangle$, $\langle \overline{A} \rangle$, $\langle \overline{L} \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$, $\langle \overline{D} \rangle$, and

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

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

$\langle \overline{O} \rangle$. HS formulas are defined by the following 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 will also exploit the standard logical connectives (disjunction \vee , implication \rightarrow , and double implication \leftrightarrow) as abbreviations. Furthermore, for any modality X , the dual universal modalities $[X]\psi$ and $[\overline{X}]\psi$ are defined as $\neg\langle X \rangle\neg\psi$ and $\neg\langle \overline{X} \rangle\neg\psi$, respectively. Given any subset of Allen’s relations $\{X_1, \dots, X_n\}$, we denote by $X_1 \dots X_n$ the HS fragment that features modalities $\langle X_1 \rangle, \dots, \langle X_n \rangle$ only.

W.l.o.g., we assume the strict semantics of HS: only intervals consisting of at least two points are considered, thus excluding point-intervals¹. Under this assumption, all HS modalities can be expressed in terms of modalities $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ (Venema 1990). HS can be viewed as a multi-modal logic with these 6 primitive modalities and its semantics can be defined over a multi-modal Kripke structure, called *abstract interval model*, where (strict) intervals are treated as atomic objects and Allen’s relations as binary relations between pairs of intervals.

Definition 1. (Molinari et al. 2015) *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 finite set of proposition letters;
- \mathbb{I} is a possibly infinite set of atomic objects (worlds);
- $A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}$ are three binary relations over \mathbb{I} ;
- $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is a (total) labeling function, which assigns a set of proposition letters to each world.

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

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

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

¹Strict semantics can easily be “relaxed” to include point-intervals. All results we prove in the paper hold for the non-strict semantics as well.



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

Kripke structures and abstract interval models

Finite state systems are usually modelled as finite Kripke structures. In (Montanari et al. 2014), the authors define a mapping from Kripke structures to abstract interval models, that allows one to specify interval properties of computations by means of HS formulas.

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

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

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

Definition 3. A track ρ over a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is a finite sequence of states $v_0 \cdots v_n$, with $n \geq 1$, such that for all $i \in \{0, \dots, n-1\}$, $(v_i, v_{i+1}) \in \delta$.

Let $\text{Trk}_{\mathcal{K}}$ be the (infinite) set of all tracks over a finite Kripke structure \mathcal{K} . For any track $\rho = v_0 \cdots v_n \in \text{Trk}_{\mathcal{K}}$, we define: $|\rho| = n + 1$, $\rho(i) = v_i$, $\text{states}(\rho) = \{v_0, \dots, v_n\} \subseteq W$, $\text{intstates}(\rho) = \{v_1, \dots, v_{n-1}\} \subseteq W$, $\text{fst}(\rho) = v_0$, and $\text{lst}(\rho) = v_n$. If $\text{fst}(\rho) = w_0$, ρ is called an *initial track*. With $\rho(i, j) = v_i \cdots v_j$, for $0 \leq i < j \leq |\rho| - 1$, we denote the subtrack of ρ bounded by positions i and j . By $\text{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ and $\text{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$ we denote the sets of all proper prefixes and suffixes of ρ , respectively. Notice that the length of tracks, prefixes, and suffixes is greater than 1, as they will be mapped into strict intervals. Finally, by $\rho \cdot \rho'$ we denote the concatenation of the tracks ρ and ρ' .

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

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

Relations $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as the Allen's relations A , B , and E , respectively. Moreover, according to the definition of σ , $p \in \mathcal{AP}$ holds over $\rho = v_0 \cdots v_n$ iff it holds

over all the states v_0, \dots, v_n of ρ . This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval iff it holds over all its subintervals.

Satisfiability of an HS formula over a Kripke structure can be given in terms of induced abstract interval models.

Definition 5. Let \mathcal{K} be a finite Kripke structure, $\rho \in \text{Trk}_{\mathcal{K}}$, and ψ be an HS formula. A track ρ of \mathcal{K} satisfies ψ , denoted as $\mathcal{K}, \rho \models \psi$, iff it holds that $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

Definition 6. Let \mathcal{K} be a finite Kripke structure and ψ be an HS formula. We say that \mathcal{K} models ψ , denoted as $\mathcal{K} \models \psi$, iff for all initial tracks $\rho \in \text{Trk}_{\mathcal{K}}$ it holds that $\mathcal{K}, \rho \models \psi$.

The *model checking problem* for HS over finite Kripke structures is the problem of deciding whether $\mathcal{K} \models \psi$.

Some examples of meaningful properties of tracks expressible in HS can be found in (Molinari et al. 2015).

The general picture

In (Montanari et al. 2014; Molinari et al. 2015), the authors show that, given a finite Kripke structure \mathcal{K} and a bound k on the structural complexity of HS formulas, that is, on the nesting depth of E and B 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 prove that the model checking problem for (full) HS is decidable, providing an algorithm with non-elementary complexity. Moreover, they show that the problem for the fragment A \bar{A} B \bar{E} , and thus for full HS, is **PSPACE**-hard (**EXPSPACE**-hard if a suitable succinct encoding of formulas is exploited).

In (Molinari, Montanari, and Peron 2015b), the authors study the fragment A \bar{A} B \bar{B} \bar{E} (and the symmetrical fragment A \bar{A} \bar{E} B \bar{B}), devising an **EXPSPACE** model checking algorithm which exploits the possibility of finding, for each track of the Kripke structure, a satisfiability-preserving track of bounded length, called *track representative*. In this way, the algorithm needs to check only tracks having a bounded maximum length. Later (Molinari, Montanari, and Peron 2015a) proved that the problem for A \bar{A} B \bar{B} \bar{E} (and A \bar{A} \bar{E} B \bar{B}) is **PSPACE**-hard (if a suitable succinct encoding of formulas is exploited, the algorithm remains in **EXPSPACE**, but a **NEXPTIME** lower bound can be given (Molinari, Montanari, and Peron 2015b)).

In (Molinari, Montanari, and Peron 2015a) the authors identify some well-behaved HS fragments, namely, \forall A \bar{A} B \bar{E} (and \exists A \bar{A} B \bar{E}) and A \bar{A} B \bar{E} , which are still expressive enough to capture meaningful interval properties of state-transition systems and whose model checking problem exhibits a computational complexity considerably lower than that of full HS. In particular, they prove that the problem is **PSPACE**-complete for the fragment A \bar{A} B \bar{E} (and its sub-fragment A \bar{B}) and **co-NP**-complete for the fragment \forall A \bar{A} B \bar{E} .

In this paper, we (almost) complete the picture for the sub-fragments of A \bar{A} B \bar{E} . We first prove that the model checking problem for A (and for \bar{A}) is **P^{NP}^[O(log n)]**-hard by a reduction from the **PARITY(SAT)** problem. Then, we devise a model checking algorithm for A \bar{A} by reducing its model

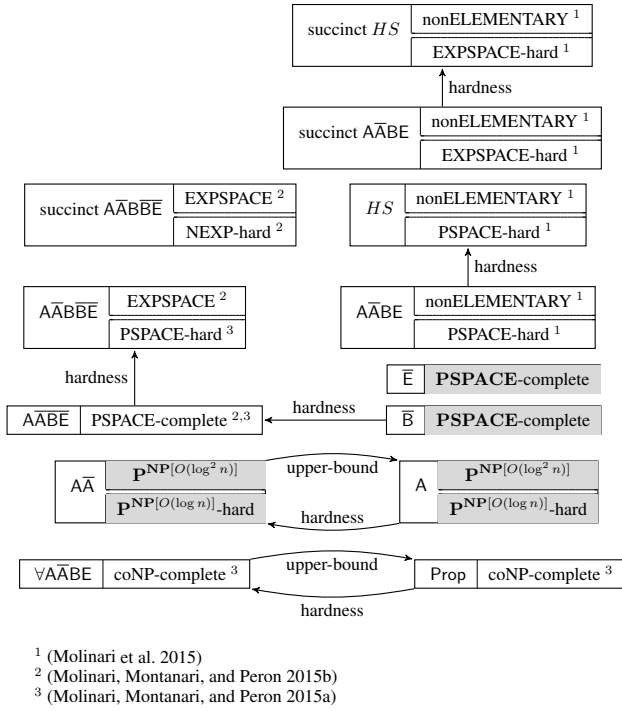


Figure 2: Complexity of model checking for HS fragments.

checking problem to the $\text{TB(SAT)}_{1 \times M}$ problem, thus proving that the former is in $\text{P}^{\text{NP}[O(\log^2 n)]}$. Finally, we show that model checking \bar{B} (and \bar{E}) formulas is PSPACE-hard by a reduction from the QBF problem.

In Figure 2 we summarize known complexity results, depicted in white boxes, and new ones, in grey boxes.

We conclude this section with a simple example (a simplified version of the one given in (Molinari, Montanari, and Peron 2015a)), showing that the considered fragments can express meaningful properties of state-transition systems.

Example 2. Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be the Kripke structure of Figure 3, where $\mathcal{AP} = \{r_0, r_1, e_0, e_1, x_0\}$. \mathcal{K} models the interactions between a scheduler S and two processes, \mathcal{P}_0 and \mathcal{P}_1 , which possibly ask for a shared resource. At

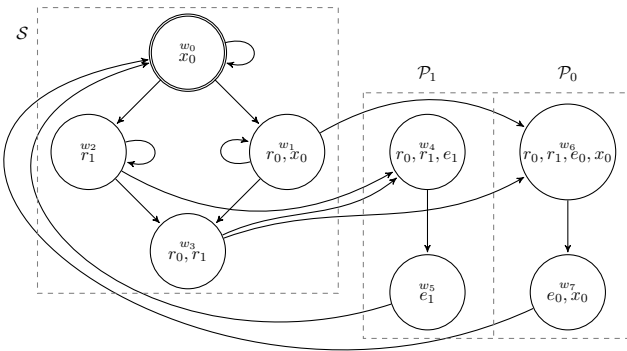


Figure 3: A simple state-transition system.

the initial state w_0 , S has not received any request from the processes yet, while in w_1 (resp., w_2) only \mathcal{P}_0 (resp., \mathcal{P}_1) has sent a request, and thus r_0 (resp., r_1) holds. As long as at most one process has sent a request, S is not forced to allocate the resource (w_1 and w_2 have self loops). At w_3 , both \mathcal{P}_0 and \mathcal{P}_1 are waiting for the shared resource, and hence both r_0 and r_1 hold there. State w_3 has transitions only towards w_4 and w_6 . At w_4 (resp., w_6) \mathcal{P}_1 (resp., \mathcal{P}_0) can access the resource: e_1 (resp., e_0) holds in w_4w_5 (resp., w_6w_7). Finally, from w_5 and w_7 , the system can only move to w_0 , where S waits for new requests from \mathcal{P}_0 and \mathcal{P}_1 .

Now, let \mathcal{P} be the set $\{r_0, r_1, e_0, e_1\}$ and let x_0 be an auxiliary proposition letter labelling the states w_0, w_1, w_6 , and w_7 , where S and \mathcal{P}_0 , but not \mathcal{P}_1 , are active.

We observe that $\mathcal{K} \models [A]\psi$ if and only if ψ holds over any (reachable) computation sub-interval.

It holds that $\mathcal{K} \models [A](r_0 \rightarrow (\langle A \rangle e_0 \vee \langle A \rangle \langle A \rangle e_0))$. Such a formula, belonging to A (and \bar{A}), expresses the following reachability property: if r_0 holds over some interval, then there is always a way to reach an interval over which e_0 holds. Obviously, this does not mean that all possible computations will necessarily lead to such an interval; however, the system will never fall in a state from which it is no more possible to satisfy requests from \mathcal{P}_0 .

It also holds that $\mathcal{K} \models [A](r_0 \wedge r_1 \rightarrow [A](e_0 \vee e_1 \vee \bigwedge_{p \in \mathcal{P}} \neg p))$ (in A and \bar{A}). Indeed, if both processes send a request to S (state w_3), then it immediately allocates the resource. Formally, if $r_0 \wedge r_1$ holds over some tracks (the only possible cases are w_3w_4 and w_3w_6), then in any possible subsequent interval of length 2 $e_0 \vee e_1$ holds, that is, \mathcal{P}_0 or \mathcal{P}_1 are executed, or $\bigwedge_{p \in \mathcal{P}} \neg p$ holds, if we consider tracks longer than 2. On the other hand, if only one process asks for the resource, then S can arbitrarily delay its allocation, that is, $\mathcal{K} \not\models [A](r_0 \rightarrow [A](e_0 \vee \bigwedge_{p \in \mathcal{P}} \neg p))$.

Finally, it holds that $\mathcal{K} \models x_0 \rightarrow \langle \bar{B} \rangle x_0$ (in \bar{B}), that is, any initial track satisfying x_0 (any such track involves states w_0, w_1, w_6 , and w_7 only) can be extended to the right in such a way that the resulting track still satisfies x_0 . This amounts to say that there exists a computation in which \mathcal{P}_1 starves. Notice that S and \mathcal{P}_0 can continuously interact without waiting for \mathcal{P}_1 . This is the case, for instance, when \mathcal{P}_1 does not ask for the shared resource at all.

$\text{P}^{\text{NP}[O(\log n)]}$ -hardness of the model checking problem for A (and \bar{A})

In this section, we prove that the model checking problem for formulas of A (and of \bar{A}), over finite Kripke structures, is $\text{P}^{\text{NP}[O(\log n)]}$ -hard. The complexity class $\text{P}^{\text{NP}[O(\log n)]}$ contains the problems decided by a deterministic polynomial time algorithm which requires only $O(\log n)$ queries to an NP oracle, being n the input size. Such a class is higher than both NP and co-NP in the polynomial time hierarchy, but lower than P^{NP} (a P^{NP} problem may require a polynomial number of queries). Like the class P^{NP} , it is closed under complementation.

A complete problem for $\text{P}^{\text{NP}[O(\log n)]}$ is PARITY(SAT) : given a set of Boolean formulas Γ , one has to decide whether

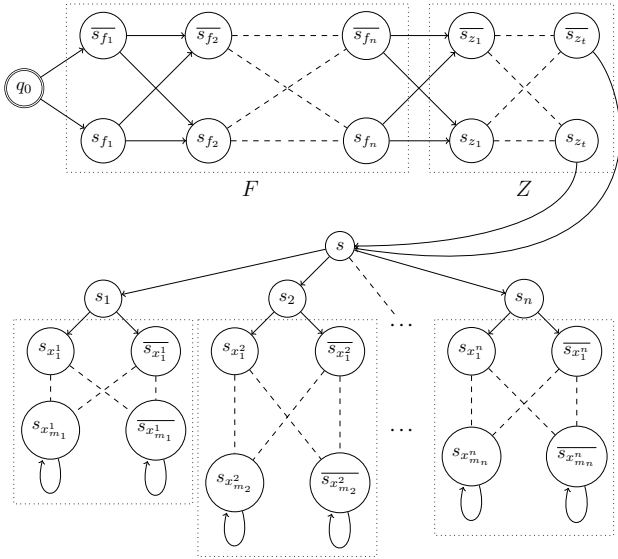


Figure 4: The Kripke structure \mathcal{K}_{PAR}^Γ .

the number of *satisfiable* formulas in Γ is odd or even (Gottlob 1995). In the following, we show how to reduce the PARITY(SAT) problem to the model checking problem for A.

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}_{PAR}^Γ and an A-formula Φ such that $\mathcal{K}_{PAR}^\Gamma \models \Phi$ if and only if the number of satisfiable Boolean formulas in Γ is *odd*.

We start by defining a Boolean formula, *parity*(F, Z), over two sets of Boolean variables, $F = \{f_1, \dots, f_n\}$ and $Z = \{z_1, \dots, z_t\}$, where $t = 3 \cdot (n - 1) + 1$. Such a formula allows one to decide the parity of the number of variables in F that evaluate to *true*. Z is a set of auxiliary variables, whose truth values are *functionally determined* by those assigned to the variables in F , that is, for a given truth assignment, the number of variables in F set to true is even if *parity*(F, Z) evaluates to true, and, in particular, its last variable z_t evaluates to true. The formula *parity*(F, Z) is defined as follows: *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 $t = 3 \cdot (n - 1) + 1$. 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: $\text{par}_1(f_1, z_1) = \neg f_1 \leftrightarrow z_1$, and, for all $i \geq 2$, $\text{par}_i(f_1, f_2, \dots, f_i, z_1, \dots, z_{\alpha+3}) = (z_{\alpha+1} \leftrightarrow (f_i \wedge \neg z_\alpha)) \wedge (z_{\alpha+2} \leftrightarrow (\neg f_i \wedge z_\alpha)) \wedge (z_{\alpha+3} \leftrightarrow (z_{\alpha+2} \vee z_{\alpha+1})) \wedge \text{par}_{i-1}(f_1, f_2, \dots, f_{i-1}, z_1, \dots, z_\alpha)$, where $\alpha = 3 \cdot (i - 2) + 1$. 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 $z_{\alpha+3}$. Notice that the length of *parity*(F, Z) is polynomial in n .

We now show how to build the Kripke structure \mathcal{K}_{PAR}^Γ , where a subset of paths 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 is shown in Figure 4. It features a couple 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 $\mathcal{AP} = \{p, q\} \cup F \cup Z \cup \{aux_i \mid 1 \leq i \leq n\} \cup \{x_{j_i}^i \mid 1 \leq i \leq n, 1 \leq j_i \leq m_i\}$, $W = \{q_0\} \cup \{s_{f_i}, \overline{s_{f_i}} \mid 1 \leq i \leq n\} \cup \{s_z, \overline{s_z} \mid z \in Z\} \cup \{s\} \cup \{s_i \mid 1 \leq i \leq n\} \cup \{s_{x_{j_i}^i}, \overline{s_{x_{j_i}^i}} \mid 1 \leq i \leq n, 1 \leq j_i \leq m_i\}$, $\delta = \{(q_0, s_{f_1}), (q_0, \overline{s_{f_1}})\} \cup \{(s_{f_i}, s_{f_{i+1}}), (s_{f_i}, \overline{s_{f_{i+1}}}), (\overline{s_{f_i}}, s_{f_{i+1}}), (\overline{s_{f_i}}, \overline{s_{f_{i+1}}}) \mid 1 \leq i < n\} \cup \{(s_{f_n}, s_{z_1}), (s_{f_n}, \overline{s_{z_1}}), (\overline{s_{f_n}}, s_{z_1}), (\overline{s_{f_n}}, \overline{s_{z_1}})\} \cup \{(s_{z_i}, s_{z_{i+1}}), (s_{z_i}, \overline{s_{z_{i+1}}}), (\overline{s_{z_i}}, s_{z_{i+1}}), (\overline{s_{z_i}}, \overline{s_{z_{i+1}}}) \mid 1 \leq i < t\} \cup \{(s_{z_t}, s), (\overline{s_{z_t}}, s)\} \cup \{(s, s_i), (s_i, s_{x_1^i}), (s_i, \overline{s_{x_1^i}}) \mid 1 \leq i \leq n\} \cup \{(s_{x_{j_i}^i}, s_{x_{j_i+1}^i}), (\overline{s_{x_{j_i}^i}}, s_{x_{j_i+1}^i}), (\overline{s_{x_{j_i}^i}}, \overline{s_{x_{j_i+1}^i}}), (\overline{s_{x_{j_i}^i}}, \overline{s_{x_{j_i+1}^i}}) \mid 1 \leq i \leq n, 1 \leq j_i < m_i\} \cup \{(s_{x_{m_i}^i}, s_{x_{m_i}^i}), (\overline{s_{x_{m_i}^i}}, \overline{s_{x_{m_i}^i}}) \mid 1 \leq i \leq n\}$, 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 $1 \leq i \leq n$, $1 \leq k_i \leq m_i$, $\mu(s_{x_{k_i}^i}) = \{aux_i\} \cup \{x_{j_i}^i \mid 1 \leq j_i \leq m_i\} \setminus \{x_{k_i}^i\}$.

According to the definition of \mathcal{K}_{PAR}^Γ , it holds that:

1. Each track 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 true in ρ iff $y \in \bigcap_{w \in \text{states}(\rho)} \mu(w)$). Conversely, for each truth assignment to the proposition letters in $F \cup Z$, there exists an initial track ρ , reaching the state s , encoding such an assignment. Notice that, among the initial tracks, those leading to s are exactly those satisfying $q \wedge \neg p$.
2. An initial track leading to s satisfies *parity*(F, Z) if the induced assignment sets an even number of f_i 's to true, and every $z \in Z$ to the truth value which is functionally implied by the values of the f_i 's.
3. A Boolean formula $\phi_i(x_1^i, \dots, x_{m_i}^i) \in \Gamma$ is satisfiable iff there exists a track $\tilde{\rho}$ starting from s and ending in a state s_i or $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, for some $j = 1, \dots, m_i$, such that $\mathcal{K}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$.

Finally, let us consider the A formula $\psi = q \wedge \neg p \wedge \text{parity}(F, Z) \wedge \bigwedge_{i=1}^n (f_i \leftrightarrow (A)(aux_i \wedge \phi_i(x_1^i, \dots, x_{m_i}^i)))$. In view of the above observations, ψ is satisfied by an initial track $\bar{\rho}$ 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 true and all $z \in Z$ accordingly, and (iii) for all $1 \leq i \leq n$, f_i is true iff there exists a track $\tilde{\rho}$ starting from s and ending in a state s_i or $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{K}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$. Notice that the length of ψ is polynomial in the input size.

Let us assume we are given an instance of PARITY(SAT) Γ with an *even* number of *satisfiable* Boolean formulas. There exists an initial track 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,

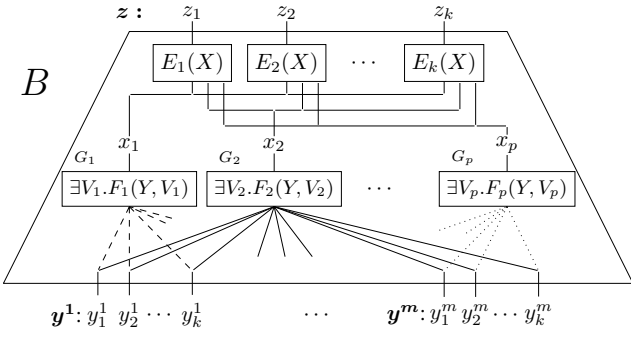


Figure 5: General form of a block.

and $\overline{s_{f_i}} \in \text{states}(\rho)$ otherwise. Moreover, ρ can be chosen so that $\mathcal{X}_{PAR}^\Gamma, \rho \models \text{parity}(F, Z)$. It immediately follows that, for all i , $\mathcal{X}_{PAR}^\Gamma, \rho \models f_i$ iff $\mathcal{X}_{PAR}^\Gamma, \rho \models \langle A \rangle (aux_i \wedge \phi_i(x_1^i, \dots, x_{m_i}^i))$, concluding that $\mathcal{X}_{PAR}^\Gamma, \rho \models \psi$.

Conversely, let ρ be an initial track such that $\mathcal{X}_{PAR}^\Gamma, \rho \models \psi$. It holds that ρ ends in s and sets an even number of f_i 's to true. Moreover, if $\mathcal{X}_{PAR}^\Gamma, \rho \models f_i$, then there exists $\tilde{\rho}$ starting from s and ending in s_i or $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{X}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$, hence $\phi_i(x_1^i, \dots, x_{m_i}^i)$ is satisfiable. If $\mathcal{X}_{PAR}^\Gamma, \rho \models \neg f_i$, then there exists no $\tilde{\rho}$ starting from s and ending in s_i or $s_{x_j^i}$ or $\overline{s_{x_j^i}}$, such that $\mathcal{X}_{PAR}^\Gamma, \tilde{\rho} \models \phi_i(x_1^i, \dots, x_{m_i}^i)$. So $\phi_i(x_1^i, \dots, x_{m_i}^i)$ 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* iff there exists an initial track ρ such that $\mathcal{X}_{PAR}^\Gamma, \rho \models \psi$. This amounts to say that Γ contains an *odd* number of satisfiable Boolean formulas (the PARITY(SAT) problem) iff $\mathcal{X}_{PAR}^\Gamma \models \Phi$, where $\Phi = \neg\psi$ (the model checking problem).

Theorem 1. *The model checking 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} .

A model checking algorithm for $A\overline{A}$

In this section, we provide a model checking algorithm for $A\overline{A}$ via a reduction to the problem $\text{TB}(\text{SAT})_{1 \times M}$. To the best of our knowledge, $\text{TB}(\text{SAT})_{1 \times M}$ is the first complete problem for the class $\mathbf{P}^{\text{NP}^{O(\log^2 n)}}$ that has been proposed in the literature (Schnoebelen 2003). We recall that $\mathbf{P}^{\text{NP}^{O(\log^2 n)}}$ is the class of problems decided by a deterministic polynomial time algorithm which requires only $O(\log^2 n)$ queries to an NP oracle, being n the input size. Like $\mathbf{P}^{\text{NP}^{O(\log n)}}$, $\mathbf{P}^{\text{NP}^{O(\log^2 n)}}$ is higher than both NP and co-NP and lower than \mathbf{P}^{NP} in the polynomial time hierarchy, and it is closed under complementation.

To start with, we give a short account of the $\text{TB}(\text{SAT})_{1 \times M}$ problem. As a preliminary step, we need to introduce the notion of *block*. A block (see Figure 5) is a circuit B whose

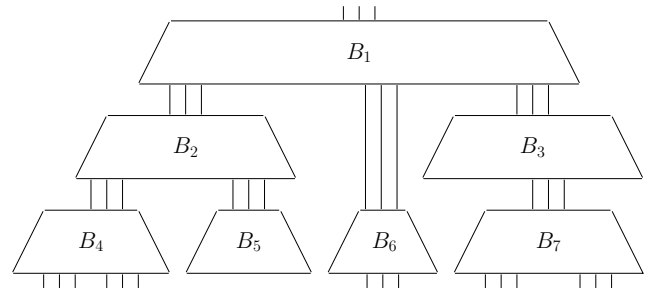


Figure 6: A tree of blocks (B_5 has degree $m = 0$).

input lines are organized in m bit vectors $\mathbf{y}^1, \dots, \mathbf{y}^m$, each one with k entries: $\mathbf{y}^i = (y_1^i, \dots, y_k^i)$. The values m and k are respectively called the *degree* and the *width* of B . Input lines are connected to p internal gates G_1, \dots, G_p . Each gate G_i queries a SAT oracle to decide whether the associated Boolean formula $F_i(Y, V_i)$ is satisfiable, 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 ($V_i \cap V_j = \emptyset$ for $j \neq i$). The output of G_i is conveyed by x_i , which evaluates to \top if and only if $F_i(Y, V_i)$ is satisfiable. Finally, k classical circuits (without oracle queries) E_1, \dots, E_k compute, from $X = \{x_1, \dots, x_p\}$, the final k bits, output of the block B , conveyed by the output lines z_1, \dots, z_k (thus, the number of output lines equals the *width*). The size of B is the total number of gates, as usual, plus the sizes of all the formulas $F_i(Y, V_i)$.

Blocks of the same width can be combined together to form a tree-structured complex circuit, called *tree of blocks* (see Figure 6 for an example). Every block B_i in the tree 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 some levels $\leq d - 1$, is at level d . In Figure 6, 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.

TB(SAT) is the problem of deciding whether a specific output z_i (of the root) of a tree of blocks T is \top or \perp , given the values for the inputs (of the leaf blocks) of T . As proved in (Schnoebelen 2003), TB(SAT) is \mathbf{P}^{NP} -complete.

The problem $\text{TB}(\text{SAT})_{1 \times M}$ is TB(SAT) with a restriction: the Boolean formulas (SAT queries) of each block B of the tree of blocks must be of the following form: $\exists \ell_1, \dots, \ell_m \exists V'_i F_i(y_{\ell_1}^1, \dots, y_{\ell_m}^m, \ell_1, \dots, \ell_m, V'_i)$, with $\ell_1, \dots, \ell_m \in \{1, \dots, k\}$, 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 “ $1 \times M$ ”. As a matter of fact, the existential quantification over the indexes ℓ_1, \dots, ℓ_m is an abuse of notation borrowed from (Schnoebelen 2003); however, $\exists \ell_j \in \{1, \dots, k\}$ is just a shorthand for k bits (belonging to its set of private variables) “ $\ell_j = 1$ ”, ..., “ $\ell_j = k$ ”, among which exactly one is 1. In (Schnoebelen 2003), the author proves that $\text{TB}(\text{SAT})_{1 \times M}$ is $\mathbf{P}^{\text{NP}^{O(\log^2 n)}}$ -complete.

We now outline a reduction from the model checking problem for $A\overline{A}$ to the problem $\text{TB}(\text{SAT})_{1 \times M}$, showing how

to build a suitable tree of blocks that allows us to decide the former problem.

Given an $\overline{A\bar{A}}$ formula ψ (w.l.o.g., we assume that ψ contains only *existential* modalities) to check over a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$, where $W = \{w_1, \dots, w_{|W|}\}$, we consider its negation $\neg\psi$ and build from it a tree 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 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 $k = |W|$.

The notion of *modal subformula* plays a fundamental role in the reduction, as it allows us to build the tree recursively.

Definition 7. Let χ be an $\overline{A\bar{A}}$ formula. The set of its modal subformulas, denoted by $\text{ModSubf}(\chi)$, is the set of subformulas of χ of the form $\langle A \rangle \chi'$ or $\langle \bar{A} \rangle \chi'$, for some $\overline{A\bar{A}}$ formula χ' , not in the scope of any other modality $\langle A \rangle$ or $\langle \bar{A} \rangle$.

For instance, we have that $\text{ModSubf}(\langle A \rangle \langle \bar{A} \rangle q) = \{\langle A \rangle \langle \bar{A} \rangle q\}$ and $\text{ModSubf}(\langle A \rangle (\langle A \rangle p) \wedge (\langle A \rangle \langle \bar{A} \rangle q)) \rightarrow \langle A \rangle p = \{\langle A \rangle p, \langle A \rangle \langle \bar{A} \rangle q\}$;

Starting from B_{root} , $T_{\mathcal{K}, \neg\psi}$ is built recursively as follows. If some block B is associated with a formula φ , then:

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

Then, the procedure is applied recursively to all the generated children of B , terminating when $\text{ModSubf}(\varphi) = \emptyset$. Notice that B has thus *degree* $m = |\text{ModSubf}(\varphi)|$.

The procedure allows us to determine the general structure of $T_{\mathcal{K}, \neg\psi}$. We now need to describe the internal structure of its blocks. As a preliminary step, we suitably transform some $\overline{A\bar{A}}$ formulas φ into Boolean ones. To this end, we replace all the occurrences of proposition letters and modal subformulas in φ by Boolean variables.

Definition 8. Given $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an $\overline{A\bar{A}}$ formula χ , we define $\bar{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})$, where $V_{\mathcal{AP}} = \{v_p \mid p \in \mathcal{AP}\}$ and $V_{\text{modSubf}} = \{v_{\chi'} \mid \chi' \in \text{ModSubf}(\chi)\}$ are sets of Boolean variables, as the Boolean formula obtained from χ by replacing each (occurrence of a) modal subformula $\chi' \in \text{ModSubf}(\chi)$ by the variable $v_{\chi'}$, and then each (occurrence of a) proposition $p \in \mathcal{AP}$ by the variable v_p .

Given a track $\rho \in \text{Trk}_{\mathcal{K}}$ and an $\overline{A\bar{A}}$ formula χ , it is easy to prove (by induction on the complexity of $\bar{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})$) 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 $\chi' \in \text{ModSubf}(\chi)$, then it holds that $\mathcal{K}, \rho \models \chi \iff \omega(\bar{\chi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$.

We are now ready to describe the internal structure of a (generic) block B in $T_{\mathcal{K}, \neg\psi}$ for a formula φ . Let us assume that B has FORWARD type. We refer again to Figure 5. First of all, each output line z_i is directly linked to the output x_i of the oracle gate G_i , avoiding circuits E_1, \dots, E_k , which can be omitted from B . The formula

$F_i(Y, V)$ of G_i ($1 \leq i \leq |W|$) is defined in Figure 7, where $V = V_{last} \cup V_{track} \cup V_{\mathcal{AP}} \cup V_{\text{modSubf}}$ is the set of (private) variables, with $V_{last} = \{v_1, v_2, \dots, v_{|W|}\}$, $V_{track} = \{v_1^1, \dots, v_{|W|}^1, v_1^2, \dots, v_{|W|}^2, \dots, v_1^{|W|^2+2}, \dots, v_{|W|}^{|W|^2+2}\}$, and $V_{\mathcal{AP}}$ and V_{modSubf} as in Definition 8 (for the sake of simplicity, we decided to write V for V_i). In Figure 7, $VAL(w_j, p)$ is just a shorthand for \top if $p \in \mu(w_j)$, and \perp otherwise; y_j^ξ denotes the input of B linked to the j -th output of the child B' associated with the formula ξ . The construction of a block of BACKWARD type is symmetric (we just replace the first conjunct v_i^1 of $F_i(Y, V)$ by v_i).

Proposition 1. Given a track $\rho \in \text{Trk}_{\mathcal{K}}$, with $|\rho| \leq |W|^2 + 2$, there is a truth assignment ω to the variables in V which satisfies the formula $\text{track}(V_{track}, V_{last}, V_{\mathcal{AP}})$, and ω is such that for any $1 \leq t \leq |\rho|$ and $1 \leq j \leq |W|$, $\rho(t) = w_j \iff \omega(v_j^t) = \top$ and $\omega(v_j^{|\rho|}) = \omega(v_j)$; moreover, for any $p \in \mathcal{AP}$, $\omega(v_p) = \top \iff \mathcal{K}, \rho \models p$. Conversely, if there is a truth assignment ω to the variables in V satisfying the s -th disjunct of $\text{track}(V_{track}, V_{last}, V_{\mathcal{AP}})$, then there is a track $\rho \in \text{Trk}_{\mathcal{K}}$, with $|\rho| = s + 1$, such that, for any $1 \leq t \leq |\rho|$ and $1 \leq j \leq |W|$, $\rho(t) = w_j \iff \omega(v_j^t) = \top$ and, for any $p \in \mathcal{AP}$, $\mathcal{K}, \rho \models p \iff \omega(v_p) = \top$.

The following theorem holds (hereafter, we write $B(z_i)$ for the output line z_i of B).

Theorem 2. Given an $\overline{A\bar{A}}$ formula ψ and a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$, for every block B of $T_{\mathcal{K}, \neg\psi}$, if B is associated with an $\overline{A\bar{A}}$ formula φ , then

- if B is FORWARD, $\forall i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ iff there exists a track $\rho \in \text{Trk}_{\mathcal{K}}$ such that $\text{fst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$;
- if B is BACKWARD, $\forall i \in \{1, \dots, |W|\}$, $B(z_i) = \top$ iff there exists a track $\rho \in \text{Trk}_{\mathcal{K}}$ such that $\text{lst}(\rho) = w_i$ and $\mathcal{K}, \rho \models \varphi$;

Proof. The proof is by induction on the level L of the block B . We skip the base case in which $L = 1$, because it is an easy simplification of what we prove in the inductive step.

Let us consider a FORWARD block B 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 track $\rho \in \text{Trk}_{\mathcal{K}}$ 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 (Molinari, Montanari, and Peron 2015b), it is proved that if φ is an $\overline{A\bar{A}}$ formula and $\mathcal{K}, \rho \models \varphi$ (as in this case), there exists a track $\rho' \in \text{Trk}_{\mathcal{K}}$, 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 1, there exists a truth assignment ω to the variables in V , that satisfies $\text{track}(V_{track}, V_{last}, V_{\mathcal{AP}})$, 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 $L \geq 2$, $\text{ModSubf}(\varphi) \neq \emptyset$ holds. So we consider a FORWARD child B' of B (if any), at a level lower than L , associated with some ξ such that $\langle A \rangle \xi \in \text{ModSubf}(\varphi)$,

$$F_i(Y, V) = \exists V \left(v_i^1 \wedge \text{track}(V_{\text{track}}, V_{\text{last}}, V_{\mathcal{AP}}) \wedge \overline{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}}) \wedge \bigwedge_{\langle A \rangle \xi \in \text{ModSubf}(\varphi)} (v_{\langle A \rangle \xi} \leftrightarrow \bigvee_{j=1}^{|\mathcal{W}|} (v_j \wedge y_j^\xi)) \wedge \bigwedge_{\langle \overline{A} \rangle \xi' \in \text{ModSubf}(\varphi)} (v_{\langle \overline{A} \rangle \xi'} \leftrightarrow \bigvee_{j=1}^{|\mathcal{W}|} (v_j^1 \wedge y_j^{\xi'})) \right),$$

where $\text{one}_t(v_1^t, v_2^t, \dots, v_{|\mathcal{W}|}^t) = \left(\bigvee_{j=1}^{|\mathcal{W}|} v_j^t \right) \wedge \left(\bigwedge_{j=1}^{|\mathcal{W}|} \bigwedge_{k=j+1}^{|\mathcal{W}|} \neg(v_j^t \wedge v_k^t) \right)$, $\text{edge}_t(v_1^t, \dots, v_{|\mathcal{W}|}^t, v_1^{t+1}, \dots, v_{|\mathcal{W}|}^{t+1}) = \bigvee_{(w_k, w_j) \in \delta} (v_k^t \wedge v_j^{t+1})$, and

$$\text{track}(V_{\text{track}}, V_{\text{last}}, V_{\mathcal{AP}}) = \bigvee_{\ell=2}^{|\mathcal{W}|^2+2} \left[\bigwedge_{t=1}^{\ell} \text{one}_t(v_1^t, v_2^t, \dots, v_{|\mathcal{W}|}^t) \wedge \bigwedge_{t=1}^{\ell-1} \text{edge}_t(v_1^t, \dots, v_{|\mathcal{W}|}^t, v_1^{t+1}, \dots, v_{|\mathcal{W}|}^{t+1}) \wedge \bigwedge_{t=1}^{|\mathcal{W}|} (v_t^\ell \leftrightarrow v_t) \wedge \bigwedge_{p \in \mathcal{AP}} \left((v_p \rightarrow \bigwedge_{t=1}^{\ell} \bigwedge_{j=1}^{|\mathcal{W}|} (v_j^t \rightarrow \text{VAL}(w_j, p))) \wedge (\neg v_p \rightarrow \bigvee_{t=1}^{\ell} \bigwedge_{j=1}^{|\mathcal{W}|} (v_j^t \rightarrow \neg \text{VAL}(w_j, p))) \right) \right]$$

Figure 7: Definition of $F_i(Y, V)$ for a block of FORWARD type.

and we apply the inductive hypothesis, which ensures that, for all j , $B'(z_j) = \top$ iff there exists a track $\bar{\rho} \in \text{Trk}_{\mathcal{X}}$ such that $\text{fst}(\bar{\rho}) = w_j$ and $\mathcal{K}, \bar{\rho} \models \xi$. Thus, $\mathcal{K}, \rho' \models \langle A \rangle \xi$ iff there exists $\bar{\rho} \in \text{Trk}_{\mathcal{X}}$, with $\text{fst}(\bar{\rho}) = \text{lst}(\rho') = w_j$, for some j , and $\mathcal{K}, \bar{\rho} \models \xi$ iff $B'(z_j) (= y_j^\xi) = \top$. 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 $y_j^\xi = \perp$, and so $\bigvee_{s=1}^{|\mathcal{W}|} (\omega(v_s) \wedge y_s^\xi)$ is false, and ω must be such that $\omega(v_{\langle A \rangle \xi}) = \perp$. To conclude, $\mathcal{K}, \rho' \models \langle A \rangle \xi$ iff $\omega(v_{\langle A \rangle \xi}) = \top$ ($\star\star$). The symmetric reasoning can be applied to BACKWARD children of B . Since $\mathcal{K}, \rho' \models \varphi$, by (\star) and ($\star\star$), we have $\omega(\overline{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$.

Let us now prove the implication (\Rightarrow). If $B(z_i) = \top$, then it holds that there exists a truth assignment ω of V satisfying $F_i(Y, V)$. In particular, ω satisfies $\text{track}(V_{\text{track}}, V_{\text{last}}, V_{\mathcal{AP}})$ and v_i^1 , thus, by Proposition 1, there exists a track $\rho \in \text{Trk}_{\mathcal{X}}$ 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 $\langle A \rangle \xi \in \text{ModSubf}(\varphi)$, $\mathcal{K}, \rho \models \langle A \rangle \xi$ iff $\omega(v_{\langle A \rangle \xi}) = \top$, and symmetrically, for all $\langle \overline{A} \rangle \xi' \in \text{ModSubf}(\varphi)$, $\mathcal{K}, \rho \models \langle \overline{A} \rangle \xi'$ iff $\omega(v_{\langle \overline{A} \rangle \xi'}) = \top$. Since $\omega(\overline{\varphi}(V_{\mathcal{AP}}, V_{\text{modSubf}})) = \top$, then $\mathcal{K}, \rho \models \varphi$. \square

The two next corollaries immediately follow.

Corollary 1. *Let ψ be an $\overline{A\overline{A}}$ formula, $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_1)$ be a Kripke structure, and B_{root} be the root block of $T_{\mathcal{K}, \neg\psi}$. It holds that $B_{\text{root}}(z_1) = \perp \iff \mathcal{K} \models \psi$.*

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

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

PSPACE-completeness of \overline{B} and \overline{E}

In this section, we prove that the model checking problem for formulas of \overline{B} and of \overline{E} , over finite Kripke structures, is PSPACE-complete. In (Molinari, Montanari, and

Peron 2015b), the authors prove that the fragment $\overline{A\overline{A}B\overline{E}}$ is in PSPACE. Membership of \overline{B} and of \overline{E} to PSPACE thus immediately follows. Here, we prove the PSPACE-hardness of the problem for \overline{B} formulas by means of a reduction from the QBF problem, that is, the problem of determining the truth of a fully-quantified Boolean formula in prenex normal form, which is known to be PSPACE-complete (see, for example, (Sipser 2012)). The proof for \overline{B} can easily be modified to show the PSPACE-hardness of the symmetric fragment \overline{E} .

Let ψ be a quantified Boolean formula $Q_n x_n Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1)$, where, for $i = 1, \dots, n$, $Q_i \in \{\exists, \forall\}$ and $\phi(x_n, x_{n-1}, \dots, x_1)$ is a quantifier-free Boolean formula over the set of variables $\text{Var} = \{x_n, \dots, x_1\}$. We define a Kripke structure $\mathcal{K}_{QBF}^{\text{Var}}$, whose initial tracks represent all the possible assignments to the variables of Var . For each variable $x \in \text{Var}$, $\mathcal{K}_{QBF}^{\text{Var}}$ features a pair of states w_x^\top and w_x^\perp , that represent a \top and \perp truth assignment to x , respectively. An example of $\mathcal{K}_{QBF}^{\text{Var}}$, with $\text{Var} = \{x, y, z\}$, is given in Figure 8.

Formally, let $\mathcal{K}_{QBF}^{\text{Var}} = (\mathcal{AP}, W, \delta, \mu, w_0)$, where:

- $\mathcal{AP} = \text{Var} \cup \{s\} \cup \{\tilde{x}_i \mid 1 \leq i \leq n\}$;
- $W = \{w_{x_i}^\ell \mid 1 \leq i \leq n, \ell \in \{\perp, \top\}\} \cup \{w_0, w_1, \text{sink}\}$;
- if $n = 0$, $\delta = \{(w_0, w_1), (w_1, \text{sink}), (\text{sink}, \text{sink})\}$;
- if $n > 0$, $\delta = \{(w_0, w_1), (w_1, w_{x_n}^\top), (w_1, w_{x_n}^\perp)\} \cup \{(w_{x_i}^\ell, w_{x_{i-1}}^m) \mid \ell, m \in \{\perp, \top\}, 2 \leq i \leq n\} \cup \{(w_{x_1}^\top, \text{sink}), (w_{x_1}^\perp, \text{sink}), (\text{sink}, \text{sink})\}$.
- $\mu(w_0) = \mu(w_1) = \text{Var} \cup \{s\} \cup \{\tilde{x}_i \mid x_i \in \text{Var}\}$;
- for all $1 \leq i \leq n$, $\mu(w_{x_i}^\top) = \text{Var} \cup \{\tilde{x}_j \mid 1 \leq j \leq i\}$ and $\mu(w_{x_i}^\perp) = (\text{Var} \setminus \{x_i\}) \cup \{\tilde{x}_j \mid 1 \leq j \leq i\}$;
- $\mu(\text{sink}) = \text{Var}$.

The quantified Boolean formula ψ is reduced to the \overline{B} formula $\xi = s \rightarrow \xi_n$, where

$$\xi_i = \begin{cases} \phi(x_n, x_{n-1}, \dots, x_1) & i = 0 \\ \overline{B}(\tilde{x}_i \wedge \xi_{i-1}) & i > 0 \wedge Q_i = \exists \\ \overline{B}(\tilde{x}_i \rightarrow \xi_{i-1}) & i > 0 \wedge Q_i = \forall \end{cases}$$

Notice that both $\mathcal{K}_{QBF}^{\text{Var}}$ and ξ can be built by using logarithmic working space. In Theorem 3, we shall show the correct-

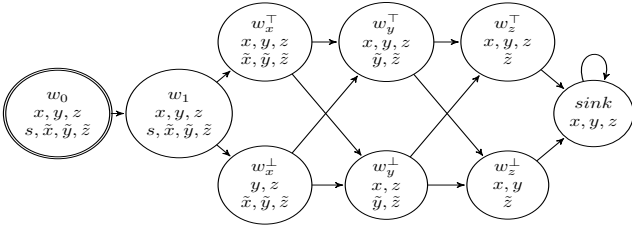


Figure 8: The Kripke structure $\mathcal{K}_{QBF}^{x,y,z}$.

ness of the reduction, namely, that ψ is true iff $\mathcal{K}_{QBF}^{Var} \models \xi$. As a preliminary step, we introduce some technical definitions and an auxiliary lemma.

Given a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and a \bar{B} formula χ , we denote by $pl(\chi)$ the set of proposition letters occurring in χ and by $\mathcal{K}_{|pl(\chi)}$ the structure obtained from \mathcal{K} by restricting the labelling of each state to $pl(\chi)$, namely, the Kripke structure $(\overline{\mathcal{AP}}, W, \delta, \bar{\mu}, w_0)$, where $\overline{\mathcal{AP}} = \mathcal{AP} \cap pl(\chi)$ and $\bar{\mu}(w) = \mu(w) \cap pl(\chi)$, for all $w \in W$. Moreover, for $v \in W$, we denote by $reach(\mathcal{K}, v)$ the subgraph of \mathcal{K} induced by the states reachable from v , namely, the Kripke structure $(\mathcal{AP}, W', \delta', \mu', v)$, where $W' = \{w \in W \mid \text{there exists } \rho \in \text{Trk}_{\mathcal{K}} \text{ with } \text{fst}(\rho) = v \text{ and } \text{lst}(\rho) = w\}$, $\delta' = \delta \cap (W' \times W')$, and $\mu'(w) = \mu(w)$, for all $w \in W'$. As usual, we say that two Kripke structures $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w'_0)$ are *isomorphic* (written $\mathcal{K} \sim \mathcal{K}'$) iff there is a *bijection* $f : W \mapsto W'$ such that (i) $f(w_0) = w'_0$; (ii) for all $u, v \in W$, $(u, v) \in \delta$ iff $(f(u), f(v)) \in \delta'$; (iii) for all $v \in W$, $\mu(v) = \mu'(f(v))$. Finally, if $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$ is the abstract interval model induced by a Kripke structure \mathcal{K} and $\rho \in \text{Trk}_{\mathcal{K}}$, we denote $\sigma(\rho)$ by $\mathcal{L}(\mathcal{K}, \rho)$.

Let \mathcal{K} and \mathcal{K}' be two Kripke structures. The following lemma, which is an immediate consequence of Lemma 1 of (Molinari, Montanari, and Peron 2015a), states that, for any \bar{B} formula ψ , if the same set of proposition letters, restricted to $pl(\psi)$, holds over two tracks $\rho \in \text{Trk}_{\mathcal{K}}$ and $\rho' \in \text{Trk}_{\mathcal{K}'}$, and the subgraphs consisting of the states reachable from, respectively, $\text{lst}(\rho)$ and $\text{lst}(\rho')$ are isomorphic, then ρ and ρ' are equivalent with respect to ψ .

Lemma 1. *Given a \bar{B} formula ψ , two Kripke structures $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w'_0)$, and two tracks $\rho \in \text{Trk}_{\mathcal{K}}$ and $\rho' \in \text{Trk}_{\mathcal{K}'}$ such that $\mathcal{L}(\mathcal{K}_{|pl(\psi)}, \rho) = \mathcal{L}(\mathcal{K}'_{|pl(\psi)}, \rho')$ and $reach(\mathcal{K}_{|pl(\psi)}, \text{lst}(\rho)) \sim reach(\mathcal{K}'_{|pl(\psi)}, \text{lst}(\rho'))$, it holds that $\mathcal{K}, \rho \models \psi$ iff $\mathcal{K}', \rho' \models \psi$.*

Theorem 3. *The model checking problem for \bar{B} formulas, over finite Kripke structures, is PSPACE-hard (under LOGSPACE reductions).*

Proof. Let ψ be the quantified Boolean formula $Q_n x_n Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1)$. We prove by induction on the number n of variables of ψ that ψ is true if and only if $\mathcal{K}_{QBF}^{x_n, \dots, x_1} \models \xi$. In the following, $\phi(x_n, x_{n-1}, \dots, x_1) \{x_i/v\}$, with $v \in \{\top, \perp\}$, denotes the formula obtained from $\phi(x_n, x_{n-1}, \dots, x_1)$ by replacing all occurrences of x_i by v . Notice that $\mathcal{K}_{QBF}^{x_n, x_{n-1}, \dots, x_1}$ and

$\mathcal{K}_{QBF}^{x_{n-1}, \dots, x_1}$ are isomorphic if restricted to the states $w_{x_{n-1}}^\top, w_{x_{n-1}}^\perp, \dots, w_{x_1}^\top, w_{x_1}^\perp, \text{sink}$ (i.e., the initial parts of both Kripke structures are eliminated), and the labelling of states is suitably restricted. Moreover, notice that only the track $w_0 w_1$ satisfies the proposition letter s in ξ .

Base case ($n = 0$). In this case, $\psi = \phi(\emptyset)$ is a Boolean formula devoid of variables. If ψ is true, then in particular $\mathcal{K}_{QBF}^\emptyset, w_0 w_1 \models \phi(\emptyset)$ and thus $\mathcal{K}_{QBF}^\emptyset \models s \rightarrow \phi(\emptyset) (= \xi)$. Conversely, if $\mathcal{K}_{QBF}^\emptyset \models s \rightarrow \phi(\emptyset)$, then $\mathcal{K}_{QBF}^\emptyset, w_0 w_1 \models \phi(\emptyset)$, and since $\phi(\emptyset)$ has no variables, it must be true.

Case $n \geq 1$. We first prove that if the formula $\psi = Q_n x_n Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1)$ is true, then $\mathcal{K}_{QBF}^{x_n, \dots, x_1} \models \xi$.

If $Q_n = \exists$, one possibility is that the quantified Boolean formula $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi'(x_{n-1}, \dots, x_1)$ is true, where $\phi'(x_{n-1}, \dots, x_1) = \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\top\}$. By inductive hypothesis, it holds that $\mathcal{K}_{QBF}^{x_{n-1}, \dots, x_1} \models \xi'$, where $\xi' = s \rightarrow \xi'_{n-1}$ and $\xi'_{n-1} = \xi_{n-1} \{x_n/\top\}$. Thus $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w'_0 w'_1 \models \xi'_{n-1}$ (w'_0 and w'_1 are the two “leftmost” states of the structure $\mathcal{K}_{QBF}^{x_n, \dots, x_1}$). By Lemma 1, $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi'_{n-1}$. Since every right extension of $w_0 w_1 w_{x_n}^\top$ models x_n , $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi_{n-1}$, and so $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 \models \langle \bar{B} \rangle (x_n \wedge \xi_{n-1}) (= \xi_n)$. To conclude, $\mathcal{K}_{QBF}^{x_n, \dots, x_1} \models s \rightarrow \xi_n (= \xi)$. The only other possible case is that $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi'(x_{n-1}, \dots, x_1)$ is true, with $\phi'(x_{n-1}, \dots, x_1) = \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\perp\}$. As before, it follows that $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\perp \models \xi_{n-1} \{x_n/\perp\}$ and thus $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 \models \langle \bar{B} \rangle (\tilde{x}_n \wedge \xi_{n-1})$.

Now, let $Q_n = \forall$. Both the formula $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\top\}$ and the formula $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\perp\}$ are true. By reasoning as in the existential case, we have that $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\perp \models \xi_{n-1}$. Thus, $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 \models [\bar{B}] (\tilde{x}_n \rightarrow \xi_{n-1})$ and $\mathcal{K}_{QBF}^{x_n, \dots, x_1} \models s \rightarrow [\bar{B}] (\tilde{x}_n \rightarrow \xi_{n-1})$.

We now prove that if $\mathcal{K}_{QBF}^{x_n, \dots, x_1} \models \xi$, then ψ is true.

If $Q_n = \exists$, then $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 \models \langle \bar{B} \rangle (\tilde{x}_n \wedge \xi_{n-1})$. So $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi_{n-1}$ or $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\perp \models \xi_{n-1}$. In the former case, $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi_{n-1} \{x_n/\top\}$ (since every right extension of $w_0 w_1 w_{x_n}^\top$ models x_n). By Lemma 1, $\mathcal{K}_{QBF}^{x_{n-1}, \dots, x_1}, w'_0 w'_1 \models \xi_{n-1} \{x_n/\top\}$, and $\mathcal{K}_{QBF}^{x_{n-1}, \dots, x_1} \models s \rightarrow \xi_{n-1} \{x_n/\top\}$. By inductive hypothesis, $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\top\}$ is true, thus $\psi = \exists x_n Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1)$ is true. In the latter case, we symmetrically have that $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\perp\}$ is true, so $\psi = \exists x_n Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1)$ is true.

If $Q_n = \forall$, then $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 \models [\bar{B}] (\tilde{x}_n \rightarrow \xi_{n-1})$. Therefore, both $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\top \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n, \dots, x_1}, w_0 w_1 w_{x_n}^\perp \models \xi_{n-1}$. By reasoning as in the existential case, we have that both the formula $Q_{n-1} x_{n-1} \dots Q_1 x_1 \phi(x_n, x_{n-1}, \dots, x_1) \{x_n/\top\}$ and the

formula $Q_{n-1}x_{n-1} \cdots Q_1x_1\phi(x_n, x_{n-1}, \dots, x_1)\{x_n/\perp\}$ are true, thus $\psi = \forall x_n Q_{n-1}x_{n-1} \cdots Q_1x_1\phi(x_n, x_{n-1}, \dots, x_1)$ is true. \square

Conclusions and future work

In this paper, we investigated the complexity of the model checking problem for all fragments of $\overline{AAB\overline{E}}$. From the **PSPACE**-hardness of the problem for \overline{B} and \overline{E} , it immediately follows that model checking is **PSPACE**-complete for any fragment of $\overline{AAB\overline{E}}$ including modalities $\langle \overline{B} \rangle$ or $\langle \overline{E} \rangle$. As for the fragments A , \overline{A} , and $A\overline{A}$, we showed that they belong to $\mathbf{P}^{\mathbf{NP}[O(\log^2 n)]}$ and are $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$ -hard, thus leaving open the question whether their model checking problem can be solved by less than $O(\log^2 n)$ queries to an **NP** oracle or a tighter lower bound can be proved (or both). As a matter of fact, any attempt to reduce $\text{TB}(\text{SAT})_{1 \times M}$ to the model checking problem for A or $A\overline{A}$ failed, as in the reduction we need an HS formula of length $\Theta(n^{\log n})$, which clearly cannot be generated in polynomial time.

As for future work, we are studying HS fragments featuring modality $\langle B \rangle$, modality $\langle E \rangle$, or both of them: although it may seem counterintuitive, B and \overline{B} (the same holds for E and \overline{E}) do not behave the same, and completely different techniques are needed to deal with the former. On the one hand, we would like to establish whether **PSPACE**-completeness can be preserved by adding modality $\langle B \rangle$ (or, alternatively, $\langle E \rangle$) to $\overline{AAB\overline{E}}$, thus improving the existing **EXSPACE** algorithms for $\overline{AAB\overline{B\overline{E}}}$ and $\overline{AA\overline{E\overline{B\overline{E}}}}$ (Molinari, Montanari, and Peron 2015b). On the other hand, we would like to check whether the interplay between $\langle B \rangle$ and $\langle E \rangle$ actually causes a significant complexity blowup, as it happens for the satisfiability problem (Bresolin et al. 2014).

References

- Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.
- Bresolin, D.; Goranko, V.; Montanari, A.; and Sciavicco, G. 2009. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic* 161(3):289–304.
- Bresolin, D.; Goranko, V.; Montanari, A.; and Sala, P. 2010. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation* 20(1):133–166.
- Bresolin, D.; Montanari, A.; Sala, P.; and Sciavicco, G. 2011. What’s decidable about Halpern and Shoham’s interval logic? The maximal fragment $\overline{AB\overline{B\overline{L}}}$. In *LICS*, 387–396. IEEE Computer Society.
- Bresolin, D.; Della Monica, D.; Goranko, V.; Montanari, A.; and 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.
- Clarke, E. M.; Grumberg, O.; and Peled, D. A. 2002. *Model Checking*. MIT Press.
- Giunchiglia, F., and Traverso, P. 1999. Planning as model checking. In *ECP*, LNCS 1809, 1–20. Springer.
- Gottlob, G. 1995. NP Trees and Carnap’s Modal Logic. *Journal of the ACM* 42(2):421–457.
- Halpern, J. Y., and Shoham, Y. 1991. A propositional modal logic of time intervals. *Journal of the ACM* 38(4):935–962.
- Lodaya, K. 2000. Sharpening the undecidability of interval temporal logic. In *ASIAN*, LNCS 1961, 290–298. Springer.
- Lomuscio, A., and Michaliszyn, J. 2013. An epistemic Halpern-Shoham logic. In *IJCAI*, 1010–1016.
- Lomuscio, A., and Michaliszyn, J. 2014. Decidability of model checking multi-agent systems against a class of EHS specifications. In *ECAI*, 543–548.
- Lomuscio, A., and Michaliszyn, J. 2015. Model checking epistemic Halpern-Shoham logic extended with regular expressions. *CoRR abs/1509.00608*.
- Lomuscio, A., and Raimondi, F. 2006. MCMAS: A model checker for multi-agent systems. In *TACAS*, LNCS 3920, 450–454. Springer.
- Marcinkowski, J., and Michaliszyn, J. 2014. The undecidability of the logic of subintervals. *Fundamenta Informaticae* 131(2):217–240.
- Molinari, A.; Montanari, A.; Murano, A.; Perelli, G.; and Peron, A. 2015. Checking interval properties of computations. *Acta Informatica*. Accepted for publication.
- Molinari, A.; Montanari, A.; and Peron, A. 2015a. Complexity of ITL model checking: some well-behaved fragments of the interval logic HS. In *TIME*, 90–100.
- Molinari, A.; Montanari, A.; and Peron, A. 2015b. A model checking procedure for interval temporal logics based on track representatives. In *CSL*, 193–210.
- Montanari, A.; Murano, A.; Perelli, G.; and Peron, A. 2014. Checking interval properties of computations. In *TIME*, 59–68.
- Montanari, A.; Puppis, G.; and Sala, P. 2010. Maximal decidable fragments of Halpern and Shoham’s modal logic of intervals. In *ICALP (2)*, LNCS 6199, 345–356. Springer.
- Moszkowski, B. 1983. *Reasoning About Digital Circuits*. Ph.D. Dissertation, Dept. of Computer Science, Stanford University, Stanford, CA.
- Pratt-Hartmann, I. 2005. Temporal prepositions and their logic. *Artificial Intelligence* 166(1-2):1–36.
- Roeper, P. 1980. Intervals and tenses. *Journal of Philosophical Logic* 9:451–469.
- Schnoebelen, P. 2003. Oracle circuits for branching-time model checking. In *ICALP*, LNCS 2719, 790–801. Springer.
- Sipser, M. 2012. *Introduction to the Theory of Computation*. International Thomson Publishing, 3rd edition.
- Venema, Y. 1990. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic* 31(4):529–547.
- Venema, Y. 1991. A modal logic for chopping intervals. *Journal of Logic and Computation* 1(4):453–476.
- Zhou, C., and Hansen, M. R. 2004. *Duration Calculus - A Formal Approach to Real-Time Systems*. Monographs in Theoretical Computer Science. An EATCS Series. Springer.