# Checking Interval Properties of Computations

Angelo Montanari[1], Aniello Murano[2], Giuseppe Perelli[2], Adriano Peron[2]
[1]Università degli Studi di Udine, Italy,
[2]Università degli Studi di Napoli “Federico II”, Napoli, Italy
angelo.montanari@uniud.it,{aniello.murano, giuseppe.perelli, adrperon}@unina.it

***Abstract*—Model checking is a powerful method widely explored in formal verification. Given a model of a system, e.g. a Kripke structure, and a formula specifying its expected behavior, one can verify whether the system meets the behavior by checking the formula against the model. Classically, system behavior is given as a formula of a temporal logic, such as LTL and the like. These logics are “point-wise” interpreted, as they describe how the system evolves state-by-state. However, there are relevant properties, such as those involving temporal aggregations, which are inherently “interval-based”, and thus asking for an interval temporal logic. In this paper, we give a formalization of the model checking problem in an interval logic setting. First, we provide an interpretation of formulas of Halpern and Shoham’s interval temporal logic HS over Kripke structures, which allows one to check interval properties of computations. Then, we prove that the model checking problem for HS against Kripke structures is decidable by a suitable small model theorem, and we outline a PSPACE decision procedure for the meaningful fragments $\mathsf{A\overline{A}B\overline{B}}$ and $\mathsf{A\overline{A}E\overline{E}}$.**

## I. Introduction

A classical problem in system design is to come up with automatic techniques to ensure reliability. In this context, *formal methods* have provided structures and algorithms that have been successfully applied in several domains. One of the most notable techniques is *model checking*, where a formal specification of the desired properties of the system is checked against a model of its behavior [6], [7], [18], [20]. The solution of the model checking problem, and thus its precise complexity, relies on the particular computational model and specification language we consider. In finite-state system verification, systems are usually modeled as *labeled state-transition graphs*, or *Kripke structures*, while specifications are formulas of a suitable (point-based) linear or branching temporal logic. The first attempt in this direction goes back to the late ’70s, when the use of the linear temporal logic LTL in program verification was proposed by Pnueli [16]. 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 analyze, for each of them, how *proposition letters*, labeling the states, change from one state to the next one along the path. The model checking problem for LTL turns out to be PSPACE-COMPLETE [7], [17].

Propositional interval temporal logics provide an alternative setting for reasoning about time. They have been applied in a variety of computer science fields, including artificial intelligence, theoretical computer science, and databases [9]. Interval-based temporal logics take intervals as their primitive temporal entities. Such a choice gives them the ability to express temporal properties, such as durative actions, accomplishments, and temporal aggregations, which cannot be dealt with in standard (point-based) temporal logics. For instance, interval temporal logics allow one to specify the property: “p has to be true in (at least) an average number of system states in a given computation sector”.

A prominent position among interval temporal logics is occupied by Halpern and Shoham’s modal logic of time intervals (HS, for short) [10]. HS features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen’s relations [1]), apart from the equality relation. As an example, the condition: “the current interval meets an interval over which p holds” can be expressed in HS by the formula $\langle \mathsf{A} \rangle \mathsf{p}$, where $\langle \mathsf{A} \rangle$ is the (existential) HS modality for Allen relation *meet*. In [10], it has been shown that the *satisfiability* problem for HS interpreted over all relevant (classes of) linear orders is highly undecidable. Since then, a lot of work has been done on the satisfiability problem for HS fragments, which showed that undecidability rules over them [4], [12], [14]. However, meaningful exceptions exist, including the logic of temporal neighborhood and the logic of sub-intervals [2], [3], [5], [15].

Here, we focus our attention on the model checking problem for HS. The idea is to interpret HS formulas on finite Kripke structures making it possible to check the correctness of the behavior of the system with respect to meaningful interval properties. To this aim, we interpret each finite path $\rho$ of a Kripke structure $\mathcal{K}$ as an interval $I$, that is, under the homogeneity assumption [19], we define the labeling of $I$ on the basis of the labeling of the states of $\rho$. Formally, we will show that Kripke structures can be suitably mapped into interval-based structures, called *abstract interval models* (AIMs, for short), over which HS formulas can be interpreted. Since Kripke structures may have loops, AIMs have, in general, an infinite domain. In order to develop a model-checking procedure, we first prove a small model theorem showing that, given an HS formula $\varphi$ and a Kripke structure $\mathcal{K}$, there exists a finite AIM which is equivalent to the one induced by $\mathcal{K}$ with respect to the satisfiability of $\varphi$. To this end, we define an equivalence relation over sequences in $\mathcal{K}$, which is parametric in the nesting depth of Allen's modalities $\langle \mathsf{B} \rangle$ and $\langle \mathsf{E} \rangle$ in $\varphi$, and we show that the resulting quotient structure is finite. Then, we devise a PSPACE model-checking procedure for two meaningful syntactic fragments of HS, namely, $\mathsf{A\overline{A}B\overline{B}}$ and $\mathsf{A\overline{A}E\overline{E}}$, that exploits a compact representation of finite AIMs.

The model checking problem for some fragments of HS (different from the ones we consider here), extended with epistemic modalities, has been studied in [13], showing that the complexity of the problem ranges from PTIME to PSPACE. As a matter of fact, the authors restrict their attention to an alternative class of Kripke structures, where the labeling function has been defined over pairs of states: a finite path (from the unravelling of a Kripke structure) takes as labeling the one associated to the pair of its extremity states. This considerably simplifies the problem and limits the applicability of the developed model checking procedure.

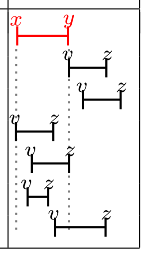
| **Relation** | **Op** | **Formal definition (w.r.t. interval structures)** | **Example** |
|---|---|---|---|
| | | |  |
| *meets* | $\langle \mathsf{A} \rangle$ | $[x,y] R_A [v,z] \Leftrightarrow y = v$ | |
| *before* | $\langle \mathsf{L} \rangle$ | $[x,y] R_L [v,z] \Leftrightarrow y < v$ | |
| *started-by* | $\langle \mathsf{B} \rangle$ | $[x,y] R_B [v,z] \Leftrightarrow x = v, z < y$ | |
| *finished-by* | $\langle \mathsf{E} \rangle$ | $[x,y] R_E [v,z] \Leftrightarrow y = z, x < v$ | |
| *contains* | $\langle \mathsf{D} \rangle$ | $[x,y] R_D [v,z] \Leftrightarrow x < v, z < y$ | |
| *overlaps* | $\langle \mathsf{O} \rangle$ | $[x,y] R_O [v,z] \Leftrightarrow x < v < y < z$ | |

Table I: Allen's interval relations and corresponding HS modalities.

The rest of the paper is organized as follows. In Section II, we introduce syntax and semantics of HS (over AIMs), and we establish a suitable connection between Kripke structures and AIMs. In Section III, we prove the small model theorem. In Section IV, we provide a suitable encoding of finite AIMs, and we show how to exploit it to develop a PSPACE model-checking procedure for two relevant fragments of HS.

## II. Interval Temporal Logic

In this section, we give syntax and semantics of HS with respect to *interval frames*. Then, we provide a mapping from Kripke structures to interval frames that allows us to interpret HS formulas over Kripke structures and thus to properly define the notion of interval-based model checking.

### A. HS *syntax and semantics*

In [1], Allen proposes an interval algebra to reason about all possible relations between pairs of (non-point) intervals in a linear order (the 6 relations in Table I and their inverses, plus the equality relation). A systematic logical study of interval reasoning started with Halpern and Shoham's work on the logic HS featuring one modality for each (non trivial) Allen's relation [10]. Existential modalities are of the form $\langle \mathsf{R} \rangle$ and $\langle \overline{\mathsf{R}} \rangle$, where $\langle \mathsf{R} \rangle \in \{\langle \mathsf{A} \rangle, \langle \mathsf{L} \rangle, \langle \mathsf{B} \rangle, \langle \mathsf{E} \rangle, \langle \mathsf{D} \rangle, \langle \mathsf{O} \rangle\}$ and $\langle \overline{\mathsf{R}} \rangle$ is the transposed modality of $\langle \mathsf{R} \rangle$. Universal modalities are simply the dual modalities.

In [10], Halpern and Shoham show that, according to the strict semantics (which excludes point-intervals), all HS modalities are definable in the fragment featuring modalities $\langle \mathsf{A} \rangle$, $\langle \mathsf{B} \rangle$, and $\langle \mathsf{E} \rangle$ and the transposed modalities $\langle \overline{\mathsf{A}} \rangle$, $\langle \overline{\mathsf{B}} \rangle$, and $\langle \overline{\mathsf{E}} \rangle$ (in case non-strict semantics is assumed, the 4 modalities $\langle \mathsf{B} \rangle$, $\langle \mathsf{E} \rangle$, $\langle \overline{\mathsf{B}} \rangle$, and $\langle \overline{\mathsf{E}} \rangle$ suffice [21]). In this paper, we assume the strict semantics (all the results can be easily adapted to non-strict semantics). The formal syntax of HS follows.

**Definition II.1** (HS syntax)**.** HS *formulas are built recursively from a set of proposition letters* $\mathcal{AP}$ *according to the following grammar (where* $\mathsf{p} \in \mathcal{AP}$*):*

$$\varphi ::= \mathsf{p} \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, \langle \mathsf{A} \rangle\varphi \,|\, \langle \overline{\mathsf{A}} \rangle\varphi \,|\, \langle \mathsf{B} \rangle\varphi \,|\, \langle \overline{\mathsf{B}} \rangle\varphi \,|\, \langle \mathsf{E} \rangle\varphi \,|\, \langle \overline{\mathsf{E}} \rangle\varphi.$$

In the sequel, if not stated otherwise, we refer to HS formulas by using the symbol $\varphi$. The other Allen's modalities can be defined as follows: $\langle \mathsf{L} \rangle\varphi = \langle \mathsf{A} \rangle\langle \mathsf{A} \rangle\varphi$, $\langle \mathsf{D} \rangle\varphi = \langle \mathsf{B} \rangle\langle \mathsf{E} \rangle\varphi = \langle \mathsf{E} \rangle\langle \mathsf{B} \rangle\varphi$, $\langle \mathsf{O} \rangle\varphi = \langle \mathsf{E} \rangle\langle \overline{\mathsf{B}} \rangle\varphi$, $\langle \overline{\mathsf{L}} \rangle\varphi = \langle \overline{\mathsf{A}} \rangle\langle \overline{\mathsf{A}} \rangle\varphi$, $\langle \overline{\mathsf{D}} \rangle\varphi = \langle \overline{\mathsf{B}} \rangle\langle \overline{\mathsf{E}} \rangle\varphi = \langle \overline{\mathsf{E}} \rangle\langle \overline{\mathsf{B}} \rangle\varphi$, $\langle \overline{\mathsf{O}} \rangle\varphi = \langle \mathsf{B} \rangle\langle \overline{\mathsf{E}} \rangle\varphi$. Let $\mathrm{AllenSet} = \{\mathsf{A}, \overline{\mathsf{A}}, \mathsf{B}, \overline{\mathsf{B}}, \mathsf{E}, \overline{\mathsf{E}}, \mathsf{L}, \overline{\mathsf{L}}, \mathsf{D}, \overline{\mathsf{D}}, \mathsf{O}, \overline{\mathsf{O}}\}$ be the set of all Allen's relations. For each existential modality $\langle \mathsf{R} \rangle$, with $\mathsf{R} \in \mathrm{AllenSet}$, the dual universal modality $[\mathsf{R}]$ is defined in the standard way: $[\mathsf{R}]\varphi = \neg\langle \mathsf{R} \rangle\neg\varphi$. Given a set of Allen's relations $\{\mathsf{R}_1, \ldots, \mathsf{R}_n\}$, we denote by $\mathsf{R}_1 \ldots \mathsf{R}_n$ the fragment of HS obtained by using only the modalities for Allen's relations $\{\mathsf{R}_1, \ldots, \mathsf{R}_n\}$. As an example, $\mathsf{A\overline{A}BE}$ is the HS fragment featuring modalities $\langle \mathsf{A} \rangle$, $\langle \overline{\mathsf{A}} \rangle$, $\langle \mathsf{B} \rangle$, and $\langle \mathsf{E} \rangle$ only.

HS can be viewed as a multi-modal logic with primitive modalities $\langle \mathsf{A} \rangle$, $\langle \mathsf{B} \rangle$, $\langle \mathsf{E} \rangle$, $\langle \overline{\mathsf{A}} \rangle$, $\langle \overline{\mathsf{B}} \rangle$, and $\langle \overline{\mathsf{E}} \rangle$. Accordingly, we can interpret HS formulas over *Multi-Modal Kripke Structure*, here called *Abstract Interval Model*, which viewed the set of intervals as an abstract set of (atomic) objects and Allen's relations as simple binary relations over this set. Such an interpretation provides a uniform, formal interpretation of HS formulas under different kinds of structures in which a concept of interval can be given. Any such structure $\mathcal{S}$ can indeed be suitably mapped into an Abstract Interval Model, making it possible to interpret HS over $\mathcal{S}$. In general, if $\mathbb{C}$ is a class of structures mappable into Abstract Interval Models, we name HS against $\mathbb{C}$ the interpretation of HS over these structures. As an example, standard interval structures can be mapped into Abstract Interval Models, thus obtaining HS against interval structures [10]. In this paper, we consider HS against Kripke Structures by providing a suitable mapping of the latter into Abstract Interval Models.

**Definition II.2** (Abstract Interval Model)**.** *Let* $\mathbb{I}$ *be a (possibly infinite) set,* $\mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}$*, and* $\mathsf{E}_{\mathbb{I}}$ *be three binary relations over* $\mathbb{I}$*, and* $\mathcal{AP}$ *be a finite set of proposition letters. An* Abstract Interval Model *(*AIM*, for short) is a tuple* $\mathcal{A} = \langle \mathcal{AP}, \mathbb{I}, \mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}, \mathsf{E}_{\mathbb{I}}, \sigma \rangle$*, where* $\sigma : \mathbb{I} \to 2^{\mathcal{AP}}$ *is a total labeling function that maps each element of* $\mathbb{I}$ *in a subset of* $\mathcal{AP}$*.*

In the interval setting, relations $\mathsf{A}_{\mathbb{I}}$, $\mathsf{B}_{\mathbb{I}}$, and $\mathsf{E}_{\mathbb{I}}$ will be interpreted as Allen's interval relations A, B, and E, respectively.

Moreover, Allen's interval relations $\overline{\mathsf{A}}$, $\overline{\mathsf{B}}$, and $\overline{\mathsf{E}}$ correspond to the inverse relations $\overline{\mathsf{A}_{\mathbb{I}}}$, $\overline{\mathsf{B}_{\mathbb{I}}}$, and $\overline{\mathsf{E}_{\mathbb{I}}}$, respectively.

**Definition II.3** (HS Semantics)**.** *Let $\mathcal{A} = \langle \mathrm{AP}, \mathbb{I}, \mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}, \mathsf{E}_{\mathbb{I}}, \sigma \rangle$ be an* AIM *and let* $\mathrm{I} \in \mathbb{I}$*. We recursively define the semantics of an* HS *formula as follows:*

- $\mathcal{A}, \mathrm{I} \models \mathsf{p}$ *iff* $\mathsf{p} \in \sigma(\mathrm{I})$*, for* $\mathsf{p} \in \mathrm{AP}$*;*
- $\mathcal{A}, \mathrm{I} \models \neg\varphi$ *iff* $\mathcal{A}, \mathrm{I} \not\models \varphi$*;*
- $\mathcal{A}, \mathrm{I} \models \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{A}, \mathrm{I} \models \varphi_1$ *and* $\mathcal{A}, \mathrm{I} \models \varphi_2$*;*
- $\mathcal{A}, \mathrm{I} \models \langle \mathsf{R} \rangle \varphi$ *iff there is* $\mathrm{J} \in \mathbb{I}$ *such that* $\mathrm{I}\ \mathsf{R}_{\mathbb{I}}\ \mathrm{J}$ *and* $\mathcal{A}, \mathrm{J} \models \varphi$*, for* $\mathsf{R}_{\mathbb{I}} \in \{\mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}, \mathsf{E}_{\mathbb{I}}\}$*;*
- $\mathcal{A}, \mathrm{I} \models \langle \overline{\mathsf{R}} \rangle \varphi$ *iff there is* $\mathrm{J} \in \mathbb{I}$ *such that* $\mathrm{J}\ \mathsf{R}_{\mathbb{I}}\ \mathrm{I}$ *and* $\mathcal{A}, \mathrm{J} \models \varphi$*, for* $\mathsf{R}_{\mathbb{I}} \in \{\mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}, \mathsf{E}_{\mathbb{I}}\}$.

### *B. From Kripke structures to Abstract Interval Models*

As we already pointed out, AIMs can be viewed as abstract interpretations of interval relations over concrete structures. Once we find a suitable mapping from a given specific structure into an AIM, we can *de facto* interpret HS formulas over the former. In this section, we introduce the notion of *Kripke structure* together with some related concepts. Then, we show how to obtain an AIM $\mathcal{A}_{\mathcal{K}}$ from a Kripke structure $\mathcal{K}$. There is no a unique way to build such an AIM. The one we propose is well suited for system verification.

**Definition II.4** (Kripke Structure)**.** *A* Kripke structure *is a tuple* $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, \delta, \mu, w_0 \rangle$*, where* $\mathrm{AP}$ *is a set of proposition letters,* $\mathrm{W}$ *is a set of states,* $\delta \subseteq \mathrm{W} \times \mathrm{W}$ *is a left-total relation over* $\mathrm{W}$*,* $\mu : \mathrm{W} \to 2^{\mathrm{AP}}$ *is a labeling function, and* $w_0 \in \mathrm{W}$ *is the initial state.*

A *track* $\rho$ over $\mathcal{K}$ is a *finite* sequence of states $v_0..v_n$, with $n \geq 1$, such that, for all $i \in [0, n[$ , it holds that $(v_i, v_{i+1}) \in \delta$. We denote by $\mathrm{Trk}_{\mathcal{K}}$ the set of all tracks over $\mathcal{K}$; moreover, for all $w \in \mathrm{W}$, we denote by $\mathrm{Trk}_{\mathcal{K}}(w)$ the set of tracks starting from $w$. $\mathrm{Trk}_{\mathcal{K}}(w_0)$ is the set of *initial tracks*. Given a track $\rho = v_0..v_n \in \mathrm{Trk}_{\mathcal{K}}$, we denote by $|\rho| = n + 1$ the number of states in $\rho$ (length of $\rho$). We denote by $\mathsf{fst}(\rho) = v_0$ and $\mathsf{lst}(\rho) = v_n$ the *first* and *last* state of $\rho$, respectively. Moroever, for all $i \in [0, n[$ , we denote by $\rho_{\leq i} = v_0..v_i$ the proper prefix of $\rho$ up to the $i$-th element, and, for all $i \in ]0, n]$, we denote by $\rho_{\geq i} = v_i..v_n$ the proper suffix of $\rho$ from the $i$-th element. We denote by $\mathsf{Pref}(\rho)$ and $\mathsf{Suff}(\rho)$ the sets of all proper prefixes and suffixes of $\rho$, respectively. Finally, we denote by $\mathsf{state}(\rho) = \{v_0, .., v_n\}$ (resp., $\mathsf{intstate}(\rho) = \{v_1, .., v_{n-1}\}$) the set of states that occur at least once in $\rho$ (resp., in $\rho$ excluding the first and the last state). In the sequel, we use symbols $\mathcal{K}$ and $\rho$ to refer to a Kripke structure and a track in $\mathrm{Trk}_{\mathcal{K}}$, respectively.

In Figure 1, we depict a Kripke structure $K_{Sched}$ modeling a scheduler that serves in rounds three processes $\mathsf{p}_1$, $\mathsf{p}_2$, and $\mathsf{p}_3$, in such a way that, in two successive rounds, the scheduler cannot serve twice the same process. $K_{Sched}$ has seven states $\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3, \overline{\mathsf{v}_1}, \overline{\mathsf{v}_2}$, and $\overline{\mathsf{v}_3}$. The computation starts in $\mathsf{v}_0$. Depending on which process $\mathsf{p}_i$, with $1 \leq i \leq 3$, is selected by the scheduler (we assume that all processes are continuously asking for the use of the common resource by sending a *request* $r_i$, for $1 \leq i \leq 3$, to the scheduler), we move to state $\mathsf{v}_i$, which is accordingly labeled with $\mathsf{p}_i$. Then, from each state $\mathsf{v}_i$, when the process $\mathsf{p}_i$ has been served, it sends an unlock command $u_i$ to the scheduler and we move to state $\overline{\mathsf{v}_i}$. This state is labeled with $\mathsf{p}_i$ as well, as it remembers the last process served by the scheduler. From $\overline{\mathsf{v}_i}$, it is only possible to satisfy new requests from a process $\mathsf{p}_j$, with $j \neq i$. Indeed, from $\overline{\mathsf{v}_i}$ only outcoming edges labeled by $r_j$, with $j \neq i$, can be taken. Let $r_j$ be the selected request. Then, we move to the corresponding state $\mathsf{v}_j$ and keep working as before. Such a Kripke structure can be easily generalized to model a scheduler that handles $n$ processes, with $n > 3$.

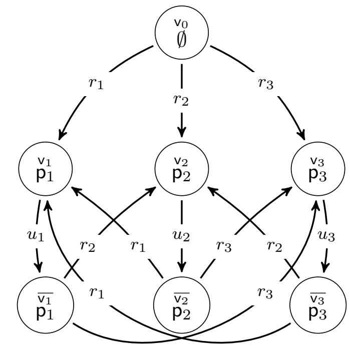



Figure 1: The Kripke structure $\mathcal{K}_{Sched}$.

If we interpret a track $\rho$ as an interval bounded by its first and last states, then a Kripke structure $\mathcal{K}$ naturally induces an abstract interval model AIM over the set $\mathrm{Trk}_{\mathcal{K}}$ of its tracks. The relations $\mathsf{A}_{\mathbb{I}}$, $\mathsf{B}_{\mathbb{I}}$, and $\mathsf{E}_{\mathbb{I}}$, corresponding to Allen's interval relations $\mathsf{A}$, $\mathsf{B}$, and $\mathsf{E}$, can be defined in terms of suitable binary relations over tracks. As for the labeling of tracks, given a track $\rho = v_0 \ldots v_n$, we set $\mathsf{p} \in \sigma(\rho)$ if and only if $\mathsf{p} \in \mu(v_0) \cap \ldots \cap \mu(v_n)$. Such a rule for track labeling conforms to the homogeneity principle in interval temporal logics, according to which a proposition letter $\mathsf{p}$ holds over an interval if and only if it holds over all its subintervals.

**Definition II.5** (Induced Abstract Interval Model)**.** *The abstract interval model induced by* $\mathcal{K}$*, called* Induced Abstract Interval Model *(*IAIM*, for short) is the* AIM $\mathcal{A}_{\mathcal{K}} = \langle \mathrm{AP}, \mathbb{I}, \mathsf{A}_{\mathbb{I}}, \mathsf{B}_{\mathbb{I}}, \mathsf{E}_{\mathbb{I}}, \sigma \rangle$*, where* $\mathrm{AP}$ *is a set of proposition letters and*

- $\mathbb{I} = \mathrm{Trk}_{\mathcal{K}}$*;*
- $\mathsf{A}_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} : \mathsf{lst}(\rho) = \mathsf{fst}(\rho')\}$*;*
- $\mathsf{B}_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} : \rho' \in \mathsf{Pref}(\rho)\}$*;*
- $\mathsf{E}_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} : \rho' \in \mathsf{Suff}(\rho)\}$*;*
- $\sigma : \mathbb{I} \to 2^{\mathrm{AP}}$ *is such that, for all* $\rho \in \mathbb{I}$*, we have that* $\sigma(\rho) = \bigcap_{w \in \mathsf{state}(\rho)} \mu(w)$.

The notion of satisfiability of an HS formula over a Kripke structure $\mathcal{K}$ and $\rho \in \mathrm{Trk}_{\mathcal{K}}$ can be given in terms of IAIMs.

**Definition II.6** (Satisfaction of HS formulas over Kripke structures)**.** *Let $\mathcal{K}$ be a Kripke structure, $\rho$ be a track in $\mathrm{Trk}_{\mathcal{K}}$, and $\varphi$ be an* HS *formula. We say that $\mathcal{K}$ and $\rho$ model $\varphi$, denoted by $\mathcal{K}, \rho \models \varphi$, iff it holds that $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi$.*

The model checking problem for HS against Kripke structures can thus be formalized as follows.

**Definition II.7** (Model checking)**.** *We say that $\mathcal{K}$ models $\varphi$, in symbols $\mathcal{K} \models \varphi$, iff, for all initial tracks $\rho \in \mathrm{Trk}_{\mathcal{K}}(w_0)$, it holds that $\mathcal{K}, \rho \models \varphi$.*

We conclude the section by specifying some meaningful properties that can be checked over $K_{Sched}$. In the following formulas, we use $wit(\{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}) \geq 2$ as a shorthand for $(\langle \mathsf{D} \rangle \mathsf{p}_1 \wedge \langle \mathsf{D} \rangle \mathsf{p}_2) \vee (\langle \mathsf{D} \rangle \mathsf{p}_1 \wedge \langle \mathsf{D} \rangle \mathsf{p}_3) \vee (\langle \mathsf{D} \rangle \mathsf{p}_2 \wedge \langle \mathsf{D} \rangle \mathsf{p}_3)$. Intuitively, it states that at least two proposition letters among $\mathsf{p}_1$, $\mathsf{p}_2$, and $\mathsf{p}_3$ occur in some state of the track. A similar interpretation can be given to shorthands of the form $wit(\{\mathsf{p}_1, \ldots, \mathsf{p}_n\}) \geq k$, for any $n$ and $k$. We also use $[G]$ as a shorthand for the (derived) universal modality and $\langle \mathsf{B} \rangle^k$ for $k$ nested applications of modality $\langle \mathsf{B} \rangle$. We start with the formula $\varphi_1 = [G](\langle \mathsf{B} \rangle^5 \top \rightarrow wit(\{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}) \geq 2)$ stating that over every interval of length greater than or equal to 7 at least two atomic propositions among $\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3$ are witnessed. Moreover, the formula $\varphi_2 = [G](\langle \mathsf{B} \rangle^k \top \rightarrow wit(\{\mathsf{p}_i\}) = 1)$ prevents the scheduler from delaying the execution of a specific process $\mathsf{p}_i$, with $i \in \{1, 2, 3\}$, too much (depending on $k$). Finally, the formula $\varphi_3 = [G](\langle \mathsf{B} \rangle^9 \top \rightarrow wit(\{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}) = 3)$ forces the scheduler to execute the three process in a strictly periodic manner. It can be easily checked that $K_{Sched}$ models $\varphi_1$, but it violates both $\varphi_2$ and $\varphi_3$.

## III. Finite model property of HS against Kripke structures

In the previous section, we have shown that, for a given Kripke structure $\mathcal{K}$, one can define a corresponding IAIM $\mathcal{A}_{\mathcal{K}}$, featuring one interval for each track of $\mathcal{K}$. Since $\mathcal{K}$ may have loops, the number of its tracks, and thus the number of intervals of $\mathcal{A}_{\mathcal{K}}$, is, in general, infinite. In this section, we prove that, given a Kripke structure $\mathcal{K}$ and an HS formula $\varphi$, there exists a finite AIM that is equivalent to the IAIM $\mathcal{A}_{\mathcal{K}}$ with respect to the satisfiability of $\varphi$.

### *A.* $\mathsf{B}_k$*- and* $\mathsf{E}_k$*-descriptors for tracks*

In order to define a suitable notion of track equivalence, we need to preliminarily show how HS formulas can be used to distinguish tracks. Consider, for instance, the HS formula $\langle \mathsf{B} \rangle^k \top$. Such a formula allows one to distinguish pairs of tracks $\rho$ and $\rho'$ such that $|\rho| < k+2$ and $|\rho'| \geq k+2$, as $\langle \mathsf{B} \rangle^k \top$ is satisfied by all and only those tracks whose length is at least $k+2$. As another example, let us consider the Kripke structure $\mathcal{K}_{Equiv}$ in Figure 2 and the tracks $v_0 v_1 v_0 v_1$ and $v_0 v_1 v_0$. It can be easily checked that formula $\langle \mathsf{A} \rangle \mathsf{q}$ is satisfied by the former track, but not by the latter one. Similarly, given the tracks $v_0 v_1 v_0 v_1$ and $v_1 v_0 v_1$, it holds that formula $\langle \overline{\mathsf{A}} \rangle \mathsf{p}$ is satisfied by the former track, but not by the latter one. In general, modalities $\langle \mathsf{A} \rangle$ and $\langle \overline{\mathsf{A}} \rangle$ can be used to distinguish between tracks that start or end at different states. Modalities $\langle \mathsf{B} \rangle$ and $\langle \mathsf{E} \rangle$ allow one to distinguish between tracks encompassing a different number of iterations of a given loop. For instance, consider the two tracks $v_1(v_0 v_1)^3$ and $v_1(v_0 v_1)^2$ consisting of three and two iterations of the loop $v_1 v_0 v_1$, respectively. It can be easily checked that the former track satisfies the formula $\langle \mathsf{B} \rangle(\langle \mathsf{A} \rangle \mathsf{p} \wedge \langle \mathsf{B} \rangle(\langle \mathsf{A} \rangle \mathsf{p} \wedge \langle \mathsf{B} \rangle \langle \mathsf{A} \rangle \mathsf{p}))$, while the latter one does not satisfy it. In general, to distinguish between tracks that differ in their length or in the number of loop iterations, formulas can exploit the nesting degree of occurrences of modality $\langle \mathsf{B} \rangle$ (or, equivalently, of modality $\langle \mathsf{E} \rangle$). To make such an intuition more precise, we introduce the notion of $\mathsf{BE}$-nesting depth.

**Definition III.1** ($\mathsf{BE}$-nesting depth)**.** *The* $\mathsf{BE}$*-nesting depth of $\varphi$, denoted by $\mathsf{Nest}_{\mathsf{BE}}(\varphi)$, is recursively defined as follows:*
$\mathsf{Nest}_{\mathsf{BE}}(\mathsf{p}) = 0$, *for* $\mathsf{p} \in \mathcal{AP}$*;* $\mathsf{Nest}_{\mathsf{BE}}(\neg\varphi) = \mathsf{Nest}_{\mathsf{BE}}(\varphi)$*;*
$\mathsf{Nest}_{\mathsf{BE}}(\varphi_1 \wedge \varphi_2) = \max\{\mathsf{Nest}_{\mathsf{BE}}(\varphi_1), \mathsf{Nest}_{\mathsf{BE}}(\varphi_2)\}$*;*
$\mathsf{Nest}_{\mathsf{BE}}(\langle \mathsf{R} \rangle \varphi) = \mathsf{Nest}_{\mathsf{BE}}(\varphi)$, *for* $\mathsf{R} \in \{\mathsf{A}, \overline{\mathsf{A}}, \overline{\mathsf{B}}, \overline{\mathsf{E}}\}$*;*
$\mathsf{Nest}_{\mathsf{BE}}(\langle \mathsf{R} \rangle \varphi) = \mathsf{Nest}_{\mathsf{BE}}(\varphi) + 1$, *for* $\mathsf{R} \in \{\mathsf{B}, \mathsf{E}\}$.

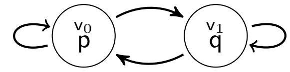



Figure 2: The Kripke structure $\mathcal{K}_{Equiv}$.

In order to define a suitable equivalence relation over tracks, we need additional information. Let us consider, for instance, the tracks ${v_0}^3 v_1 v_0$ and $v_0 v_1 {v_0}^3$ of $\mathcal{K}_{equiv}$ in Figure 2. Both tracks involve the same loop, with the same number of iterations, but they differ in the order of loop occurrence. Such a difference can be detected by the formula $\langle \mathsf{B} \rangle(\langle \mathsf{A} \rangle \mathsf{q} \wedge \langle \mathsf{B} \rangle(\langle \mathsf{A} \rangle \mathsf{p} \wedge \langle \mathsf{B} \rangle \langle \mathsf{A} \rangle \mathsf{p}))$, which is satisfied by the former track, but not by the latter one.

In the following, we distill the essential characteristics of tracks that play a key role in satisfiability of HS formulas. They will be at at the basis of the definition of the equivalence relation over tracks. The idea is that two tracks can be considered equivalent (for a given nesting depth $k$) if they both conform to a pair of descriptors $(\mathcal{D}_{\mathsf{B}}, \mathcal{D}_{\mathsf{E}})$, accounting for the $\mathsf{B}$ and the $\mathsf{E}$ relations, respectively. In order to handle satisfiability of proposition letters, $(\mathcal{D}_{\mathsf{B}}, \mathcal{D}_{\mathsf{E}})$ keeps information about the states occurring in tracks; moreover,

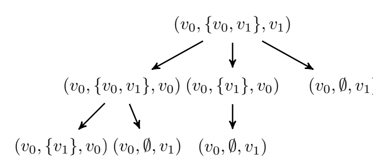



Figure 3: The $\mathsf{B}_2$-descriptor for $\rho$.

to deal with satisfiability of formulas of the forms $\langle \mathsf{A} \rangle \varphi$ and $\langle \overline{\mathsf{A}} \rangle \varphi$, it maintains information about starting and ending states of the track. More precisely, a *descriptor element* associated with a track $\rho$ is a triple $(v_{in}, \mathrm{S}, v_{fin})$ consisting of its starting state, the set of its internal (proper) states, and its ending state. In addition, to manage satisfiability of formulas of the forms $\langle \mathsf{B} \rangle \varphi$ (resp., $\langle \mathsf{E} \rangle \varphi$), $\mathcal{D}_\mathsf{B}$ (resp., $\mathcal{D}_\mathsf{E}$) maintains information about all possible descriptor elements associated with prefixes (resp., suffixes) of $\rho$. No additional information is needed to deal with modalities $\langle \overline{\mathsf{B}} \rangle$ and $\langle \overline{\mathsf{E}} \rangle$.

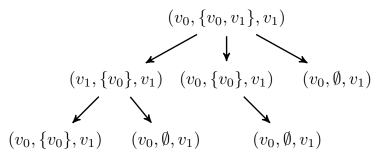


Figure 4: The $\mathsf{E}_2$-descriptor for $\rho$.

Such a construction is then repeatedly applied to prefixes of prefixes (resp., suffixes of suffixes), thus obtaining a pair of trees of depth $k$ whose nodes are labeled with descriptor elements. As an example, consider the track $\rho = v_0 v_1 v_0 v_0 v_1$ of $\mathcal{K}_{equiv}$ and assume nesting depth $k = 2$. The pair $(\mathcal{D}_\mathsf{B}, \mathcal{D}_\mathsf{E})$ of descriptors for $\rho$ is the couple of labeled trees in Figure 3 and Figure 4, respectively.

**Definition III.2** (B- and E-descriptors)**.** *An* R-descriptor*, with* $\mathsf{R} \in \{\mathsf{B}, \mathsf{E}\}$*, is a labeled tree* $\mathcal{D} = \langle \mathrm{V}, E, \lambda \rangle$*, where* $\mathrm{V}$ *is the set of vertexes,* $E \subseteq \mathrm{V} \times \mathrm{V}$ *is the set of edges, and* $\lambda : \mathrm{V} \to \mathrm{W} \times 2^{\mathrm{W}} \times \mathrm{W}$ *is the labeling function, that satisfies the following conditions:*

1) *for all* $v, v', v'' \in \mathrm{V}$*, with* $(v, v'), (v, v'') \in E$*, if* $\mathsf{sub}(v')$ *is isomorphic to* $\mathsf{sub}(v'')$*, then* $v' = v''$ *(we denote by* $\mathsf{sub}(v)$ *the largest labeled subtree of* $\mathrm{V}$ *rooted in* $v$*);*
2) *for all* $v, v' \in \mathrm{V}$ *such that* $(v, v') \in E$*,* $\lambda(v) = (v_{in}, \mathrm{S}, v_{fin})$*, and* $\lambda(v') = (v'_{in}, \mathrm{S}', v'_{fin})$*, it holds that:*
   - *if* $\mathsf{R} = \mathsf{B}$*, then* $v_{in} = v'_{in}$*,* $\mathrm{S}' \subseteq \mathrm{S}$*, and* $v'_{fin} \in \mathrm{S}$*;*
   - *if* $\mathsf{R} = \mathsf{E}$*, then* $v'_{in} \in \mathrm{S}$*,* $\mathrm{S}' \subseteq \mathrm{S}$*, and* $v_{fin} = v'_{fin}$*.*

The *depth* of a B-descriptor (resp., E-descriptor) $\langle \mathrm{V}, E, \lambda \rangle$ is the depth of the tree graph $\langle \mathrm{V}, E \rangle$. A B-descriptor (resp., E-descriptor) of depth $k$ is called a $\mathsf{B}_k$*-descriptor* (resp., $\mathsf{E}_k$*-descriptor*). A $\mathsf{B}_0$-descriptor (resp., $\mathsf{E}_0$-descriptor) $\mathcal{D}$ is a descriptor consisting of the root only, denoted by $root(\mathcal{D})$. By condition 1, it holds that, for any Kripke structure $\mathcal{K}$ and any $k \in \mathbb{N}$, each $\mathsf{B}_k$-descriptor (resp., $\mathsf{E}_k$-descriptor) is a finite labeled tree (the number of children of each node is finite) of hight $k$. Moreover, the number of $\mathsf{B}_k$-descriptors (resp., $\mathsf{E}_k$-descriptors) for $\mathcal{K}$ is finite as well.

Hereafter, we consider two descriptors equal up to isomorphism. Thanks to such an equality notion, we can suitably connect B-descriptors (resp., E-descriptors) with tracks.

**Definition III.3** ($\mathsf{B}_k$- and $\mathsf{E}_k$-descriptors for tracks)**.** *The* $\mathsf{R}_k$-descriptor *for a track* $\rho$*, with* $\mathsf{R} \in \{\mathsf{B}, \mathsf{E}\}$ *and* $k \in \mathbb{N}$*, is inductively defined as follows:*

- *for* $k = 0$*, the* $\mathsf{R}_k$*-descriptor for* $\rho$ *is the* R*-descriptor* $\mathcal{D} = \langle \{root(\mathcal{D})\}, \emptyset, \lambda \rangle$*, with* $\lambda(root(\mathcal{D})) = (\mathsf{fst}(\rho), \mathsf{intstate}(\rho), \mathsf{lst}(\rho))$*;*
- *for* $k > 0$*, the* $\mathsf{R}_k$*-descriptor for* $\rho$ *is the* R*-descriptor* $\mathcal{D} = \langle \mathrm{V}, E, \lambda \rangle$*, with* $\lambda(root(\mathcal{D})) = (\mathsf{fst}(\rho), \mathsf{intstate}(\rho), \mathsf{lst}(\rho))$*, that satisfies the following properties:*
  - *if* $\mathsf{R} = \mathsf{B}$:
    1) *for each prefix* $\rho'$ *of* $\rho$*, there is* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$ *and* $\mathsf{sub}(v)$ *is the* $\mathsf{R}_{k-1}$*-descriptor for* $\rho'$*;*
    2) *for each vertex* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$*, there is a prefix* $\rho'$ *of* $\rho$ *such that* $\mathsf{sub}(v)$ *is the* $\mathsf{R}_{k-1}$*-descriptor for* $\rho'$*.*
  - *if* $\mathsf{R} = \mathsf{E}$:
    1) *for each suffix* $\rho'$ *of* $\rho$*, there is* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$ *and* $\mathsf{sub}(v)$ *is the* $\mathsf{R}_{k-1}$*-descriptor for* $\rho'$*;*
    2) *for each vertex* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$*, there is a suffix* $\rho'$ *of* $\rho$ *such that* $\mathsf{sub}(v)$ *is the* $\mathsf{R}_{k-1}$*-descriptor for* $\rho'$*.*

Notice that not all the B-descriptors are B-descriptors for some track $\rho$ in the Kripke structure. Consider, for instance, the Kripke structure $\mathcal{K}$ of Figure 5 and the $\mathsf{B}_1$-descriptor $\mathcal{D}$ having $(v_0, \{v_1, v_2\}, v_3)$ as its root and the two children $(v_0, \{v_1\}, v_2)$ and $(v_0, \{v_2\}, v_1)$. It is evident that the two tracks $\rho_1 = v_0 v_1 v_2 v_3$ and $\rho_2 = v_0 v_2 v_1 v_3$ associated with the root are not described by the $\mathsf{B}_1$-descriptor $\mathcal{D}$, since $\rho_1$ has not a prefix represented by $(v_0, \{v_2\}, v_1)$ and $\rho_2$ has not a prefix represented by $(v_0, \{v_1\}, v_2)$.

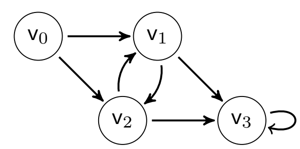


Figure 5: A Kripke structure $\mathcal{K}$.

We are now ready to introduce the notion of $k$-descriptor equivalence.

**Definition III.4.** *Let* $\mathcal{K}$ *be a Kripke structure,* $\rho, \rho'$ *be two tracks over* $\mathcal{K}$*, and* $k \in \mathbb{N}$*. We say that* $\rho$ *and* $\rho'$ *are* $k$-descriptor equivalent*, denoted by* $\rho \sim_k \rho'$*, if and only if they have the same* $\mathsf{B}_k$*- and* $\mathsf{E}_k$*-descriptors.*

It can be easily checked that the $k$-descriptor equivalence is an equivalence relation. Moreover, each equivalence class is univocally identified by a pair $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ of descriptors whose roots have the same labeling.

### *B. Quotient Induced Abstract Interval Models*

We now show how descriptors allow one to obtain a finite representation of an IAIM, called *quotient* IAIM. To start with, for any $k \in \mathbb{N}$, we denote by $k$-Desc the set of all pairs

$(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ such that there exists a track in $\mathrm{Trk}_{\mathcal{K}}$ which has $\mathcal{D}_{\mathsf{B}_k}$ and $\mathcal{D}_{\mathsf{E}_k}$ respectively as its $\mathsf{B}_k$- and $\mathsf{E}_k$-descriptor or, equivalently, the set of equivalence classes in $\mathrm{Trk}_{\mathcal{K}}$. Since $k$-Desc can be viewed as the quotient set of $\mathrm{Trk}_{\mathcal{K}}$ with respect to descriptor equivalence, we denote its elements by $[\rho]_{\sim_k}$. Allen's relations over $k$-Desc can be defined as follows.

**Definition III.5** (Allen's relations over $k$-Desc)**.** *Let* $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k}), (\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k}) \in k\text{-Desc}$*, with* $\mathcal{D}_{\mathsf{B}_k} = \langle \mathrm{V}_\mathsf{B}, E_\mathsf{B}, \lambda_\mathsf{B}\rangle$, $\mathcal{D}_{\mathsf{E}_k} = \langle \mathrm{V}_\mathsf{E}, E_\mathsf{E}, \lambda_\mathsf{E}\rangle$, $\mathcal{D}'_{\mathsf{B}_k} = \langle \mathrm{V}'_\mathsf{B}, E'_\mathsf{B}, \lambda'_\mathsf{B}\rangle$*, and* $\mathcal{D}'_{\mathsf{E}_k} = \langle \mathrm{V}'_\mathsf{E}, E'_\mathsf{E}, \lambda'_\mathsf{E}\rangle$*. We say that:*

1) $((\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k}), (\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})) \in \mathsf{A}_{\mathrm{Desc}}$ *iff* $\lambda_\mathsf{B}(root(\mathcal{D}_{\mathsf{B}_k})) = (v_{in}, \mathrm{S}, v_{fin})$, $\lambda'_\mathsf{B}(root(\mathcal{D}'_{\mathsf{B}_k})) = (v'_{in}, \mathrm{S}', v'_{fin})$*, and* $v_{fin} = v'_{in}$*;*
2) $((\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k}), (\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})) \in \mathsf{B}_{\mathrm{Desc}}$ *iff there exists* $v \in \mathrm{V}_\mathsf{B}$ *such that* $(root(\mathcal{D}_{\mathsf{B}_k}), v) \in E_\mathsf{B}$ *and* $\mathsf{sub}_{\mathcal{D}_{\mathsf{B}_k}}(v)$ *is isomorphic to the subtree of* $\mathcal{D}'_{\mathsf{B}_k}$ *obtained by removing the nodes of depth* $k$*;*
3) $((\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k}), (\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})) \in \mathsf{E}_{\mathrm{Desc}}$ *iff there exists* $v \in \mathrm{V}_\mathsf{E}$ *such that* $(root(\mathcal{D}_{\mathsf{E}_k}), v) \in E_\mathsf{E}$ *and* $\mathsf{sub}_{\mathcal{D}_{\mathsf{E}_k}}(v)$ *is isomorphic to the subtree of* $\mathcal{D}'_{\mathsf{E}_k}$ *obtained by removing the nodes of depth* $k$*.*

Intuitively, Item 1 of Definition III.5 states that Allen relation $\mathsf{A}$ holds between $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ and $(\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})$ if and only if the ending state in the root label of $\mathcal{D}_{\mathsf{B}_k}$ (and then of $\mathcal{D}_{\mathsf{E}_k}$) is equal to the initial state in the root label of $\mathcal{D}'_{\mathsf{B}_k}$ (and then of $\mathcal{D}'_{\mathsf{E}_k}$). By definition of descriptors, if $\rho$ and $\rho'$ are tracks that are represented by $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ and $(\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})$, respectively, then $\mathsf{lst}(\rho) = \mathsf{fst}(\rho')$ and thus Allen relation $\mathsf{A}$ holds between them. Item 2 of Definition III.5 states that Allen relation $\mathsf{B}$ holds between $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ and $(\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})$ if and only if there exists a subtree of $\mathcal{D}_{\mathsf{B}_k}$, rooted in a child of the root, which is isomorphic to the $\mathsf{B}_k$-descriptor $\mathcal{D}'_{\mathsf{B}_k}$ up to nodes of level $k-1$. By definition of descriptor, all tracks represented by $\mathcal{D}'_{\mathsf{B}_k}$ are prefixes of at least one track represented by $\mathcal{D}_{\mathsf{B}_k}$. Finally, Item 3 of Definition III.5 states that Allen relation $\mathsf{E}$ holds between $(\mathcal{D}_{\mathsf{B}_k}, \mathcal{D}_{\mathsf{E}_k})$ and $(\mathcal{D}'_{\mathsf{B}_k}, \mathcal{D}'_{\mathsf{E}_k})$ if and only if there exists a subtree of $\mathcal{D}_{\mathsf{E}_k}$, rooted in a child of the root, which is isomorphic to the $\mathsf{E}_k$-descriptor $\mathcal{D}'_{\mathsf{E}_k}$ up to nodes of level $k-1$. As in the case of Item 2, all tracks represented by $\mathcal{D}'_{\mathsf{E}_k}$ are suffixes of at least one track represented by $\mathcal{D}_{\mathsf{E}_k}$. A graphical account of the behavior of Allen relation $\mathsf{B}$ over descriptors (the case of $\mathsf{E}$ is analogous) is given in Figure 6, which puts in evidence the correspondence between the subtree with black vertexes in Figure 6(a) and that in Figure 6(b).

Definition III.5 can be easily generalized to pairs of descriptors belonging to $k$-Desc and $k'$-Desc, with $k \neq k'$ (in case of $\mathsf{A}_{\mathrm{Desc}}$, all possible values for $k, k'$ must be considered, while in case of $\mathsf{B}_{\mathrm{Desc}}$ and $\mathsf{E}_{\mathrm{Desc}}$, only pairs $k, k'$, with $k' < k$, must be taken into consideration).

Quotient induced abstract interval models of depth $k$ (quotient IAIM for short) are finite AIM consisting of $h$-Desc, with $h \leq k$. Let $\Omega = \bigcup_{h \leq k} h\text{-Desc}$. Quotient IAIM of depth $k$ are formally defined as follows.

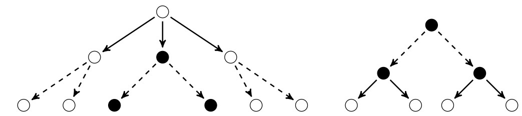



Figure 6: A graphical representation of the $\mathsf{B}$ relation between $\mathsf{B}$-descriptors.

**Definition III.6** (Quotient IAIM of depth $k$)**.** *Let* $\mathcal{K}$ *be a Kripke structure and* $k$ *be a natural number, which represents the nesting depth of an* HS *formula. The* quotient IAIM *of depth* $k$ *is the finite* AIM $\mathcal{A}/\!\sim_k = \langle \mathrm{AP}, \Omega, \mathsf{A}_{\mathrm{Desc}}, \mathsf{B}_{\mathrm{Desc}}, \mathsf{E}_{\mathrm{Desc}}, \sigma\rangle$*, where the mapping* $\sigma : \Omega \to 2^{\mathrm{AP}}$ *is defined as follows: for all* $(\mathcal{D}_\mathsf{B}, \mathcal{D}_\mathsf{E}) \in \Omega$*, with* $\lambda(root(\mathcal{D}_\mathsf{B})) = \lambda(root(\mathcal{D}_\mathsf{E})) = (v_{in}, \mathrm{S}, v_{fin})$*, it holds that* $\sigma((\mathcal{D}_\mathsf{B}, \mathcal{D}_\mathsf{E})) = \mu(v_{in}) \cap \mu(v_{fin}) \cap \bigcap_{v \in \mathrm{S}} \mu(v)$*.*

It can be shown that the finite quotient IAIM and the possibly infinite IAIM $\mathcal{A}_{\mathcal{K}}$ induced by $\mathcal{K}$ are equivalent with respect to satisfiability of HS formulas with nesting depth at most $k$, as formally stated by the following theorem.

**Theorem III.1** (Satisfiability preservation)**.** *Let* $\mathcal{K}$ *be a finite Kripke structure,* $\rho, \rho'$ *be two tracks in* $\mathrm{Trk}_{\mathcal{K}}$*,* $\mathcal{A}_{\mathcal{K}}$ *be the* IAIM *associated with* $\mathcal{K}$*, and* $\varphi$ *be an* HS *formula with* $\mathsf{Nest}_{\mathsf{BE}}(\varphi) = k$*. If* $\rho$ *and* $\rho'$ *have the same* $\mathsf{B}_k$*- and* $\mathsf{E}_k$*-descriptors, then* $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi$ *iff* $\mathcal{A}_{\mathcal{K}}, \rho' \models \varphi$*.*

Since satisfiability of HS formulas is preserved by $k$-descriptor equivalence, from Theorem III.1 it immediately follows that the model checking problem $\mathcal{K}, \rho \models \varphi$ can be reduced to the problem $\mathcal{A}/\!\sim_k, [\rho]_{\sim_k} \models \varphi$.

**Corollary III.1.** *Let* $\mathcal{K}$ *be a finite Kripke structure,* $\rho \in \mathrm{Trk}_{\mathcal{K}}$*, and* $\varphi$ *be an* HS *formula with* $\mathsf{Nest}_{\mathsf{BE}}(\varphi) \leq k$*. Then, it holds that* $\mathcal{K}, \rho \models \varphi$ *iff* $\mathcal{A}/\!\sim_k, [\rho]_{\sim_k} \models \varphi$*.*

## IV. Model checking procedures for $\mathsf{A\overline{A}B\overline{B}}$ / $\mathsf{A\overline{A}E\overline{E}}$

The decidability of the model checking problem for HS formulas against Kripke structures immediately follows from the small model property stated by Corollary III.1. Given a Kripke structure $\mathcal{K}$ of size $n$ and an HS formula $\varphi$, with $\mathsf{Nest}_{\mathsf{BE}}(\varphi) = k$, one can interpret the quotient IAIM $\mathcal{A}/\!\sim_k$ as a *multi-modal Kripke structure* and the formula $\varphi$ as a *multi-modal logic formula*. In [8], [11], it is proved that the model checking problem for multi-modal Kripke structures and formulas is solvable in PTIME with respect to both the size of the model and the size of the formula. Thus, by exploiting this result, the model checking problem for HS against Kripke structures can be dealt with by first constructing $\mathcal{A}/\!\sim_k$ and then solving the model checking problem for a multi-modal logic. Such a translation to multi-modal logic is useful to prove that the model checking problem for HS against Kripke structures is decidable; however, it is of little help in the identification of a significant upper bound to the complexity of the problem. In this section, we outline PSPACE model checking procedures for two meaningful fragments of HS that make an essential use of a compact representation of $\mathsf{B}_k$- and $\mathsf{E}_k$-descriptors.

### A. Compact $\mathsf{B}_k$- and $\mathsf{E}_k$-descriptors

As a preliminary step, we introduce and describe the compact representation of descriptors the proposed model checking procedures rely on. We focus our attention on $\mathsf{B}_k$-descriptors, as a compact representation of $\mathsf{E}_k$-descriptors can be obtained in a completely symmetric way.

Let $\mathcal{D}_{\mathsf{B}}$ be the $\mathsf{B}_k$-descriptor $\langle \mathrm{V}, E, \lambda \rangle$ for a track $\rho$ and let $(v_{in}, \mathrm{S}, v_{fin})$ be the label of its root. We show that $\mathcal{D}_{\mathsf{B}}$ can be equivalently represented by a structure, called *compact* $\mathsf{B}_k$*-descriptor*, whose size is polynomial in both the size of S and the nesting degree $k$. Compact $\mathsf{B}_k$-descriptors exploit the fact that the set of descriptor elements associated with the prefixes of a given track can be suitably arranged. Let $\Xi$ be the set $\{\lambda(v) : v \in \mathrm{V}\}$ of descriptor elements that occur as labels of elements in $\mathcal{D}_{\mathsf{B}}$. We define a transitive relation $R_t$ on $\Xi$ as follows. Let $\rho'$ and $\rho''$ be two proper prefixes of $\rho$ and let $d' = (v_{in}, \mathrm{S}', v_{fin}{}')$ and $d'' = (v_{in}, \mathrm{S}'', v_{fin}{}'')$ be the descriptor elements for $\rho'$ and $\rho''$ in $\Xi$, respectively. We say that $d' R_t d''$ if (and only if) $\mathrm{S}' \cup \{v_{fin}{}'\} \subseteq \mathrm{S}''$. It is immediate to check that $R_t$ is transitive. Moreover, it trivially holds that if $\rho'$ is proper prefix of $\rho''$, denoted by $\rho' \prec \rho''$, then $d' R_t d''$. In general, it may happen that both $d' R_t d''$ and $d'' R_t d'$ hold for some descriptor elements $d'$ and $d''$. Consider a track $\rho = \mathsf{v}_0{}^k$ for some $k \geq 5$. The corresponding set of descriptor elements $\Xi$ is equal to $\{(\mathsf{v}_0, \emptyset, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0\}, \mathsf{v}_0)\}$, and it holds that $(\mathsf{v}_0, \{\mathsf{v}_0\}, \mathsf{v}_0) R_t (\mathsf{v}_0, \{\mathsf{v}_0\}, \mathsf{v}_0)$ As another example, consider the track $\rho = \mathsf{v}_0(\mathsf{v}_1\mathsf{v}_0)^4\mathsf{v}_0$ for the Kripke structure in Figure 2, and the descriptor elements $d' = (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)$ and $d'' = (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0)$. It can be easily checked that both $d' R_t d''$ and $d'' R_t d'$. As a general rule, by definition of $R_t$, we have that, whenever both $d' R_t d''$ and $d'' R_t d'$ hold, then $S' = S''$ and both $v_{fin}{}'$ and $v_{fin}{}''$ belong to $S' (= S'')$. On the contrary, a pair of descriptor elements $d', d''$ is strictly ordered by $R_t$, that is, $d' R_t d''$ and not $d'' R_t d'$ if (and only if) one of the following two cases holds: (i) $\mathrm{S}' \subset \mathrm{S}''$ and $v_{fin}{}' \in \mathrm{S}''$ or (ii) $\mathrm{S}' = \mathrm{S}''$, $v_{fin}{}' \in \mathrm{S}'$ and $v_{fin}{}'' \notin \mathrm{S}''$. As an example, let us consider the two prefixes $\mathsf{v}_0\mathsf{v}_0\mathsf{v}_1\mathsf{v}_2$ and $\mathsf{v}_0\mathsf{v}_0\mathsf{v}_1\mathsf{v}_2\mathsf{v}_2$ (the former is a proper prefix of the latter), whose descriptor elements are respectively $d' = (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_2)$ and $d'' = (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2\}, \mathsf{v}_2)$. By condition (i), it holds that $d' R_t d''$ and not $d'' R_t d'$.

Tracks are generated by a (finite) Kripke structure. By definition, their length is finite, but they can be arbitrarily long. Given a track $\rho$, we can associate a decriptor element $d'$ with any prefix $\rho' \in \mathsf{Pref}(\rho)$, with $|\rho'| \geq 2$. Accordingly, we can associate an ordered sequence of element descriptors $\sigma$ with the ordered sequence of prefixes of $\rho$. A compact $\mathsf{B}_k$-descriptor for $\rho$ can be viewed as a compact representation of the sequence $\sigma$ and of the relationships among its elements up to depth $k$. Such a representation can be obtained by identifying and suitably encoding those (maximal) contiguous subsequences $\sigma'$ of $\sigma$ such that (i) each descriptor element occurring in $\sigma'$ occurs at least two times (in $\sigma'$), and (ii) for all pairs of distinct descriptor elements $d', d''$ occurring in $\sigma'$, there exists an occurrence of $d'$ (in $\sigma'$) that preceeds an occurrence of $d''$ (in $\sigma'$) and vice versa. In terms of $R_t$, any such subsequence $\sigma'$ can be viewed as an equivalence class, the simplest case being that of singleton classes (when $\sigma'$ consists of 2 or more consecutive occurrences of the same descriptor element). To contract the sequence $\sigma$, we replace each $\sigma'$ by the set of all and only the descriptor elements belonging to the corresponding equivalence class[1]. To build a compact $\mathsf{B}_k$-descriptor, we further need to extract from each equivalence class $\mathcal{C}$ of the resulting contracted sequence one occurrence of each descriptor element belonging to it, and to insert them immediately before $\mathcal{C}$ in the *same order* they were in the original sequence. Such a little expansion of the contracted sequence is necessary because modality $\langle \mathsf{B} \rangle$ is able to distinguish the first prefix associated with a descriptor element in $\mathcal{C}$ from the subsequent ones.

A few examples of contracted sequences of descriptor elements follow. Let $\rho$ be the track $\mathsf{v}_0(\mathsf{v}_1\mathsf{v}_0)^3\mathsf{v}_0$. The contracted sequence for $\rho$ is $\sigma = (\mathsf{v}_0, \emptyset, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), \{(\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)\}$ (notice that this is the contracted sequence for all tracks $\mathsf{v}_0(\mathsf{v}_1\mathsf{v}_0)^n\mathsf{v}_0$, with $n \geq 3$). Consider now the track $\rho' = \mathsf{v}_0(\mathsf{v}_0\mathsf{v}_1)^3\mathsf{v}_0$ The contracted sequence for $\rho'$ is $\sigma' = (\mathsf{v}_0, \emptyset, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0\}, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1), \{(\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)\} (\neq \sigma)$. Finally, let us consider the track $\rho'' = \mathsf{v}_0(\mathsf{v}_1\mathsf{v}_0)^2\mathsf{v}_1$, which is a prefix of $\rho$. The contracted sequence for $\rho''$ is $\sigma'' = (\mathsf{v}_0, \emptyset, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1) (\neq \sigma)$.

Compact $\mathsf{B}_k$-descriptors can be obtained from contracted sequences by suitably connecting a descriptor element associated with a given prefix $\rho$ to the descriptor elements associated with the prefixes of $\rho$ up to depth $k$ (see Definition IV.1 below).

Let us now formalize the notion of compact $\mathsf{B}_k$-descriptor. Consider the contracted sequence of descriptor elements introduced above. As a preliminary step, we linearize contracted sequences by replacing each set (that is, equivalence class) that occurs in them by the sequence of its elements in the same order as the order of their first occurrences. Moreover, whenever the resulting sequence features two occurrences of the same descriptor element $d$, we replace the first one by $fst(d)$. Consider again tracks $\rho, \rho'$, and $\rho''$. Once such a rewriting has been applied, we get the following sequences: $\sigma = (\mathsf{v}_0, \emptyset, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_1\}, \mathsf{v}_0), fst((\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)), fst((\mathsf{v}_0, \{\mathsf{v}_0,$

[1]Notice that different sequences may generate the same contracted sequence.

$\mathsf{v}_1\}, \mathsf{v}_0)), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0)$, $\sigma' = (\mathsf{v}_0, \emptyset, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0\}, \mathsf{v}_1), fst((\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0)), fst((\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)$, and $\sigma'' = (\mathsf{v}_0, \emptyset, \mathsf{v}_1), (\mathsf{v}_0, \{\mathsf{v}_1\}, \mathsf{v}_0), fst((\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)), fst((\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_0)), (\mathsf{v}_0, \{\mathsf{v}_0, \mathsf{v}_1\}, \mathsf{v}_1)$.

A compact $\mathsf{B}_k$-descriptor consists of a set of descriptor elements $\Xi = \{d_1, \ldots, d_n\}$ and a set of *first occurrences* $fst(\Xi) = \{fst(d_{i_1}), \ldots, fst(d_{i_m})\}$, with $\{d_{i_1}, \ldots, d_{i_m}\} \subseteq \Xi$ and $m \leq n$, the transitive relation $R_t$ on $\Xi$, and a function $Bmap$ that takes as input an element in $\Xi \cap fst(\Xi)$ and a depth level $l$, with $0 \leq l \leq k-1$, and returns the greatest (with respect to the total ordering induced by the linearized contracted sequence) descriptor element, or copy of it, associated with a prefix at depth $l$. We make use of an auxiliary function $descr$ to extract a descriptor from a copy of it, that is, we define a function $descr$ such that $descr(d) = d$, if $d$ is a descriptor element, and $descr(d) = d'$, if $d = fst(d')$. We extend $descr$ to sets $S \subseteq \Xi \cap fst(\Xi)$ in the obvious way.

**Definition IV.1.** *A compact* $\mathsf{B}_k$*-descriptor over a set of state symbols* $\mathrm{W}$ *is a tuple* $\langle DElm, d_{root}, R_t, Bmap\rangle$ *such that:*

1) $DElm$ *is a set of elements of forms* $(v_{in}, \mathrm{S}, v_{fin})$ *(*descriptor elements*) or* $fst((v_{in}, \mathrm{S}, v_{fin}))$ *(*first occurrences*), with* $v_{in}, v_{fin} \in \mathrm{W}$ *and* $\mathrm{S} \subseteq \mathrm{W}$*;*
2) $R_t$ *is a transitive relation over* $descr(DElm)$*;*
3) *for each* $d \in descr(DElm)$*,* $fst(d) \in DElm$ *iff there exist* $d' \in descr(DElm)$ *such that* $dR_td'$ *and* $d'R_td$*;*
4) *the elements of* $DElm$ *can be ordered in a sequence* $d_0 \ldots d_n$*, with* $descr(d_i) = (v_{in_i}, \mathrm{S}_i, v_{fin_i})$*, in such a way that:*
   - $descr(d_i)R_tdescr(d_{i+1})$*, for all* $0 \leq i < n$*;*
   - *for all* $d, fst(d) \in DElm$*, if* $fst(d) = d_i$ *and* $d = d_j$*, then* $i < j$*;*
   - *for all* $d_i, d_j \in descr(DElm)$ *such that* $d_iR_td_j$ *and* $d_jR_td_i$*, if* $fst(d_i) = d_h$ *and* $fst(d_j) = d_m$*, then* $i < j$ *iff* $h < k$*;*
   - $\mathrm{S}_0 = \emptyset$*;* $\mathrm{S}_{i+1} = \mathrm{S}_i \cup \{v_{fin_i}\}$*, for all* $0 \leq i < n$*;*
   - *if* $v_{fin_{i+1}} \notin \mathrm{S}_i$*, then* $d_iR_td_{i+1}$ *and not* $d_{i+1}R_td_i$*;*
   - *if* $v_{fin_{i+1}} \in \mathrm{S}_i$*, then* $d_iR_td_{i+1}$ *and* $d_{i+1}R_td_i$*;*
5) $Bmap : DElm \times \{0, .., k-1\} \to DElm$ *is such that*
   - *if* $Bmap(d_i, l) = d_j$ *and* $i < j$*, then* $d_jR_td_i$*;*
   - *otherwise (*$Bmap(d_i, l) = d_j$ *and* $j \leq i$*), for all* $j < r < i$*,* $d_iR_td_r$*.*

The compact $\mathsf{B}_k$-descriptor for a track $\rho$ can be equivalently derived from its descriptor $\mathcal{D}_{\mathsf{B}}$ as follows. For each pair of descriptor elements $d'$ and $d''$,

- $d'R_td''$ if (and only if) there exists an edge $(v, v') \in E$, with $\lambda(v) = d''$ and $\lambda(v') = d'$;
- for all $v_1, v_2, v_3, v_4 \in \mathrm{V}$ such that $\lambda(v_1) = \lambda(v_3) = d''$, $\lambda(v_2) = \lambda(v_4) = d'$, and $(v_1, v_2), (v_4, v_3) \in E$, $fst(d')$ preceeds $fst(d'')$ (in the ordering induced by the compact $\mathsf{B}_k$-descriptor) if and only if there is $v \in \mathrm{V}$ such that (i) there exists $v' \in \mathrm{V}$ such that $(v, v') \in E$ and $\lambda(v') = d'$ and (ii) there exists no $v'' \in \mathrm{V}$ such that $(v, v'') \in E$ and $\lambda(v'') = d''$.

By construction, descriptor elements in a descriptor $\mathcal{D}_{\mathsf{B}}$ range over $\Xi$. We need to identify those nodes, labeled by some $d \in \Xi$, that act as elements in $fst(\Xi)$. To this end, we introduce an auxiliary total function $Fst : \mathrm{V} \to \Xi \cup fst(\Xi)$, that replaces, whenever necessary, a descriptor element $d$ labeling a descriptor node by $fst(d)$. $Fst$ is defined as follows. Let $d = \lambda(v)$ and assume that both $dR_td'$ and $d'R_td$ hold for some $d'$ (not necessarily distinct from $d$), which labels other nodes of the descriptor. If there are no edges $(v, v') \in E$ with $\lambda(v') = d'$ for some $d'$ such that $fst(d)$ preceeds $fst(d')$ (in the ordering induced by the compact $\mathsf{B}_k$-descriptor), then $Fst(v) = fst(d)$. Otherwise, $Fst(v) = \lambda(v)$.

**Definition IV.2.** *The compact* $\mathsf{B}_k$*-descriptor for the descriptor* $\mathcal{D}_{\mathsf{B}}$*, denoted by* $cmpt(\mathcal{D}_{\mathsf{B}})$*, is the tuple* $\langle DElm, d_{root}, R_t, Bmap\rangle$*, where:*

- $DElm = Fst(\mathrm{V})$*;*
- $d_{root}$ *is the* root *element descriptor of* $\mathcal{D}_{\mathsf{B}}$*;*
- $R_t$ *is the above-defined transitive relation on* $\Xi$*;*
- *let* $d_0 \ldots d_n$ *be the total ordering obtained by pairing* $R_t$ *and the ordering of the elements of* $fst(\Xi)$*; then,* $Bmap : DElm \times \{0, \ldots, k-1\} \to DElm$ *is the map defined as follows:* $Bmap(d, i) = d_j$*, where* $j$ *is the maximum index such that* $(v, v') \in E, depth(v) = i, Fst(v) = d$*, and* $Fst(v') = d_j$*.*

It can be easily checked that Definition IV.2 fulfills the requirements of Definition IV.1.

The compact $\mathsf{B}_k$-descriptor for a descriptor $\mathcal{D}_{\mathsf{B}}$ is polynomial (precisely, quadratic) in the size of the descriptor element $d = (v_{in}, \mathrm{S}, v_{fin})$ decorating the root of $\mathcal{D}_{\mathsf{B}}$. In particular, $|\Xi \cup fst(\Xi)|$ is quadratic in $|\mathrm{S}|$, $|R_t|$ is linear in $|\Xi \cup fst(\Xi)|$, and $|Bmap|$ is linear in $|\Xi \cup fst(\Xi)|$ and $k$.

To prove the correctness of the construction, we show how a descriptor can be recovered from its compact representation.

Let $CD = \langle DElm, d_{root}, R_t, Bmap\rangle$ be a compact $\mathsf{B}_k$-descriptor, $d_0 \ldots d_n$ be the total ordering of the elements of $DElm$, and $d \in DElm$. We denote by $\mathsf{sub}_{\mathsf{B}}(CD, d)$ the compact $\mathsf{B}_{k-1}$-descriptor $\langle DElm', d, R'_t, Bmap'\rangle$, rooted in $d$, such that:

1) $DElm' = \{d_i : i \leq j \wedge d = d_j\} \cup \{d' : d'R_td\}$;
2) $R'_t$ is the restriction of $R_t$ to $descr(DElm')$;
3) $Bmap'$ is such that $DElm'(d', j) = DElm(d', j+1)$, for $d' \in DElm'$ and $0 \leq j \leq k-2$.

Given a compact $\mathsf{B}_k$-descriptor $CD$, the expanded descriptor $\mathcal{D}_{\mathsf{B}}$ for $CD$, written $expand(CD)$, is defined as follows.

**Definition IV.3** ($\mathsf{B}_k$-expanded descriptors)**.** *The* expanded $\mathsf{B}_k$-descriptor for $CD = \langle DElm, d_{root}, R_t, Bmap\rangle$*, written*

*$expand(CD)$, is the* $\mathsf{B}_k$*-descriptor* $\mathcal{D} = \langle \mathrm{V}, E, \lambda \rangle$, *with* $\lambda(root(\mathcal{D})) = descr(d_{root})$, *inductively defined as follows:*

- *for* $k = 0$, $\mathcal{D} = \langle \{root(\mathcal{D})\}, \emptyset, \lambda \rangle$;
- *for* $k > 0$, $\mathcal{D} = \langle \mathrm{V}, E, \lambda \rangle$, *where:*
  ***(i)*** *for each* $d_i \in DElm$ *such that* $d_j = Bmap(d_{root}, 0)$ *and* $i < j$ *or* $d_i R_t d_j$, *there is* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$ *and* $\mathsf{sub}(v)$ *is the expanded* $\mathsf{B}_{k-1}$*-descriptor for* $\mathsf{sub}_\mathsf{B}(CD, d_i)$*;*
  ***(ii)*** *for each vertex* $v \in \mathrm{V}$ *such that* $(root(\mathcal{D}), v) \in E$, *there is* $d_i \in DElm$ *such that* $d_j = Bmap(d_{root}, 0)$ *and* $i < j$ *or* $d_i R_t d_j$, *and* $\mathsf{sub}(v)$ *is the expanded descriptor* $\mathsf{B}_{k-1}$*-descriptor for* $\mathsf{sub}_\mathsf{B}(CD, d_i)$.

The following proposition formally states the correctness of the construction of compact $\mathsf{B}_k$-descriptors.

**Proposition IV.1.** *Given a* $\mathcal{D}_{\mathsf{B}_k}$ *for a track* $\rho$, $\mathcal{D}_{\mathsf{B}_k}$ *is isomorphic to* $expand(cmpct(\mathcal{D}_{\mathsf{B}_k}))$.

The construction of compact $\mathsf{B}_k$-descriptors can be symmetrically done for $\mathcal{D}_{\mathsf{E}_k}$ and the analogues of Definitions IV.1, IV.2, IV.3, and Proposition IV.1 can be easily stated (they are omitted here due to space limitations).

### *B. A* PSPACE *model checking algorithm*

We start by showing that given a compact $\mathsf{B}_k$-descriptor $CD$ of the form $\langle DElm, d_{root}, \preceq, Bmap \rangle$, rooted in a descriptor element $d_{root} = (v_{in}, \mathrm{S}, v_{fin})$, and a Kripke structure $\mathcal{K}$, we can check whether $CD$ *is compatible* with $\mathcal{K}$, namely, if there exists a track $\rho$ in $\mathcal{K}$ starting from $v_{in}$ and ending at $v_{fin}$ such that $CD$ is the compact representation of a descriptor $\mathcal{D}_\mathsf{B}$ for $\rho$, that is, $cmpct(\mathcal{D}_\mathsf{B})$).

**Proposition IV.2.** *Let* $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, \delta, \mu, w_0 \rangle$ *be a Kripke structure,* $CD = \langle DElm, d_{root}, \preceq, Bmap \rangle$ *be a compact* $\mathsf{B}_k$*-descriptor over the set of symbol states* $\mathrm{W}$, *and* $d_0 \dots d_n$ *be the above-defined total ordering of the elements of* $DElm$, *with* $descr(d_i) = (v_{in_i}, \mathrm{S}_i, v_{fin_i})$. *If the conditions*

1) *for all* $0 \le i < n$, *if* $v_{fin_{i+1}} \notin \mathrm{S}_i$, *then there is an edge* $(v_{fin_i}, v_{fin_{i+1}}) \in \delta$,
2) *for all* $0 \le i < n$, *if* $v_{fin_{i+1}} \in \mathrm{S}_i$, $d_i R_t d_{i+1}$, *and* $d_{i+1} R_t d_i$, *then there is a path in* $\delta$ *from* $v_{fin_i}$ *to* $v_{fin_{i+1}}$ *involving only states in* $\mathrm{S}_i$, *and*
3) *if* $d_i, \dots d_{i+v}$ *is a maximal sequence of (adjacent) descriptor elements such that* $d_{i+j} R_t d_{i+j+1}$, $d_{i+j+1} R_t d_{i+j}$, *for* $0 \le j < v$, *then there is a track* $\rho = s_0..s_m$ *in* $\mathcal{K}$ *and indexes* $j_0, .., j_v$ *such that* $\rho(j_h) = v_{fin_{i+h}}$, *with* $0 \le h \le v$, $state(\rho(0, j_0 - 1)) = S_i$, $state(\rho(0, j_0 - 2)) \subset S_i$, *the sub-track* $\rho(j_0, j_h - 1)$ *has no occurrences of* $v_{fin_{i+t}}$, *with* $0 \le h \le t \le v$,

*are satisfied, then there is a track* $\rho$ *in* $\mathcal{K}$ *such that* $\mathcal{D}_\mathsf{B}$ *is a descriptor for* $\rho$ *and* $cmpct(\mathcal{D}_\mathsf{B}) = CD$.

Proposition IV.2 gives a sufficient condition for a compact descriptor to be witnessed in a given Kripke structure. Notice also that checking requirements 1–3 can be done in time polynomial in the size of the Kripke structure.

**Algorithm 1:** Model-checking algorithm

```
ModCheck(K, s, φ):
k := Nest_B(φ);
ans := 1
CD is the first compact descriptor of depth k
while ans = 1 Or CD is the last compact descriptor do
    ans := Chk(K, CD, φ)
    CD is replaced with the next compact descriptor
return ans
```

Figure 7: The model-checking algorithm.

The operation of generating a compact descriptor $CD$ and checking whether it is witnessed by a track $\rho$ of a given Kripke structure can thus be executed in polynomial time. Moreover, the $k-1$ compact descriptors for all the possible prefixes of $\rho$ are given by $\mathsf{sub}_\mathsf{B}(CD, d)$, with $d \in DElm$ a descriptor element in $CD$. This implies that enumerating the $k-1$ compact descriptors corresponding to prefixes of the track $\rho$ takes time linear in the size of the compact descriptor $CD$. Hence, we can devise a PSPACE algorithm that solves the model checking problem for the fragment $\mathsf{A\overline{A}B\overline{B}}$ in polynomial space. In a symmetrical way, the analogue of Proposition IV.2 can be given for the compact descriptors for descriptors $\mathcal{D}_{\mathsf{E}_k}$, and a PSPACE algorithm can be obtained for the fragment $\mathsf{A\overline{A}E\overline{E}}$.

Let $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, \delta, \mu, w_0 \rangle$ be a Kripke structure and $\varphi$ be an $\mathsf{A\overline{A}B\overline{B}}$ formula. The model checking procedure depicted in Figure 7 exploits an auxiliary function $\mathrm{Chk}(\mathcal{K}, CD, \varphi)$, checking whether an input compact descriptor $CD$ satisfies $\varphi$. Assuming without loss of generality that the descriptors can be ordered exploiting a suitable encoding, the algorithm sequentially enumerates all the possible compact descriptors $CD$ witnessed by $\mathcal{K}$ having the root of the form $(v_{in}, \mathrm{S}, v_{fin})$, with $v_{in}$ an initial state of $\mathcal{K}$. The auxiliary function $\mathrm{Chk}(\mathcal{K}, CD, \varphi)$ is sequentially invoked for each of them.

The most significant cases of the auxiliary function $\mathrm{Chk}(\mathcal{K}, CD, \varphi)$ are reported in Figure 8. In the case of atomic formulas (i.e., $\varphi \in \mathrm{AP}$), it returns true (1) or false (0) according to the labeling of states in the root of $CD$. If $\varphi$ is not an atomic formula, it makes a recursive call according to the structure of $\varphi$. If $\varphi = \langle \mathsf{R} \rangle \psi$ (resp., $\varphi = [\mathsf{R}]\psi$), for $\mathsf{R} \in \{\mathsf{A}, \overline{\mathsf{A}}\}$, the function generates sequentially (exploiting the encoding order) all the possible descriptors $CD'$ that are in the relation $\mathsf{R}$ with $CD$. The function $\mathrm{Chk}(\mathcal{K}, CD', \psi)$ is then sequentially (recursively) invoked recording the maximal (resp., minimal) value returned by all the invocations. If $\varphi = \langle \mathsf{B} \rangle \psi$ (resp., $\varphi = [\mathsf{B}]\psi$), the function sequentially invokes itself over all the compact descriptors $\mathsf{sub}_\mathsf{B}(CD, d)$ with $d$ chosen among the elements of $CD$ recording the maximal (resp., minimal) value returned by all the invocations. If $\varphi = \langle \overline{\mathsf{B}} \rangle \psi$ (resp., $\varphi = [\overline{\mathsf{B}}]\psi$), the function sequentially generates all the descriptors $CD'$ such

**Algorithm 2:** Model-checking function

```
Function Chk(K, CD, φ):
if φ = ⟨A⟩ψ then
    ans := 0
    CD' is the first compact descriptor of depth k
    while ans = 0 Or CD' is the last compact
    descriptor do
        if CD' ∈ A(CD) then
            ans := Chk(K, CD', ψ)
        CD' is replaced with the next compact
        descriptor
    return ans
if φ = ⟨B⟩ψ then
    ans := 0
    d is the first element in DElm
    while ans = 0 Or d is the last element in DElm
    do
        CD' := sub_B(CD, d)
        ans := Chk(K, CD', ψ)
        d is replaced with the next element in DElm
        w.r.t. the relation ⪯
    return ans
```

Figure 8: The model-checking function.

that $CD = \mathsf{sub}_{\mathsf{B}}(CD', d_{root})$, where $d_{root}$ is the root of $CD$, and invokes itself on $CD'$ recording the maximal (resp., minimal) value returned by all the invocations.

Clearly, the number of descriptors to be kept in memory at each time is bounded by the number of recursive calls of the function, which is in turn bounded by the nesting depth of interval operators. Thus, the space complexity of the function is given by $|\varphi| \times |CD|$, which is polynomial in the size of both the Kripke structure and the formula.

## V. Conclusion

In this paper, we studied the model checking problem for Halpern and Shoham logic of time intervals HS interpreted over finite Kripke structures. Given a finite Kripke structure modeling the system of interest, we first build an Abstract Interval Model (AIM) that allows us to deal with intervals as abstract objects. Then, by proving a small model theorem for HS over a Kripke structure mapped into an AIM, we obtain a decidability result for the considered model checking problem. In such a way, we strengthen a new point of view in formal verification that shifts the attention from a classical point-wise approach to an interval-based framework. In addition, we provide a PSPACE upper-bound to the model checking problem for two meaningful fragments of HS, namely, $\mathsf{A\overline{A}B\overline{B}}$ and $\mathsf{A\overline{A}E\overline{E}}$. The proof is based on the definition of a compact representation of descriptors associated with traces of the Kripke structure, which is an important contribution by its own. We believe that this PSPACE upper bound can be lifted to the whole logic by finding the proper interplay between $\mathsf{B}$ and $\mathsf{E}$ descriptors, and we leave this as a future work.

## References


[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation*, 20(1):133–166, 2010.

[3] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic*, 161(3):289–304, 2009.

[4] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. The dark side of interval temporal logic: marking the undecidability border. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):41–83, 2014.

[5] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. What's decidable about Halpern and Shoham's interval logic? The maximal fragment $\mathsf{A\overline{B}\overline{B}\overline{L}}$. In *LICS'11*, pages 387–396. IEEE Comp. Society Press, 2011.

[6] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1981.

[7] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2002.

[8] D. Gabbay. *Many-dimensional modal logics: theory and applications*. Elsevier North Holland, 2003.

[9] V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

[10] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[11] M. Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2006.

[12] K. Lodaya. Sharpening the undecidability of interval temporal logic. In *ASIAN'00*, LNCS 1961, pages 290–298, 2000.

[13] A. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *IJCAI'13*. IJCAI/AAAI, 2013.

[14] J. Marcinkowski and J. Michaliszyn. The undecidability of the logic of subintervals. *Fundamenta Informaticae*, 131(2):217–240, 2014.

[15] A. Montanari, G. Puppis, and P. Sala. Maximal decidable fragments of Halpern and Shoham's modal logic of intervals. In *ICALP'10 (2)*, LNCS 6199, pages 345–356, 2010.

[16] A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.

[17] A. Pnueli. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science*, 13:45–60, 1981.

[18] J.P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in CESAR. In *SP'81*, LNCS 137, pages 337–351. Springer, 1981.

[19] P. Roeper. Intervals and tenses. *Journal of Philosophical Logic*, 9:451–469, 1980.

[20] M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86*, pages 332–344. IEEE Comp. Society Press, 1986.

[21] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.