

A Model Checking Procedure for Interval Temporal Logics based on Track Representatives

Alberto Molinari¹, Angelo Montanari¹, and Adriano Peron²

- 1 Department of Mathematics and Computer Science
University of Udine, Italy
molinari.alberto@gmail.com; angelo.montanari@uniud.it
- 2 Department of Electrical Engineering and Information Technology
University of Napoli Federico II, Italy
adrperon@unina.it

Abstract

Model checking is commonly recognized as one of the most effective tools for system verification. While it has been systematically investigated in the context of classical, point-based temporal logics, it is still largely unexplored in the interval logic setting. Recently, a non-elementary model checking algorithm for Halpern and Shoham’s modal logic of time intervals HS, interpreted over finite Kripke structures, has been proposed, together with a proof of the EXPSPACE-hardness of the problem. In this paper, we devise an EXPSPACE model checking procedure for two meaningful HS fragments. It exploits a suitable contraction technique that allows one to replace sufficiently long tracks of a Kripke structure by equivalent shorter ones.

1998 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases Interval Temporal Logic, Model Checking, Complexity

Digital Object Identifier 10.4230/LIPIcs.CSL.2015.193

1 Introduction

Given a formal specification of the desired properties of a system and a model of its behaviour, model checking algorithms allow one to verify the former against the latter [6]. While the model checking problem has been systematically investigated in the context of classical, point-based temporal logics, it is still largely unexplored in the interval logic setting.

Interval temporal logic (ITL) has been proposed as a more expressive formalism for temporal representation and reasoning than standard point-based one [9, 24]. On the positive side, expressiveness of ITLs makes them well suited for a number of applications in a variety of fields, including formal verification, computational linguistics, and planning, e.g., [20, 22]. On the negative side, in most cases their satisfiability problem turns out to be undecidable, and, in the few cases of decidable ITLs, the standard proof machinery, like Rabin’s theorem, is usually not applicable.

A prominent position among ITLs is occupied by Halpern and Shoham’s modal logic of time intervals (HS, for short) [9]. 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. In [9], 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 has shown that undecidability prevails over them (see [2] for an up-to-date account of undecidable fragments). However,



© Alberto Molinari, Angelo Montanari, and Adriano Peron;
licensed under Creative Commons License CC-BY

24th EACSL Annual Conference on Computer Science Logic (CSL 2015).

Editor: Stephan Kreutzer; pp. 193–210



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

meaningful exceptions exist, including the interval logic of temporal neighbourhood and the interval logic of sub-intervals [3, 4, 5, 19].

In this paper, we focus our attention on the model checking problem for ITLs. Unlike the case of satisfiability checking, little work has been done on model checking [13, 14, 16, 18] (it is worth pointing out that, in contrast to the case of point-based, linear temporal logics, there is not an easy reduction from the model checking problem to the validity/satisfiability one). In the classical formulation of the model checking problem [6], systems are usually modelled as (finite) labelled state-transition graphs (Kripke structures), and point-based temporal logics are used to analyse, for each path/track in a Kripke structure, how proposition letters labelling the states change from one state to the next one along the path. To check interval properties of computations, one needs to collect information about states into computation stretches. This amounts to interpreting each finite path of a Kripke structure as an interval, and to suitably defining its labelling on the basis of the labelling of the states that compose it.

In [13, 14], Lomuscio and Michaliszyn address the model checking problem for epistemic extensions of some HS fragments. In [13], they focus their attention on the fragment $HS[B, E, D]$ of Allen's relations *started-by*, *finished-by*, and *contains* extended with epistemic modalities. They consider a restricted form of model checking which verifies the given specification against a single (finite) initial computation interval (*not* all possible initial computation intervals), and prove that it is a PSPACE-complete problem. Moreover, they show that the problem for the purely temporal fragment of the logic is in PTIME. In [14], they show that the picture drastically changes with other HS fragments that allow one to access infinitely many tracks/intervals. In particular, they prove that the model checking problem for the fragment $HS[A, \bar{B}, L]$ of Allen's relations *meets*, *starts*, and *before*, extended with epistemic modalities, is decidable with a non-elementary upper bound.

In [16, 18], Montanari et al. outline a general characterization of the model checking problem for full HS, interpreted over finite Kripke structures (under the homogeneity assumption [23]). Their semantic assumptions differ from those made in [13], making it difficult to compare the two research contributions. In both cases, formulas of ITL are evaluated over finite paths/tracks obtained from the unravelling of a finite Kripke structure. However, in [18] a proposition letter holds over an interval (track) if and only if it holds over all its states (homogeneity principle), while in [13] truth of proposition letters is defined over pairs of states (the endpoints of tracks/intervals). In [18], the authors introduce the basic elements of the picture, namely, the interpretation of HS formulas over (abstract) interval models, the mapping of finite Kripke structures into (abstract) interval models, the notion of track descriptor, and a small model theorem proving (with a non-elementary procedure) the decidability of the model checking problem for full HS against finite Kripke structures. However, technical details of the proofs are not fully worked out and no lower bound to the complexity of the problem, that is, no hardness result, is given. In addition, they outline a PSPACE model checking procedure for two HS fragments, but it turns out to be flawed. In [16], Molinari et al. work out the model checking problem for full HS in all its details, and prove that it is EXPSPACE-hard.

In this paper, we prove that the model checking problem for two large HS fragments, namely, the fragment $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ of Allen's relations *meets*, *met-by*, *started-by*, *starts* and *finishes*, and the fragment $HS[A, \bar{A}, E, \bar{B}, \bar{E}]$ of Allen's relations *meets*, *met-by*, *finished-by*, *starts* and *finishes*, is in EXPSPACE. Moreover, we prove that it is NEXP-hard, provided that a succinct encoding of formulas is used (otherwise, we can only give an NP-hardness result).

■ **Table 1** Allen’s interval 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	$\langle O \rangle$	$[x, y] \mathcal{R}_O [v, z] \iff x < v < y < z$	

The paper is organized as follows. In Section 2 we provide some background knowledge. In Section 3 we introduce the key notion of descriptor sequence for a track of a finite Kripke structure, and we exploit it to define an indistinguishability (equivalence) relation over tracks. In Section 4 we prove a small model theorem, showing that we can select a track representative of bounded length from each equivalence class, we outline a model checking procedure, and we provide a lower bound to the complexity of the problem. Conclusions give a short assessment of the work done and describe future research directions. Due to space limitations, all proofs are omitted; they can be found in [17].

2 Background Knowledge

2.1 The interval temporal logic HS

An interval algebra to reason about intervals and their relative order was first proposed by Allen [1]; then, a systematic logical study of ITLs was done by Halpern and Shoham, who introduced the logic HS featuring one modality for each Allen’s interval relation [9], except for equality. Table 1 depicts 6 of the 13 Allen’s relations together with the corresponding HS (existential) modalities. The other 7 are equality and the 6 inverse relations (given a binary relation \mathcal{R} , the inverse relation $\bar{\mathcal{R}}$ is such that $b\bar{\mathcal{R}}a$ if and only if $a\mathcal{R}b$).

The language of HS features 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, namely, $\langle A \rangle$, $\langle L \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle D \rangle$, $\langle O \rangle$, $\langle \bar{A} \rangle$, $\langle \bar{L} \rangle$, $\langle \bar{B} \rangle$, $\langle \bar{E} \rangle$, $\langle \bar{D} \rangle$ and $\langle \bar{O} \rangle$. HS formulas are defined as follows:

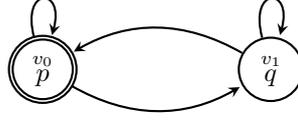
$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle\psi \mid \langle \bar{X} \rangle\psi, \quad \text{with } p \in \mathcal{AP}, X \in \{A, L, B, E, D, O\}.$$

We will make use of the standard abbreviations of propositional logic. Moreover, for all X , dual universal modalities $[X]\psi$ and $[\bar{X}]\psi$ are respectively defined as $\neg\langle X \rangle\neg\psi$ and $\neg\langle \bar{X} \rangle\neg\psi$.

We will assume the *strict semantics* of HS: only intervals made of at least two points are allowed.¹ All HS modalities can be expressed in terms of $\langle A \rangle$, $\langle B \rangle$, and $\langle E \rangle$, and the transposed modalities $\langle \bar{A} \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$ as follows: $\langle L \rangle\psi \equiv \langle A \rangle\langle A \rangle\psi$, $\langle \bar{L} \rangle\psi \equiv \langle \bar{A} \rangle\langle \bar{A} \rangle\psi$, $\langle D \rangle\psi \equiv \langle B \rangle\langle E \rangle\psi$, $\langle O \rangle\psi \equiv \langle E \rangle\langle \bar{B} \rangle\psi$, $\langle \bar{D} \rangle\psi \equiv \langle \bar{B} \rangle\langle \bar{E} \rangle\psi$, and $\langle \bar{O} \rangle\psi \equiv \langle B \rangle\langle \bar{E} \rangle\psi$.

Given any subset of Allen’s relations $\{X_1, \dots, X_n\}$, we denote by $HS[X_1, \dots, X_n]$ the fragment of HS that features modalities X_1, \dots, X_n only.

¹ HS modalities are *mutually exclusive* and *jointly exhaustive* only in the strict semantics, i.e., exactly one of them holds between any two intervals. However, the strict semantics can easily be “relaxed” to include point intervals, and all results we are going to prove hold for the non-strict semantics as well.



■ **Figure 1** The Kripke structure \mathcal{K}_{Equiv} .

HS can be viewed as a multi-modal logic with the 6 primitive modalities $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle \bar{A} \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$. Accordingly, HS semantics can be defined over a multi-modal Kripke structure, called here an *abstract interval model*, in which (strict) intervals are treated as atomic objects and Allen's relations as simple binary relations between pairs of them.

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

Intuitively, in the interval setting, \mathbb{I} is a set of intervals, $A_{\mathbb{I}}, B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as Allen's interval relations A (*meets*), B (*started-by*), and E (*finished-by*), respectively, and σ assigns to each interval 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 defined by structural induction on the formula as follows:

- (i) $\mathcal{A}, I \models p$ iff $p \in \sigma(I)$, for any proposition letter $p \in \mathcal{AP}$;
- (ii) $\mathcal{A}, I \models \neg\psi$ iff it is not true that $\mathcal{A}, I \models \psi$;
- (iii) $\mathcal{A}, I \models \psi \wedge \phi$ iff $\mathcal{A}, I \models \psi$ and $\mathcal{A}, I \models \phi$;
- (iv) $\mathcal{A}, I \models \langle X \rangle \psi$, for $X \in \{A, B, E\}$, iff there exists $J \in \mathbb{I}$ such that $I X_{\mathbb{I}} J$ and $\mathcal{A}, J \models \psi$;
- (v) $\mathcal{A}, I \models \langle \bar{X} \rangle \psi$, for $\bar{X} \in \{\bar{A}, \bar{B}, \bar{E}\}$, iff there exists $J \in \mathbb{I}$ such that $J X_{\mathbb{I}} I$ and $\mathcal{A}, J \models \psi$.

2.2 Kripke structures and abstract interval models

In this section, we define a mapping from Kripke structures to abstract interval models that makes it possible to specify properties of systems by means of HS formulas.

► **Definition 2.** A finite Kripke structure \mathcal{K} is a tuple $(\mathcal{AP}, W, \delta, \mu, w_0)$, 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 which hold at that state, while δ is the transition relation which constrains the evolution of the system over time.

Figure 1 depicts a Kripke structure, \mathcal{K}_{Equiv} , with two states (the initial state is identified by a double circle). Formally, \mathcal{K}_{Equiv} is defined by the following quintuple:

$$(\{p, q\}, \{v_0, v_1\}, \{(v_0, v_0), (v_0, v_1), (v_1, v_0), (v_1, v_1)\}, \mu, v_0),$$

where $\mu(v_0) = \{p\}$ and $\mu(v_1) = \{q\}$.

► **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 $\text{lst}(\rho) = v_n$; moreover $\rho(i, j) = v_i \cdots v_j$

is a subtrack of ρ for $0 \leq i < j \leq |\rho| - 1$. Finally, $\text{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper prefixes of ρ , and $\text{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper suffixes of ρ . Notice that the length of tracks, prefixes, and suffixes is greater than 1, as they will be mapped into strict intervals. If $\text{fst}(\rho) = w_0$, ρ is said to be an *initial track*. In the following, we will denote by $\rho \cdot \rho'$ the concatenation of the tracks ρ and ρ' , and by ρ^n the track obtained by concatenating n copies of ρ .

An abstract interval model (over $\text{Trk}_{\mathcal{X}}$) can be naturally associated with a finite Kripke structure by interpreting every track as an interval bounded by its first and last states.

► **Definition 4** ([16]). The abstract interval model induced by a finite Kripke structure $\mathcal{X} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is the abstract interval model $\mathcal{A}_{\mathcal{X}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where $\mathbb{I} = \text{Trk}_{\mathcal{X}}$, $A_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \text{lst}(\rho) = \text{fst}(\rho')\}$, $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Pref}(\rho)\}$, $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Suff}(\rho)\}$, and $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ with $\sigma(\rho) = \bigcap_{w \in \text{states}(\rho)} \mu(w)$ for all $\rho \in \mathbb{I}$.

In Definition 4, relations $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as Allen's interval relations A , B , and E , respectively. Moreover, according to the definition of σ , a proposition letter $p \in \mathcal{AP}$ holds over $\rho = v_0 \cdots v_n$ if and only if 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 if and only if it holds over all of its subintervals.

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

► **Definition 5** (Satisfiability of HS formulas over Kripke structures). Let \mathcal{X} be a finite Kripke structure, ρ be a track in $\text{Trk}_{\mathcal{X}}$, and ψ be an HS formula. We say that the pair (\mathcal{X}, ρ) satisfies ψ , denoted by $\mathcal{X}, \rho \models \psi$, if and only if it holds that $\mathcal{A}_{\mathcal{X}}, \rho \models \psi$.

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

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

Some meaningful properties of tracks that are expressible in HS can be found in [16]. For instance, the formula $[B]\perp$ can be used to select all and only the tracks of length 2. Indeed, given any ρ with $|\rho| = 2$, independently of \mathcal{X} , it holds that $\mathcal{X}, \rho \models [B]\perp$, because ρ has no (strict) prefixes. On the other hand, it holds that $\mathcal{X}, \rho \models \langle B \rangle \top$ if (and only if) $|\rho| > 2$. Let $\ell(k)$ be a shorthand for $[B]^{k-1}\perp \wedge \langle B \rangle^{k-2}\top$: it holds that $\mathcal{X}, \rho \models \ell(k)$ if and only if $|\rho| = k$.

2.3 The notion of B_k -descriptor

For any finite Kripke structure \mathcal{X} , one can find a corresponding induced abstract interval model $\mathcal{A}_{\mathcal{X}}$, featuring one interval for each track of \mathcal{X} . Since \mathcal{X} has loops (each state must have at least one successor), the number of its tracks, and thus the number of intervals of $\mathcal{A}_{\mathcal{X}}$, is infinite. In [16], given a finite Kripke structure and an HS formula φ , the authors show how to obtain a *finite* representation for each (possibly infinite) set of tracks which are equivalent with respect to satisfiability of HS formulas of the same structural complexity as φ . By making use of such a representation, they prove that the model checking problem for (full) HS is decidable (with a non-elementary upper bound) and it is EXPSPACE-hard if a suitable encoding of HS formulas is exploited [16]. In this paper, we restrict our attention to $HS[A, \bar{A}, B, \bar{B}, E]$ (and the symmetric $HS[A, \bar{A}, E, \bar{B}, \bar{E}]$) and we provide a lower complexity model checking algorithm for it. We start with the definition of some basic notions.

► **Definition 7.** Let ψ be an $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula. The B-nesting depth of ψ , denoted by $\text{Nest}_B(\psi)$, is defined by induction on the complexity of the formula as follows:

- (i) $\text{Nest}_B(p) = 0$, for any proposition letter $p \in \mathcal{AP}$;
- (ii) $\text{Nest}_B(\neg\psi) = \text{Nest}_B(\psi)$;
- (iii) $\text{Nest}_B(\psi \wedge \phi) = \max\{\text{Nest}_B(\psi), \text{Nest}_B(\phi)\}$;
- (iv) $\text{Nest}_B(\langle B \rangle \psi) = 1 + \text{Nest}_B(\psi)$;
- (v) $\text{Nest}_B(\langle X \rangle \psi) = \text{Nest}_B(\psi)$, for $X \in \{A, \bar{A}, \bar{B}, \bar{E}\}$.

Making use of Definition 7, we can introduce a relation of k -equivalence over tracks.

► **Definition 8.** Let \mathcal{X} be a finite Kripke structure and ρ and ρ' be two tracks in $\text{Trk}_{\mathcal{X}}$. We say that ρ and ρ' are k -equivalent if and only if, for every $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula ψ with $\text{Nest}_B(\psi) = k$, $\mathcal{X}, \rho \models \psi$ if and only if $\mathcal{X}, \rho' \models \psi$.

It can be easily proved that k -equivalence propagates downwards.

► **Proposition 9.** Let \mathcal{X} be a finite Kripke structure and ρ and ρ' be two tracks in $\text{Trk}_{\mathcal{X}}$. If ρ and ρ' are k -equivalent, then they are h -equivalent, for all $0 \leq h \leq k$.

We are now ready to define the key notion of *descriptor* for a track of a Kripke structure.

► **Definition 10** ([16]). Let $\mathcal{X} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho \in \text{Trk}_{\mathcal{X}}$, and $k \in \mathbb{N}$. The B_k -descriptor for ρ is a labelled tree $\mathcal{D} = (V, E, \lambda)$ of depth k , where V is a finite set of vertices, $E \subseteq V \times V$ is a set of edges, and $\lambda : V \mapsto W \times 2^W \times W$ is a node labelling function, inductively defined as follows:

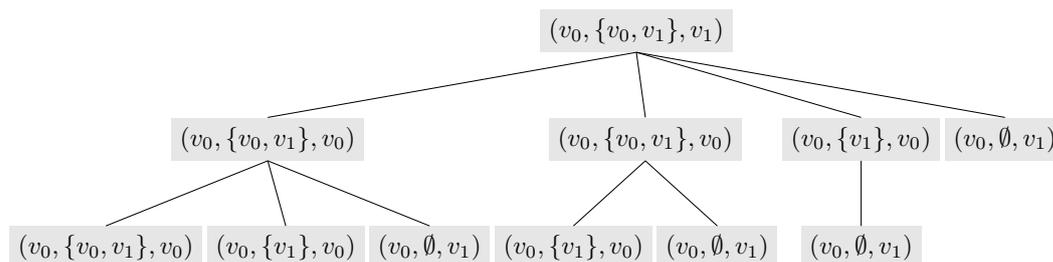
- for $k = 0$, the B_k -descriptor for ρ is the tree $\mathcal{D} = (\text{root}(\mathcal{D}), \emptyset, \lambda)$, where $\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$;
- for $k > 0$, the B_k -descriptor for ρ is the tree $\mathcal{D} = (V, E, \lambda)$, where $\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$, which satisfies the following conditions:
 1. for each prefix ρ' of ρ , there exists $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E$ and the subtree rooted in v is the B_{k-1} -descriptor for ρ' ;
 2. for each vertex $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E$, there exists a prefix ρ' of ρ such that the subtree rooted in v is the B_{k-1} -descriptor for ρ' ;
 3. for all pairs of edges $(\text{root}(\mathcal{D}), v'), (\text{root}(\mathcal{D}), v'') \in E$, if the subtree rooted in v' is isomorphic to the subtree rooted in v'' , then $v' = v''$ (here and in the following, we write subtree for maximal subtree).

Condition 3 of Definition 10 simply states that no two subtrees whose roots are siblings can be isomorphic. A B_0 -descriptor \mathcal{D} for a track consists of its root only, which is denoted by $\text{root}(\mathcal{D})$. A label of a node will be referred to as a *descriptor element*.

Basically, for any $k \geq 0$, the label of the root of the B_k -descriptor \mathcal{D} for ρ is the triple $(\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$. Each prefix ρ' of ρ is associated with some subtree whose root is labelled with $(\text{fst}(\rho'), \text{intstates}(\rho'), \text{lst}(\rho'))$ and is a child of the root of \mathcal{D} . Such a construction is then iteratively applied to the children of the root until either depth k is reached or a track of length 2 is being considered on a node.

Hereafter, two descriptors will be considered *equal up to isomorphism*.

As an example, in Figure 2 we show the B_2 -descriptor for the track $\rho = v_0v_1v_0v_0v_0v_0v_1$ of \mathcal{X}_{Equiv} (Figure 1). It is worth noticing that there exist two distinct prefixes of ρ , that is, the tracks $\rho' = v_0v_1v_0v_0v_0v_0$ and $\rho'' = v_0v_1v_0v_0v_0$, which have the same B_1 -descriptor. Since, according to Definition 10, no tree can occur more than once as a subtree of the same node (in this example, the root), in the B_2 -descriptor for ρ prefixes ρ' and ρ'' are represented



■ **Figure 2** The B_2 -descriptor for the track $v_0v_1v_0v_0v_0v_0v_1$ of \mathcal{K}_{Equiv} .

by the same tree (the first subtree of the root on the left). In general, it holds that the root of a descriptor for a track with h proper prefixes does not necessarily have h children.

In general, B -descriptors do not convey enough information to determine which track they were built from; however, they can be exploited to determine which $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formulas are satisfied by the track from which they were built.

In [16], the authors prove that, for a finite Kripke structure \mathcal{K} , there is a *finite number* (non-elementary w.r.t. $|W|$ and k) of possible B_k -descriptors; moreover the number of nodes of a descriptor has a non-elementary upper bound as well. Since the number of tracks of \mathcal{K} is infinite, and for any $k \in \mathbb{N}$ the set of B_k -descriptors for its tracks is finite, at least one B_k -descriptor must be the B_k -descriptor of *infinitely many* tracks; thus B_k -descriptors naturally induce an equivalence relation of finite index over the set of tracks of a finite Kripke structure (*k-descriptor equivalence relation*).

► **Definition 11.** Let \mathcal{K} be a finite Kripke structure, $\rho, \rho' \in \text{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. We say that ρ and ρ' are k -descriptor equivalent ($\rho \sim_k \rho'$) iff the B_k -descriptors for ρ and ρ' coincide.

The following lemma holds.

► **Lemma 12.** Let $k \in \mathbb{N}$, $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure and $\rho_1, \rho'_1, \rho_2, \rho'_2$ be tracks in $\text{Trk}_{\mathcal{K}}$ such that $(\text{lst}(\rho_1), \text{fst}(\rho'_1)) \in \delta$, $(\text{lst}(\rho_2), \text{fst}(\rho'_2)) \in \delta$, $\rho_1 \sim_k \rho_2$ and $\rho'_1 \sim_k \rho'_2$. Then $\rho_1 \cdot \rho'_1 \sim_k \rho_2 \cdot \rho'_2$.

The next proposition immediately follows from Lemma 12.

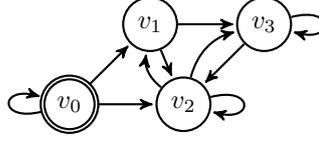
► **Proposition 13 (Left and right extensions).** Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, ρ, ρ' be two tracks in $\text{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$, and $\bar{\rho} \in \text{Trk}_{\mathcal{K}}$. If $(\text{lst}(\rho), \text{fst}(\bar{\rho})) \in \delta$, then $\rho \cdot \bar{\rho} \sim_k \rho' \cdot \bar{\rho}$, and if $(\text{lst}(\bar{\rho}), \text{fst}(\rho)) \in \delta$, then $\bar{\rho} \cdot \rho \sim_k \bar{\rho} \cdot \rho'$.

The next theorem proves that, for any pair of tracks $\rho, \rho' \in \text{Trk}_{\mathcal{K}}$, if $\rho \sim_k \rho'$, then ρ and ρ' are k -equivalent (see Definition 8). Since the set of B_k -descriptors for the tracks of a finite Kripke structure \mathcal{K} is finite (or, in other words, the equivalence relation \sim_k has a finite index), there always exists a finite number of B_k -descriptors that “satisfy” an $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula ψ with $\text{Nest}_B(\psi) = k$ (this can be formally proved by a quotient construction [16]).

► **Theorem 14 ([16]).** Let \mathcal{K} be a finite Kripke structure, ρ and ρ' be two tracks in $\text{Trk}_{\mathcal{K}}$, $\mathcal{A}_{\mathcal{K}}$ be the abstract interval model induced by \mathcal{K} , and ψ be a formula of $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ with $\text{Nest}_B(\psi) = k$. If $\rho \sim_k \rho'$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \psi \iff \mathcal{A}_{\mathcal{K}}, \rho' \models \psi$.

3 Clusters and descriptor element indistinguishability

A B_k -descriptor provides a finite encoding for a possibly infinite set of tracks (the tracks associated with that descriptor). Unfortunately, the representation of B_k -descriptors as trees



■ **Figure 3** An example of finite Kripke structure.

labelled over descriptor elements is highly redundant. For example, given any pair of subtrees rooted in some children of the root of a descriptor, it is always the case that one of them is a subtree of the other: the two subtrees are associated with two (different) prefixes of a track and one of them is necessarily a prefix of the other. In practice, the size of the tree representation of B_k -descriptors prevents their direct use in model checking algorithms, and makes it difficult to determine the intrinsic complexity of B_k -descriptors.

In this section, we devise a more compact representation of B_k -descriptors. Each class of the k -descriptor equivalence relation is a set of k -equivalent tracks. For every such class, we select a track representative whose length is (exponentially) bounded in both the size of W (the set of states of the Kripke structure) and k . In order to set such a bound, we consider suitable ordered sequences (possibly with repetitions) of descriptor elements of a B_k -descriptor. Let us define the *descriptor sequence* for a track as the ordered sequence of descriptor elements associated with its prefixes. In a descriptor sequence, descriptor elements can obviously be repeated: we devise a criterion to avoid such repetitions whenever they cannot be distinguished by any $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula of B -nesting depth up to k .

► **Definition 15.** Let $\rho = v_0v_1 \dots v_n$ be a track of a finite Kripke structure. The descriptor sequence ρ_{ds} for ρ is $d_0 \dots d_{n-1}$, where $d_i = \rho_{ds}(i) = (v_0, \text{intstates}(v_0 \dots v_{i+1}), v_{i+1})$, for $i \in \{0, \dots, n-1\}$. We denote the set of descriptor elements occurring in ρ_{ds} by $DElm(\rho_{ds})$.

For example, let us consider the finite Kripke structure of Figure 3 and the track $\rho = v_0v_0v_0v_1v_2v_1v_2v_3v_3v_2v_2$. The descriptor sequence for ρ is:

$$\rho_{ds} = (v_0, \emptyset, v_0) \left[(v_0, \{v_0\}, v_0) \right] (v_0, \{v_0\}, v_1) (v_0, \{v_0, v_1\}, v_2) \\ \left[(v_0, \Gamma, v_1) (v_0, \Gamma, v_2) \right] (v_0, \Gamma, v_3) \left[(v_0, \Delta, v_3) (v_0, \Delta, v_2) (v_0, \Delta, v_2) \right], \quad (*)$$

where $\Gamma = \{v_0, v_1, v_2\}$, $\Delta = \{v_0, v_1, v_2, v_3\}$. $DElm(\rho_{ds})$ is the set $\{(v_0, \emptyset, v_0), (v_0, \{v_0\}, v_0), (v_0, \{v_0\}, v_1), (v_0, \{v_0, v_1\}, v_2), (v_0, \Gamma, v_1), (v_0, \Gamma, v_2), (v_0, \Gamma, v_3), (v_0, \Delta, v_2), (v_0, \Delta, v_3)\}$.

To express the relationships between descriptor elements occurring in a descriptor sequence, we introduce a binary relation, R_t . Intuitively, given two descriptor elements d' and d'' of a descriptor sequence, the relation $d' R_t d''$ holds if d' and d'' are the descriptor elements of two tracks ρ' and ρ'' , respectively, and ρ' is a prefix of ρ'' .

► **Definition 16.** Let ρ_{ds} be the descriptor sequence for a track ρ and let $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v_{in}, S'', v''_{fin})$ be two descriptor elements in ρ_{ds} . Then, $d' R_t d''$ iff $S' \cup \{v'_{fin}\} \subseteq S''$.

The relation R_t is transitive: for all descriptor elements d', d'', d''' , if $d' R_t d''$ and $d'' R_t d'''$, then $S' \cup \{v'_{fin}\} \subseteq S''$ and $S'' \cup \{v''_{fin}\} \subseteq S'''$; it follows that $S' \cup \{v'_{fin}\} \subseteq S'''$, and thus $d' R_t d'''$. R_t is neither an equivalence relation nor a quasiorder, since R_t is neither reflexive (e.g., $(v_0, \{v_0\}, v_1) \not R_t (v_0, \{v_0\}, v_1)$), nor symmetric (e.g., $(v_0, \{v_0\}, v_1) R_t (v_0, \{v_0, v_1\}, v_1)$ and $(v_0, \{v_0, v_1\}, v_1) \not R_t (v_0, \{v_0\}, v_1)$), nor antisymmetric (e.g., $(v_0, \{v_1, v_2\}, v_1) R_t (v_0, \{v_1, v_2\}, v_2)$ and $(v_0, \{v_1, v_2\}, v_2) R_t (v_0, \{v_1, v_2\}, v_1)$, but the two elements are distinct).

It can be easily shown that R_t pairs descriptor elements of increasing prefixes of a track.

► **Proposition 17.** *Let ρ_{ds} be the descriptor sequence for the track $\rho = v_0v_1 \cdots v_n$. Then, $\rho_{ds}(i) R_t \rho_{ds}(j)$ for all $0 \leq i < j < n$.*

We now introduce a distinction between two types of descriptor elements.

► **Definition 18.** A descriptor element (v_{in}, S, v_{fin}) is a Type-1 descriptor element if $v_{fin} \notin S$, while it is a Type-2 descriptor element if $v_{fin} \in S$.

It can be easily checked that a descriptor element $d = (v_{in}, S, v_{fin})$ is Type-1 if and only if R_t is not reflexive in d : (i) if $d R_t d$, then $S \cup \{v_{fin}\} \not\subseteq S$, and thus $v_{fin} \notin S$, and (ii) if $v_{fin} \notin S$, then $d R_t d$. It follows that a Type-1 descriptor element cannot occur more than once in a descriptor sequence. On the other hand, Type-2 descriptor elements may occur multiple times in a descriptor sequence, and if a descriptor element occurs more than once, then it is necessarily of Type-2.

► **Proposition 19.** *If both $d' R_t d''$ and $d'' R_t d'$ for $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v_{in}, S'', v''_{fin})$ then $v'_{fin} \in S'$, $v''_{fin} \in S''$ and $S' = S''$; thus both d' and d'' are Type-2 descriptor elements.*

We are now ready to give a general characterization of the descriptor sequence ρ_{ds} for a track ρ : ρ_{ds} is composed of some (maximal) subsequences, consisting of occurrences of Type-2 descriptor elements on which R_t is symmetric, separated by occurrences of Type-1 descriptor elements. This can be formalized by means of the notion of cluster.

► **Definition 20.** A cluster C of (Type-2) descriptor elements is a maximal set of descriptor elements $\{d_1, \dots, d_s\} \subseteq DElm(\rho_{ds})$ such that $d_i R_t d_j$ and $d_j R_t d_i$ for all $i, j \in \{1, \dots, s\}$.

Thanks to maximality, clusters are pairwise disjoint: if C and C' are distinct clusters, $d \in C$ and $d' \in C'$, either $d R_t d'$ and $d' R_t d$, or $d' R_t d$ and $d R_t d'$.

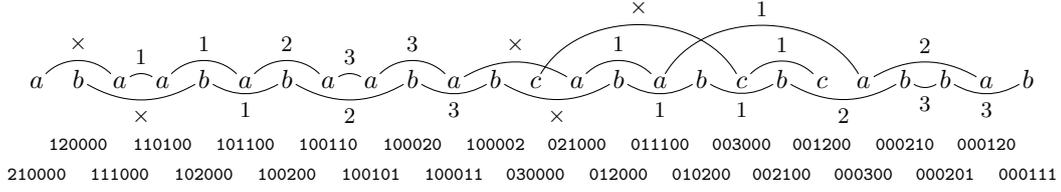
It can be easily checked that the descriptor elements of a cluster C are contiguous in ρ_{ds} (in other words, they form a subsequence of ρ_{ds}), that is, occurrences of descriptor elements of C are never shuffled with occurrences of descriptor elements not belonging to C .

► **Definition 21.** Let ρ_{ds} be a descriptor sequence and C be one of its clusters. The subsequence of ρ_{ds} associated with C is the subsequence $\rho_{ds}(i, j)$, with $i \leq j < |\rho_{ds}|$, including all and only the occurrences of the descriptor elements in C .

Notice that two subsequences associated with two distinct clusters C and C' in a descriptor sequence must be separated by at least one occurrence of a Type-1 descriptor element. For example, with reference to the descriptor sequence (*) for $\rho = v_0v_0v_0v_1v_2v_1v_2v_3v_3v_2v_2$ of the Kripke structure in Figure 3, the subsequences associated with clusters are enclosed in boxes.

While R_t allows us to order any pair of Type-1 descriptor elements, as well as any Type-1 descriptor element with respect to a Type-2 descriptor element, it does not give any means to order Type-2 descriptor elements belonging to the same cluster. This, together with the fact that Type-2 elements may have multiple occurrences in a descriptor sequence, implies that we need to somehow limit the number of occurrences of Type-2 elements in order to give a bound on the length of track representatives of B_k -descriptors.

To this end, we introduce an equivalence relation that allows us to put together indistinguishable occurrences of the same descriptor element in a descriptor sequence, that is, to detect those occurrences which are associated with prefixes of the track with the same B_k -descriptor. The idea is that a track representative for a B_k -descriptor should not include indistinguishable occurrences of the same descriptor element.



■ **Figure 4** The track $\rho = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2v_2v_3v_2$ of the finite Kripke structure depicted in Figure 3 generates the descriptor sequence $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, \{v_1, v_2\}, v_3)abaababaababcababcabbab$, where a, b , and c stand for (v_0, \emptyset, v_1) , $(v_0, \{v_1, v_2, v_3\}, v_3)$, $(v_0, \{v_1, v_2, v_3\}, v_2)$, and $(v_0, \{v_1, v_2, v_3\}, v_1)$, respectively. Here we show the subsequence $\rho_{ds}(3, |\rho_{ds}| - 1)$ associated with the cluster $C = \{a, b, c\}$. Pairs of k -indistinguishable consecutive occurrences of descriptor elements are connected by a rounded edge labelled by k . Edges labelled by \times link occurrences which are not 1-indistinguishable. The values of all missing edges can be derived from the properties established by Proposition 24 and 26. At the bottom of the figure, for each position, we give the associated configurations: $c(3) = (2, 1, 0, 0, 0, 0)$, $c(4) = (1, 2, 0, 0, 0, 0)$, and so forth.

► **Definition 22.** Let ρ_{ds} be a descriptor sequence and $k \geq 1$. We say that two occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, of the same descriptor element d are k -indistinguishable if (and only if):

- (for $k = 1$) $DElm(\rho_{ds}(0, i - 1)) = DElm(\rho_{ds}(0, j - 1))$;
- (for $k \geq 2$) for all $i \leq \ell \leq j - 1$, there exists $0 \leq \ell' \leq i - 1$ such that $\rho_{ds}(\ell)$ and $\rho_{ds}(\ell')$ are $(k - 1)$ -indistinguishable.

From Definition 22, it follows that two indistinguishable occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$ of the same descriptor element necessarily belong to the same subsequence of ρ_{ds} associated with a cluster. In general, it is always the case that $DElm(\rho_{ds}(0, i - 1)) \subseteq DElm(\rho_{ds}(0, j - 1))$ for $i < j$; 1-indistinguishability also guarantees $DElm(\rho_{ds}(0, i - 1)) = DElm(\rho_{ds}(0, j - 1))$. From this, it easily follows that the two first occurrences of a descriptor element are not 1-indistinguishable.

Proposition 23 and 24 state some basic properties of the k -indistinguishability relation.

► **Proposition 23.** Let $k \geq 2$ and $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, be two k -indistinguishable occurrences of the same descriptor element in a descriptor sequence ρ_{ds} . Then, $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are also $(k - 1)$ -indistinguishable.

► **Proposition 24.** Let $k \geq 1$ and $\rho_{ds}(i)$ and $\rho_{ds}(m)$, with $0 \leq i < m < |\rho_{ds}|$, be two k -indistinguishable occurrences of the same descriptor element in a descriptor sequence ρ_{ds} . If $\rho_{ds}(j) = \rho_{ds}(m)$, for some $i < j < m$, then $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are k -indistinguishable.

In Figure 4, we give some examples of k -indistinguishability relations, for $k \in \{1, 2, 3\}$, for a track of the finite Kripke structure depicted in Figure 3.

The next theorem establishes a fundamental connection between k -indistinguishability of descriptor elements and k -descriptor equivalence of tracks.

► **Theorem 25.** Let ρ_{ds} be the descriptor sequence for a track ρ . Two occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, of the same descriptor element are k -indistinguishable if and only if $\rho(0, i + 1) \sim_k \rho(0, j + 1)$.

Notice that k -indistinguishability between occurrences of descriptor elements is defined only for pairs of prefixes of the same track, while the relation of k -descriptor equivalence can be applied to pairs of any tracks of a Kripke structure.

The next proposition easily follows from Theorem 25.

► **Proposition 26.** *Let $\rho_{ds}(i)$, $\rho_{ds}(j)$, and $\rho_{ds}(m)$, with $0 \leq i < j < m < |\rho_{ds}|$, be three occurrences of the same descriptor element. If both the pair $\rho_{ds}(i)$ and $\rho_{ds}(j)$ and the pair $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are k -indistinguishable, for some $k \geq 1$, then $\rho_{ds}(i)$ and $\rho_{ds}(m)$ are k -indistinguishable, as well.*

4 A model checking procedure based on track representatives

In this section, we will exploit the k -indistinguishability relation between descriptor elements in a descriptor sequence ρ_{ds} for a track ρ to possibly replace ρ by a k -descriptor equivalent, shorter track ρ' of bounded length. This allows us to find, for each B_k -descriptor \mathcal{D}_{B_k} (witnessed by a track of the considered finite Kripke structure \mathcal{X}), a *track representative* $\tilde{\rho}$ in \mathcal{X} such that (i) \mathcal{D}_{B_k} is the B_k -descriptor for $\tilde{\rho}$ and (ii) the length of $\tilde{\rho}$ is bounded. Thanks to property (ii), we can check all the track representatives of a finite Kripke structure by simply visiting its unravelling up to a bounded depth.

The notion of track representative can be explained as follows. Let ρ_{ds} be the descriptor sequence for a track ρ . If there are two occurrences of the same descriptor element $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, which are k -indistinguishable (we let $\rho = \rho(0, j+1) \cdot \bar{\rho}$, with $\bar{\rho} = \rho(j+2, |\rho| - 1)$), then we can replace ρ by the k -descriptor equivalent, shorter track $\rho(0, i+1) \cdot \bar{\rho}$: by Theorem 25, $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same B_k -descriptor and thus, by Proposition 13, $\rho = \rho(0, j+1) \cdot \bar{\rho}$ and $\rho(0, i+1) \cdot \bar{\rho}$ have the same B_k -descriptor. Moreover, since $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are occurrences of the same descriptor element, $\rho(i+1) = \rho(j+1)$ and so the track $\rho(0, i+1) \cdot \bar{\rho}$ is witnessed in the finite Kripke structure. By iteratively applying such a contraction method, we can find a track ρ' which is k -descriptor equivalent to ρ , whose descriptor sequence is devoid of k -indistinguishable occurrences of descriptor elements. A *track representative* is a track that fulfils this property.

We now show how to give a bound to the length of track representatives. We start by stating some technical properties. The next proposition provides a bound to the distance within which we observe a repeated occurrence of some descriptor element in the descriptor sequence for a track. We preliminarily observe that, for any track ρ , $|DElm(\rho_{ds})| \leq |W|^2 + 1$, where W is the set of states of the finite Kripke structure. Indeed, in the descriptor sequence, the sets of internal states of prefixes of ρ increase monotonically with respect to the “ \subseteq ” relation. As a consequence, at most $|W|$ distinct sets may occur, excluding \emptyset which can occur only in the first descriptor element. Moreover, these sets can be paired with all possible final states which are at most $|W|$.

► **Proposition 27.** *For each track ρ of \mathcal{X} , associated with a descriptor element d , there exists a track ρ' of \mathcal{X} , associated with the same descriptor element d , such that $|\rho'| \leq 2 + |W|^2$.*

Proposition 27 will be used in the unravelling Algorithm 1 as a termination criterion (referred to as *0-termination criterion*) for unravelling a finite Kripke structure when it is not necessary to observe multiple occurrences of the same descriptor element: *to get a track representative for all descriptor elements, witnessed in a finite Kripke structure with set of states W and initial state v , we can avoid considering tracks longer than $2 + |W|^2$, while exploring the unravelling of the Kripke structure from v .*

Let us now consider the problem of establishing a bound for tracks devoid of pairs of k -indistinguishable occurrences of descriptor elements. We first notice that, in a descriptor sequence ρ_{ds} for a track ρ , there are at most $|W|$ occurrences of Type-1 descriptor elements. On the contrary, Type-2 descriptor elements can occur multiple times and thus, to bound the length of ρ_{ds} , one has to constrain the *number* and the *length* of the subsequences of ρ_{ds}

associated with clusters. As for their number, it suffices to observe that they are separated by Type-1 descriptor elements, and hence at most $|W|$ of them, related to distinct clusters, can occur in a descriptor sequence.

As for their length, we can proceed as follows. First, for any cluster C , it holds that $|C| \leq |W|$ as all (Type-2) descriptor elements of C share the same set S of internal states and their final states v_{fin} must belong to S . In the following, we consider the (maximal) subsequence $\rho_{ds}(u, v)$ of ρ_{ds} associated with a specific cluster C , for some $0 \leq u \leq v \leq |\rho_{ds}| - 1$, and when we mention an index i , we implicitly assume that $u \leq i \leq v$, that is, i refers to a position in the subsequence. We sequentially scan such a subsequence suitably recording the multiplicity of occurrences of descriptor elements into an auxiliary structure. To detect indistinguishable occurrences of descriptor elements up to indistinguishability $s \geq 1$, we use $s + 3$ arrays $Q_{-2}(), Q_{-1}(), Q_0(), Q_1(), \dots, Q_s()$. Array elements are sets of descriptor elements of C . Given an index i , the sets at position i , $Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \dots, Q_s(i)$, store information about indistinguishability for multiple occurrences of descriptor elements in the subsequence up to position $i > u$. To exemplify, if the scan function finds an occurrence of the descriptor element $d \in C$ at position i , that is, $\rho_{ds}(i) = d$, we have that:

1. $Q_{-2}(i)$ contains all descriptor elements of C which have never occurred in $\rho_{ds}(u, i)$;
2. $d \in Q_{-1}(i)$ if d has never occurred in $\rho_{ds}(u, i - 1)$ and $\rho_{ds}(i) = d$, that is, $\rho_{ds}(i)$ is the first occurrence of d in $\rho_{ds}(u, i)$;
3. $d \in Q_0(i)$ if d occurs at least twice in $\rho_{ds}(u, i)$ and the occurrence $\rho_{ds}(i)$ of d is *not* 1-indistinguishable from the last occurrence of d in $\rho_{ds}(u, i - 1)$;
4. $d \in Q_t(i)$ (for some $t \geq 1$) if the occurrence $\rho_{ds}(i)$ of d is t -indistinguishable, but *not* $(t + 1)$ -indistinguishable, from the last occurrence of d in $\rho_{ds}(u, i - 1)$.

In particular, at position u (the first of the subsequence), $Q_{-1}(u)$ contains only the descriptor element $d = \rho_{ds}(u)$, $Q_{-2}(u)$ is the set $C \setminus \{d\}$, and $Q_0(u), Q_1(u), \dots$ are empty sets.

In general, arrays $Q_{-2}(), Q_{-1}(), Q_0(), Q_1(), \dots, Q_s()$ satisfy the following constraints: for all i , $\bigcup_{m=-2}^s Q_m(i) = C$ and, for all i and all $m \neq m'$, $Q_m(i) \cap Q_{m'}(i) = \emptyset$.

Intuitively, at every position i , $Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \dots, Q_s(i)$ describe a *state* of the scanning process of the subsequence. The change of the state produced by the transition from position $i - 1$ to position i while scanning the sequence is formally defined by the function f , reported in Figure 5, which maps the descriptor sequence ρ_{ds} and a position i to the tuple of sets $(Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \dots, Q_s(i))$.

Notice that, whenever a descriptor element $\rho_{ds}(i) = d$ is such that $d \in Q_z(i - 1)$ and $d \in Q_{z'}(i)$, with $z < z'$ (cases (a), (b) and (d) of the definition of f), all $Q_{z''}(i)$, with $z'' > z'$, are empty sets and, for all $z'' \geq z'$, all elements in $Q_{z''}(i - 1)$ belong to $Q_{z'}(i)$. Consider, for instance, the following scenario: in a subsequence of ρ_{ds} , associated with some cluster C , $\rho_{ds}(h) = \rho_{ds}(i) = d \in C$ and $\rho_{ds}(h') = \rho_{ds}(i') = d' \in C$, for some $h < h' < i < i'$ and $d \neq d'$, and there are not other occurrences of d and d' in $\rho_{ds}(h, i')$. If $\rho_{ds}(h)$ and $\rho_{ds}(i)$ are exactly z' -indistinguishable, by definition of the indistinguishability relation, $\rho_{ds}(h')$ and $\rho_{ds}(i')$ can be no more than $(z' + 1)$ -indistinguishable. Thus, if d' is in $Q_{z''}(i - 1)$, for some $z'' > z'$, we can safely “downgrade” it to $Q_{z'}(i)$, because we know that, when we meet the next occurrence of d' ($\rho_{ds}(i')$), $\rho_{ds}(h')$ and $\rho_{ds}(i')$ will be no more than $(z' + 1)$ -indistinguishable.

In the following, we will make use of an abstract characterisation of the state of the arrays at a given position i , as determined by the scan function f , called *configuration*, that only considers the cardinality of the sets of arrays. Theorem 29 states that, when a descriptor subsequence is scanned, configurations never repeat since the sequence of configurations is

$f(\rho_{ds}, u) = (C \setminus \{d\}, \{d\}, \emptyset, \dots, \emptyset)$ with $\rho_{ds}(u) = d$;

For all $i > u$: $f(\rho_{ds}, i) = (Q_{-2}(i), Q_{-1}(i), Q_0(i), \dots, Q_s(i)) =$

$$\left\{ \begin{array}{l} (Q_{-2}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=-1}^s Q_m(i-1), \emptyset, \dots, \emptyset) \text{ if } \rho_{ds}(i) \text{ is the first occurrence of } d \text{ in } \\ \rho_{ds}(u, i); \text{ (a)} \\ (Q_{-2}(i-1), Q_{-1}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=0}^s Q_m(i-1), \emptyset, \dots, \emptyset) \text{ if } \rho_{ds}(i) = d, d \in Q_{-1}(i-1), \\ \text{and } \rho_{ds}(i) \text{ is at least the second occurrence of } d \text{ in } \rho_{ds}(u, i) \text{ and it is not 1-indistinguishable} \\ \text{from the immediately preceding occurrence of } d; \text{ (b)} \\ (Q_{-2}(i-1), Q_{-1}(i-1), \{d\} \cup Q_0(i-1), Q_1(i-1) \setminus \{d\}, \dots, Q_s(i-1) \setminus \{d\}) \text{ if } \rho_{ds}(i) = d, \\ d \in \bigcup_{m=0}^s Q_m(i-1), \text{ and } \rho_{ds}(i) \text{ is at least the second occurrence of } d \text{ in } \rho_{ds}(u, i) \text{ and it is not} \\ \text{1-indistinguishable from the immediately preceding occurrence of } d; \text{ (c)} \\ (Q_{-2}(i-1) \setminus \{d\}, \dots, Q_{t-1}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=t}^s Q_m(i-1), \emptyset, \dots, \emptyset) \text{ if } \rho_{ds}(i) = d, \rho_{ds}(i) \\ \text{is } t\text{-indistinguishable (for some } t \geq 1), \text{ but not } (t+1)\text{-indistinguishable, to the immediately} \\ \text{preceding occurrence of } d, \text{ and } d \in \bigcup_{m=-2}^{t-1} Q_m(i-1); \text{ (d)} \\ (Q_{-2}(i-1), \dots, Q_{t-1}(i-1), \{d\} \cup Q_t(i-1), Q_{t+1}(i-1) \setminus \{d\}, \dots, Q_s(i-1) \setminus \{d\}) \text{ if } \rho_{ds}(i) = \\ d, \rho_{ds}(i) \text{ is } t\text{-indistinguishable (for some } t \geq 1), \text{ but not } (t+1)\text{-indistinguishable, to the} \\ \text{immediately preceding occurrence of } d, \text{ and } d \in \bigcup_{m=t}^s Q_m(i-1). \text{ (e)} \end{array} \right.$$

■ **Figure 5** Definition of the scan function f .

strictly decreasing according to the lexicographical order $>_{lex}$. This property will allow us to establish the desired bound to the length of track representatives.

► **Definition 28.** Let ρ_{ds} be the descriptor sequence for a track ρ and i be a position in the subsequence of ρ_{ds} associated with a given cluster. The *configuration at position i* , written $c(i)$, is the tuple $c(i) = (|Q_{-2}(i)|, |Q_{-1}(i)|, |Q_0(i)|, |Q_1(i)|, \dots, |Q_s(i)|)$, where $f(\rho_{ds}, i) = (Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \dots, Q_s(i))$.

An example of a configuration sequence is given in Figure 4.

► **Theorem 29.** Let ρ_{ds} be the descriptor sequence for a track ρ and $\rho_{ds}(u, v)$, for some $u < v$, be the subsequence associated with a cluster C . For all $u < i \leq v$, if $\rho_{ds}(i) = d$, then it holds that $d \in Q_t(i-1)$, $d \in Q_{t+1}(i)$, and $c(i-1) >_{lex} c(i)$, for some $t \in \{-2, -1\} \cup \mathbb{N}$.

We show now how to select all and only those tracks which do not feature any pair of k -indistinguishable occurrences of descriptor elements. To this end, we make use of a scan function f which exploits $k+3$ arrays (the value $k+3$ derives from the k of descriptor element indistinguishability, plus the three arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$). Theorem 29 guarantees that, while scanning a subsequence, configurations are never repeated. This allows us to set an upper bound to the length of a track such that, whenever exceeded, the descriptor sequence for the track features at least a pair of k -indistinguishable occurrences of a descriptor element. The bound is essentially given by the number of possible configurations for $k+3$ arrays.

By an easy combinatorial argument, we can prove the following proposition.

► **Proposition 30.** For all $n, t \in \mathbb{N}^+$, the number of distinct t -tuples of natural numbers whose sum equals n is $\varepsilon(n, t) = \binom{n+t-1}{n} = \binom{n+t-1}{t-1}$.

Proposition 30 provides two upper bounds for $\varepsilon(n, t)$: $\varepsilon(n, t) \leq (n+1)^{t-1}$ and $\varepsilon(n, t) \leq t^n$.

Since a configuration $c(i)$ of a cluster \mathcal{C} is a $(k+3)$ -tuple whose elements add up to $|\mathcal{C}|$, Proposition 30 allows us to conclude that there are at most $\varepsilon(|\mathcal{C}|, k+3) = \binom{|\mathcal{C}|+k+2}{k+2}$ distinct configurations of size $(k+3)$, whose integers add up to $|\mathcal{C}|$. Moreover, since configurations never repeat while scanning a subsequence associated with a cluster \mathcal{C} , $\varepsilon(|\mathcal{C}|, k+3)$ is an upper bound to the length of such a subsequence.

Now, for any track ρ , ρ_{ds} has at most $|W|$ subsequences associated with distinct clusters $\mathcal{C}_1, \mathcal{C}_2, \dots$, and thus if the following upper bound to the length of ρ is exceeded, then there is at least one pair of k -indistinguishable occurrences of a descriptor element in ρ_{ds} : $|\rho| \leq 1 + (|\mathcal{C}_1| + 1)^{k+2} + (|\mathcal{C}_2| + 1)^{k+2} + \dots + (|\mathcal{C}_s| + 1)^{k+2} + |W|$, where $s \leq |W|$ and the last addend is to count occurrences of Type-1 descriptor elements. Since clusters are disjoint and their union is a subset of $DElm(\rho_{ds})$, and $|DElm(\rho_{ds})| \leq 1 + |W|^2$, we get two upper bounds:

$$|\rho| \leq 1 + (|\mathcal{C}_1| + |\mathcal{C}_2| + \dots + |\mathcal{C}_s| + |W|)^{k+2} + |W| \leq 1 + (|DElm(\rho_{ds})| + |W|)^{k+2} + |W| \leq 1 + (1 + |W|^2 + |W|)^{k+2} + |W| \leq 1 + (1 + |W|)^{2k+4} + |W|,$$

and, analogously,

$$|\rho| \leq 1 + (k+3)^{|\mathcal{C}_1|} + (k+3)^{|\mathcal{C}_2|} + \dots + (k+3)^{|\mathcal{C}_s|} + |W| \leq 1 + (k+3)^{|\mathcal{C}_1| + |\mathcal{C}_2| + \dots + |\mathcal{C}_s|} + |W| \leq 1 + (k+3)^{|DElm(\rho_{ds})|} + |W| \leq 1 + (k+3)^{|W|^2+1} + |W|.$$

The upper bound for $|\rho|$ is then the least of the two given upper bounds:

$$\tau(|W|, k) = \min \{1 + (1 + |W|)^{2k+4} + |W|, 1 + (k+3)^{|W|^2+1} + |W|\}.$$

► **Theorem 31.** *Let \mathcal{X} be a finite Kripke structure and ρ be a track in $Trk_{\mathcal{X}}$. If $|\rho| > \tau(|W|, k)$, there exists another track in $Trk_{\mathcal{X}}$, whose length is less than or equal to $\tau(|W|, k)$, which has the same B_k -descriptor as ρ .*

Theorem 31 allows us to define a termination criterion to bound the depth of the unravelling of a finite Kripke structure ($(k \geq 1)$ -*termination criterion*), while searching for track representatives for witnessed B_k -descriptors: *for any $k \geq 1$, to get a track representative for every B_k -descriptor with initial state v and witnessed in a finite Kripke structure with set of states W , we can avoid taking into consideration tracks longer than $\tau(|W|, k)$ while exploring the unravelling of the structure from v .*

Algorithm 1 (the *unravelling algorithm*) explores the unravelling of the input Kripke structure \mathcal{X} to find the track representatives for all witnessed B_k -descriptors. The upper bound $\tau(|W|, k)$ on the maximum depth of the unravelling ensures the termination of the algorithm, which never returns a track ρ if there exist k -indistinguishable occurrences of a descriptor element in ρ_{ds} .

The next theorem proves soundness and completeness of Algorithm 1.

► **Theorem 32.** *Let \mathcal{X} be a finite Kripke structure, v be a state in W , and $k \in \mathbb{N}$. For every track ρ of \mathcal{X} , with $\text{fst}(\rho) = v$ and $|\rho| \geq 2$, the unravelling algorithm returns a track ρ' of \mathcal{X} , with $\text{fst}(\rho') = v$, such that ρ and ρ' have the same B_k -descriptor and $|\rho'| \leq \tau(|W|, k)$.*

As an example, $\rho' = v_0 v_1 v_2 v_3 v_3 v_2 v_3 v_2 v_3 v_2 v_1 v_3 v_2 v_3 v_2 v_1 v_2 v_1 v_3 v_2$ is returned by Algorithm 1 in place of the track ρ of Figure 4; it can be checked that ρ'_{ds} does not contain any pair of 3-indistinguishable occurrences of a descriptor element and that ρ and ρ' have the same B_3 -descriptor.

Algorithm 1 $\text{Unrav}(\mathcal{X}, v, k, \text{direction})$

```

1: if direction = FORW then
2:   Unravel  $\mathcal{X}$  starting from  $v$  according to  $\ll \triangleright$  “ $\ll$ ” is an arbitrary order of the nodes of  $\mathcal{X}$ 
3:   For every new node of the unravelling met during the visit, return the track  $\rho$  from  $v$  to the
   current node only if:
4:   if  $k = 0$  then
5:     Apply the 0-termination criterion
6:   else
7:     if The last descriptor element  $d$  of (the descriptor sequence of) the current track  $\rho$  is
      $k$ -indistinguishable from a previous occurrence of  $d$  then
8:       do not return  $\rho$  and backtrack to  $\rho(0, |\rho| - 2) \cdot \bar{v}$ , where  $\bar{v}$  is the minimum state (w.r.t.
        $\ll$ ) greater than  $\rho(|\rho| - 1)$  such that  $(\rho(|\rho| - 2), \bar{v})$  is an edge of  $\mathcal{X}$ .
9:     else if direction = BACKW then
10:      Unravel  $\bar{\mathcal{X}}$  starting from  $v$  according to  $\ll \triangleright \bar{\mathcal{X}}$  is  $\mathcal{X}$  with transposed edges
11:      For every new node of the unravelling met during the visit, consider the track  $\rho$  from the
      current node to  $v$ , and recalculate descriptor elements indistinguishability from scratch (left to
      right); return the track only if:
12:      if  $k = 0$  then
13:        Apply the 0-termination criterion
14:      else
15:        if There exist two  $k$ -indistinguishable occurrences of a descriptor element  $d$  in (the descriptor
        sequence of) the current track  $\rho$  then
16:          do not return  $\rho$ 
17:        Do not visit tracks of length greater than  $\tau(|W|, k)$ 

```

Algorithm 2 $\text{ModCheck}(\mathcal{X}, \psi)$

```

1:  $k \leftarrow \text{Nest}_B(\psi)$ 
2:  $u \leftarrow \text{New}(\text{Unrav}(\mathcal{X}, w_0, k, \text{FORW}))$ 
3: while  $u.\text{hasMoreTracks}()$  do
4:    $\tilde{\rho} \leftarrow u.\text{getNextTrack}()$ 
5:   if  $\text{Check}(\mathcal{X}, k, \psi, \tilde{\rho}) = 0$  then
6:     return 0: “ $\mathcal{X}, \tilde{\rho} \not\models \psi$ ”
7: return 1: “ $\mathcal{X} \models \psi$ ”

```

In the *forward mode* of Algorithm 1 (used to deal with $\langle A \rangle$ and $\langle \bar{B} \rangle$ modalities), the direction of track exploration and that of indistinguishability checking are the same, so we can stop extending a track as soon as the first pair of k -indistinguishable occurrences of a descriptor element is found in the descriptor sequence, suggesting an easy termination criterion for stopping the unravelling of tracks. In the *backward mode* (exploited in the case of $\langle \bar{A} \rangle$ and $\langle \bar{E} \rangle$ modalities), such a straightforward criterion cannot be adopted, because tracks are explored right to left (the opposite direction with respect to the edges of the Kripke structure), while the indistinguishability relation over descriptor elements is computed left to right. In general, changing the prefix of a considered track requires recomputing from scratch the descriptor sequence and the indistinguishability relation over descriptor elements. In particular, k -indistinguishable occurrences of descriptor elements can be detected in the middle of a subsequence, and not necessarily at the end.

Building on Algorithm 1 we can easily define the model checking procedure $\text{ModCheck}(\mathcal{X}, \psi)$ (Algorithm 2). $\text{ModCheck}(\mathcal{X}, \psi)$ exploits the procedure $\text{Check}(\mathcal{X}, k, \psi, \tilde{\rho})$ (Algorithm 3) which checks a formula ψ of B-nesting depth k against a track $\tilde{\rho}$ of the Kripke structure \mathcal{X} . $\text{Check}(\mathcal{X}, k, \psi, \tilde{\rho})$ basically calls itself recursively on the subformulas of ψ , and uses the unravelling Algorithm 1 to deal with $\langle A \rangle$, $\langle \bar{A} \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$ modalities.

Algorithm 3 $\text{Check}(\mathcal{X}, k, \psi, \tilde{\rho})$

```

1: if  $\psi = \top$  then
2:   return 1
3: else if  $\psi = \perp$  then
4:   return 0
5: else if  $\psi = p \in \mathcal{AP}$  then
6:   if  $p \in \bigcap_{s \in \text{states}(\tilde{\rho})} \mu(s)$  then
7:     return 1 else return 0
8: else if  $\psi = \neg\varphi$  then
9:   return 1 -  $\text{Check}(\mathcal{X}, k, \varphi, \tilde{\rho})$ 
10: else if  $\psi = \varphi_1 \wedge \varphi_2$  then
11:   if  $\text{Check}(\mathcal{X}, k, \varphi_1, \tilde{\rho}) = 0$  then
12:     return 0
13:   else
14:     return  $\text{Check}(\mathcal{X}, k, \varphi_2, \tilde{\rho})$ 
15: else if  $\psi = \langle A \rangle \varphi$  then
16:    $u \leftarrow \text{New}(\text{Unrav}(\mathcal{X}, \text{lst}(\tilde{\rho}), k, \text{FORW}))$ 
17:   while  $u.\text{hasMoreTracks}()$  do
18:      $\rho \leftarrow u.\text{getNextTrack}()$ 
19:     if  $\text{Check}(\mathcal{X}, k, \varphi, \rho) = 1$  then
20:       return 1
21:   return 0
22: else if  $\psi = \langle \bar{A} \rangle \varphi$  then
23:    $u \leftarrow \text{New}(\text{Unrav}(\mathcal{X}, \text{fst}(\tilde{\rho}), k, \text{BACKW}))$ 
24:   while  $u.\text{hasMoreTracks}()$  do
25:      $\rho \leftarrow u.\text{getNextTrack}()$ 
26:     if  $\text{Check}(\mathcal{X}, k, \varphi, \rho) = 1$  then
27:       return 1
28:   return 0
29: else if  $\psi = \langle B \rangle \varphi$  then
30:   for each  $\bar{\rho}$  prefix of  $\tilde{\rho}$  do
31:     if  $\text{Check}(\mathcal{X}, k - 1, \varphi, \bar{\rho}) = 1$  then
32:       return 1
33:   return 0
34: else if  $\psi = \langle \bar{B} \rangle \varphi$  then
35:   for each  $v \in W$  s.t.  $(\text{lst}(\tilde{\rho}), v) \in \delta$  do
36:     if  $\text{Check}(\mathcal{X}, k, \varphi, \tilde{\rho} \cdot v) = 1$  then
37:       return 1
38:    $u \leftarrow \text{New}(\text{Unrav}(\mathcal{X}, v, k, \text{FORW}))$ 
39:   while  $u.\text{hasMoreTracks}()$  do
40:      $\rho \leftarrow u.\text{getNextTrack}()$ 
41:     if  $\text{Check}(\mathcal{X}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$  then
42:       return 1
43:   return 0
44: else if  $\psi = \langle \bar{E} \rangle \varphi$  then
45:   for each  $v \in W$  s.t.  $(v, \text{fst}(\tilde{\rho})) \in \delta$  do
46:     if  $\text{Check}(\mathcal{X}, k, \varphi, v \cdot \tilde{\rho}) = 1$  then
47:       return 1
48:    $u \leftarrow \text{New}(\text{Unrav}(\mathcal{X}, v, k, \text{BACKW}))$ 
49:   while  $u.\text{hasMoreTracks}()$  do
50:      $\rho \leftarrow u.\text{getNextTrack}()$ 
51:     if  $\text{Check}(\mathcal{X}, k, \varphi, \rho \cdot \tilde{\rho}) = 1$  then
52:       return 1
53:   return 0

```

The model checking algorithm `ModCheck` requires *exponential working space*, as it uses an instance of the unravelling algorithm and some additional space for a track $\tilde{\rho}$. Analogously, every recursive call to `Check` needs an instance of the unravelling algorithm and space for a track. There are at most $|\psi|$ simultaneously active calls to `Check`, so the total space needed by the considered algorithms is $(|\psi| + 1) \cdot O(|W| + \text{Nest}_B(\psi)) \cdot \tau(|W|, \text{Nest}_B(\psi))$ bits overall, where $\tau(|W|, \text{Nest}_B(\psi))$ is the maximum length of track representatives, and $O(|W| + \text{Nest}_B(\psi))$ bits are needed to represent a state of \mathcal{X} , a descriptor element, and a counter for k -indistinguishability.

Notice that formulas ψ of the fragment $HS[A, \bar{A}, \bar{B}, \bar{E}]$ can be checked in polynomial space, as for these formulas $\text{Nest}_B(\psi) = 0$.

We conclude this section by proving that the model checking problem for formulas of $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$, interpreted over finite Kripke structures, is NEXP-hard when a suitable encoding of formulas is exploited. Such an encoding is succinct in the sense that the following binary-encoded shorthands are exploited: $\langle B \rangle^k \psi$ stands for k repetitions of $\langle B \rangle$ before ψ , where k is represented in binary (the same for all the other HS modalities); moreover, $\bigwedge_{i=l, \dots, r} \psi(i)$ denotes a conjunction of formulas which contain some occurrences of the index i as exponents (l and r are binary encoded naturals), e.g., $\bigwedge_{i=1, \dots, 5} \langle B \rangle^i \top$. Finally, we denote by $\text{expand}(\psi)$ the expanded form of ψ , where all exponents k are removed from ψ , by explicitly repeating k times each HS modality with such an exponent, and big conjunctions are replaced by conjunctions of formulas without indexes.

► **Theorem 33.** *The model checking problem for $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formulas against finite Kripke structures is NEXP-hard, if formulas are succinctly encoded; otherwise, it is NP-hard.*

This result is obtained by means of a reduction from the acceptance problem for a language L decided by a *non-deterministic one-tape* Turing machine M (w.l.o.g.) that halts in $O(2^{n^k})$ computation steps on any input of size n , where $k > 0$ is a constant.

Finally, it is not difficult to show that there exists a constant $c > 0$ such that, for all succinct $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formulas ψ , $|\text{expand}(\psi)| \leq 2^{|\psi|^c}$. Thus the model checking algorithm still runs in exponential space with respect to the succinct input formula ψ —by preliminarily expanding ψ to $\text{expand}(\psi)$ —as $\tau(|W|, \text{Nest}_B(\text{expand}(\psi)))$ is exponential in $|W|$ and $|\psi|$. This allows us to conclude that the model checking problem for succinct $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ formulas is between NEXP and EXPSPACE.

5 Conclusion and future work

In this paper, we devised an EXPSPACE model checking algorithm for the HS fragments $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ and $HS[A, \bar{A}, E, \bar{B}, \bar{E}]$ (the known bound for full HS is non-elementary [18]). The algorithm rests on a contraction method that allows us to restrict the verification of the input formula to a finite subset of tracks of bounded size, called track representatives. We also proved that the problem is NEXP-hard, provided that a succinct encoding of formulas is used; otherwise, we can only prove that it is NP-hard (we conjecture that, in this latter case, $HS[A, \bar{A}, B, \bar{B}, \bar{E}]$ is PSPACE-hard). As for the other HS fragments, we showed that $HS[A, \bar{A}, \bar{B}, \bar{E}]$ is in PSPACE, and we conjecture that it is PSPACE-complete. Another interesting fragment is $HS[A, \bar{A}]$ (the logic of temporal neighbourhood): it can be easily shown that it is coNP-hard, but we can only think of PSPACE model checking algorithms.

As for future work, it is worth exploring the model checking problem for full HS and its fragments under other semantic interpretations, relaxing the homogeneity assumption. In this respect, existing work on Duration Calculus (DC) model checking seems to be relevant [7, 8, 10, 12, 15, 21]. DC extends interval temporal logic with an explicit notion of state. States are denoted by state expressions and characterized by a duration (the time period during which the system remains in a given state). Recent results on DC model checking as well as an account of related work can be found in [11].

Acknowledgements. The work by Adriano Peron has been supported by the SHERPA collaborative project, which has received funding from the European Community 7-th Framework Programme (FP7/2007-2013) under grant agreements ICT-600958. He is solely responsible for its content. The paper does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein. The work by Angelo Montanari has been supported by the GNCS project *Algorithms to model check and synthesize safety-critical systems*. We would like to thank the reviewers for their useful comments and suggestions.

References

- 1 J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- 2 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.

- 3 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.
- 4 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.
- 5 D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. What’s decidable about Halpern and Shoham’s interval logic? The maximal fragment $AB\bar{B}\bar{L}$. In *LICS*, pages 387–396, 2011.
- 6 E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2002.
- 7 M. Fränzle. Model-checking dense-time Duration Calculus. *Formal Aspects of Computing*, 16(2):121–139, 2004.
- 8 M. Fränzle and M. R. Hansen. Efficient model checking for Duration Calculus? *International Journal of Software and Informatics*, 3(2-3):171–196, 2009.
- 9 J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
- 10 M. R. Hansen. Model-checking discrete Duration Calculus. *Formal Aspects of Computing*, 6(6A):826–845, 1994.
- 11 M. R. Hansen, A. D. Phan, and A. W. Brekling. A practical approach to model checking Duration Calculus using Presburger Arithmetic. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):251–278, 2014.
- 12 K. Lodaya. A language-theoretic view of verification. In *Modern Applications of Automata Theory*, pages 149–170, 2012.
- 13 A. R. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *IJCAI*, pages 1010–1016, 2013.
- 14 A. R. Lomuscio and J. Michaliszyn. Decidability of model checking multi-agent systems against a class of EHS specifications. In *ECAI*, pages 543–548, 2014.
- 15 R. Meyer, J. Faber, J. Hoenicke, and A. Rybalchenko. Model checking Duration Calculus: a practical approach. *Formal Aspects of Computing*, 20(4-5):481–505, 2008.
- 16 A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. Checking Interval Properties of Computations. Technical Report 2015/01, Dept. of Math. and CS, University of Udine, 2015. <https://www.dimi.uniud.it/assets/preprints/1-2015-montanari.pdf>.
- 17 A. Molinari, A. Montanari, and A. Peron. A Model Checking Procedure for Interval Temporal Logics based on Track Representatives. Technical Report 2015/02, Dept. of Math. and CS, University of Udine, 2015. <https://www.dimi.uniud.it/assets/preprints/2-2015-montanari.pdf>.
- 18 A. Montanari, A. Murano, G. Perelli, and A Peron. Checking interval properties of computations. In *TIME*, pages 59–68, 2014.
- 19 A. Montanari, G. Puppis, and P. Sala. Maximal decidable fragments of Halpern and Shoham’s modal logic of intervals. In *ICALP (2)*, LNCS 6199, pages 345–356, 2010.
- 20 B. Moszkowski. *Reasoning About Digital Circuits*. PhD thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.
- 21 P. K. Pandya. Model checking $CTL^*[DC]$. In *TACAS*, LNCS 2031, pages 559–573, 2001.
- 22 I. Pratt-Hartmann. Temporal prepositions and their logic. *Artificial Intelligence*, 166(1-2):1–36, 2005.
- 23 P. Roeper. Intervals and tenses. *Journal of Philosophical Logic*, 9:451–469, 1980.
- 24 Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.