



UNIVERSITÀ
DEGLI STUDI
DI UDINE

Università degli studi di Udine

Constraining Cycle Alternations in Model Checking for Interval Temporal Logic

Original

Availability:

This version is available <http://hdl.handle.net/11390/1089177> since 2016-10-11T13:58:21Z

Publisher:

Published

DOI:10.1016/j.entcs.2016.03.015

Terms of use:

The institutional repository of the University of Udine (<http://air.uniud.it>) is provided by ARIC services. The aim is to enable open access to all the world.

Publisher copyright

(Article begins on next page)

Constraining Cycle Alternations in Model Checking for Interval Temporal Logic

Alberto Molinari^{a,1} Angelo Montanari^{a,2} Adriano Peron^{b,3}

^a *Department of Mathematics and Computer Science, University of Udine*

^b *Department of Electrical Engineering and Information Technology, University of Napoli Federico II*

Abstract

Model checking is one of the most successful techniques in system verification. While a variety of methods and tools exist to check properties expressed in point-based temporal logics, like LTL and CTL, model checking for interval temporal logic has entered the research agenda only very recently. In previous work, we devised a non-elementary model checking procedure for Halpern and Shoham's modal logic of time intervals, interpreted over finite Kripke structures, and an EXPSPACE algorithm for two meaningful fragments of it. In this paper, we show that the latter algorithm can be suitably tailored in order to check a *subset of the computations of a system*, that satisfy a given bound on the number of cycle alternations, by making use of a *polynomial* (instead of exponential) *working space*. We also prove that such a revised algorithm turns out to be complete for Kripke structures whose strongly connected components are simple cycles.

Keywords: Interval Temporal Logic, Model Checking, Computational Complexity

1 Introduction

Model checking is one of the most effective techniques in system verification, that allows one to verify a formal specification of the desired properties of a system against a model of its behavior. It has been widely and systematically investigated in the context of classical, point-based temporal logics, whereas it is still almost unexplored in the interval logic setting. In [4,12], the authors propose interval temporal logic (ITL) as a natural and expressive formalism for temporal representation and reasoning. On the one hand, thanks to its high expressiveness (compared to that of standard point-based logic), ITL is well suited for a number of computer science applications, ranging from computational linguistics to formal verification, from constraint reasoning to planning [10,11]. On the other hand, undecidability of the satisfiability problem for ITLs is the rule and decidability the exception.

¹ Email: molinari.alberto@gmail.com

² Email: angelo.montanari@uniud.it

³ Email: adrperon@unina.it

Halpern and Shoham’s modal logic of time intervals (HS, for short) is probably the most famous logic among ITLs [4]. It 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. The satisfiability problem for HS, interpreted over all relevant (classes of) linear orders, is highly undecidable. Moreover, undecidability rules also over HS fragments; luckily, meaningful exceptions exist, including the interval logic of temporal neighbourhood and the temporal logic of sub-intervals [3].

In this paper, we focus our attention on the model checking problem for HS and its fragments [5,6,7,8,9], for which little work has been done, if compared to LTL or CTL model checking. In the classical formulation of model checking, systems are modelled as (finite) labelled state-transition graphs (Kripke structures), and point-based temporal logics are used to analyse, for each path in the graph, how proposition letters labelling the states change from one state to the next one along the path. In HS model checking, to verify interval properties of computations, we interpret each finite path of a Kripke structure (track) as an interval, whose labeling is defined on the basis of that of the states composing it.

In [5,6], Lomuscio and Michaliszyn address the model checking problem for some HS fragments, extended with epistemic operators. In [5], they focus their attention on the fragment $\text{HS}[B, E, D]$ of Allen’s relations *started-by*, *finished-by*, and *contains* extended with epistemic modalities. They consider a restricted form of model checking, that verifies a specification against a single (finite) initial computation, and prove that it is a PSPACE-complete problem. In addition, they show that the problem for the purely temporal fragment of the logic is in PTIME. In [6], they prove that the model checking problem for the fragment $\text{HS}[A, \bar{B}, L]$ of Allen’s relations *meets*, *starts*, and *before*, extended with epistemic modalities, is decidable in non-elementary time. The radically different complexity of the two fragments is not surprising, as the latter allows one to access infinitely many intervals.

In [7,9], Montanari et al. characterize the model checking problem for full HS, interpreted over finite Kripke structures. As in [5,6], formulas of HS are evaluated over finite paths/tracks obtained from the unravelling of a finite Kripke structure. However, in [7,9] a proposition letter holds over an interval (track) if and only if it holds over all its states (homogeneity principle), while in [5,6] truth of proposition letters is defined over pairs of states (the endpoints of tracks/intervals). This makes it difficult to compare the two research contributions. In [9], 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. In [7], Molinari et al. work out such a proposal in all its technical details, and they prove that the problem is EXPSPACE-hard. In [8], we consider two large HS fragments, namely, $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ of Allen’s relations *meets*, *met-by*, *started-by*, *starts*, and *finishes*, and $\text{HS}[A, \bar{A}, E, \bar{B}, \bar{E}]$ of Allen’s relations *meets*, *met-by*, *finished-by*, *starts*, and *finishes*, and we prove that the model checking problem for them is in EXPSPACE. Moreover, we show that it is NEXP-hard, provided that a succinct encoding of formulas is used (otherwise, we

can only give an NP-hardness result).

In this paper, we show how to suitably tailor the algorithm given in [8] to check a meaningful subset of the computations of a system by using polynomial (instead of exponential) working space. The rationale is closed to that of *bounded model checking* (BMC) [2]. In BMC, one searches for a counterexample to a relevant property in computations whose length is bounded by a given integer k . Either a bug is found, and the procedure ends, or one can increase k and repeat. BMC is in general *incomplete* (if the bound is not high enough), and thus it can only be exploited for *falsification*, that is, to find counterexamples, rather than to check the validity of a formula. Unlike BMC, our approach does not set a constraint on the maximum length of the considered computations. What is bounded is the alternation of different cycles of a Kripke structure in tracks: the proposed algorithm tries to falsify HS formulas by restricting its attention to a subset of (representatives of) possible system computations (tracks), which do not alternate too many times among different cycles of the considered Kripke structure. As a byproduct, we show that the algorithm is complete for Kripke structures whose strongly connected components are simple cycles.

The paper is organized as follows. In Section 2 we provide some background knowledge. In Section 3 we introduce the 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, following [8], we show how it is possible to select a track representative of bounded length from each equivalence class, and we outline a model checking procedure for $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$. In Section 5, we describe the aforementioned PSPACE algorithm that constrains the number of cycle alternations. Conclusions provide a short assessment of the work done.

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 [4], except for equality. Table 1 depicts 6 of the 13 Allen's relations together with the corresponding HS (existential) modalities. The remaining 7 are equality and the 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

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

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: $\psi ::= p \mid \neg\psi \mid \psi \wedge \phi \mid \langle X \rangle\psi \mid \langle \bar{X} \rangle\psi$, with $p \in \mathcal{AP}$ and $X \in \{A, L, B, E, D, O\}$. We will use 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 allowed⁴. All HS modalities can be expressed in terms of modalities $\langle A \rangle$, $\langle B \rangle$, 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 $\text{HS}[X_1, \dots, X_n]$ the fragment of HS that features modalities X_1, \dots, X_n only.

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 $\langle \bar{E} \rangle$ and its semantics can be defined over a multi-modal Kripke structure, here called *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 2.1 [7] 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.

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*), resp., 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:

- 1) $\mathcal{A}, I \models p$ iff $p \in \sigma(I)$, for any proposition letter $p \in \mathcal{AP}$;
- 2) $\mathcal{A}, I \models \neg\psi$ iff $\mathcal{A}, I \not\models \psi$;
- 3) $\mathcal{A}, I \models \psi \wedge \phi$ iff $\mathcal{A}, I \models \psi$ and $\mathcal{A}, I \models \phi$;
- 4) $\mathcal{A}, I \models \langle X \rangle\psi$, for $X \in \{A, B, E\}$, iff there is $J \in \mathbb{I}$ such that $I X_{\mathbb{I}} J$ and $\mathcal{A}, J \models \psi$;
- 5) $\mathcal{A}, I \models \langle \bar{X} \rangle\psi$, for $\bar{X} \in \{\bar{A}, \bar{B}, \bar{E}\}$, iff there is $J \in \mathbb{I}$ 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 allows one to specify system properties by means of HS formulas.

Definition 2.2 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 that hold at that state, while δ is the transition relation that constrains the evolution of the system over time.

⁴ Strict semantics can be easily "relaxed" to include point intervals, and all results we are going to prove hold for non-strict semantics as well.

Definition 2.3 A track ρ of 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 $(v_i, v_{i+1}) \in \delta$ for all $0 \leq i < n$.

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 $0 \leq i < j < |\rho|$, and $\text{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ (resp., $\text{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$) is the set of all proper prefixes (resp., suffixes) of ρ . Notice that the length of tracks, prefixes, and suffixes is greater than 1, as they will be mapped into strict intervals. We say that ρ is an *initial track* if $\text{fst}(\rho) = w_0$. Finally, we 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 can be associated with a finite Kripke structure by interpreting every track as an interval bounded by its first and last states.

Definition 2.4 [7] The abstract interval model induced by $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where $\mathbb{I} = \text{Trk}_{\mathcal{K}}$, $A_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \text{lst}(\rho) = \text{fst}(\rho')\}$, $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Pref}(\rho)\}$, $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \text{Suff}(\rho)\}$, and $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is such that $\sigma(\rho) = \bigcap_{w \in \text{states}(\rho)} \mu(w)$ for all $\rho \in \mathbb{I}$.

In Definition 2.4, relations $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as Allen's relations A , B , and E , respectively. Moreover, according to the definition of σ , a proposition letter $p \in \mathcal{AP}$ holds over $\rho = v_0 \cdots v_n$ iff it holds over all the states v_0, \dots, v_n of ρ . This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval 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 2.5 Let \mathcal{K} be a finite Kripke structure, ρ be a track in $\text{Trk}_{\mathcal{K}}$, and ψ be an HS formula. We say that the pair (\mathcal{K}, ρ) satisfies ψ , denoted by $\mathcal{K}, \rho \models \psi$, iff $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

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

Definition 2.6 Let \mathcal{K} be a finite Kripke structure and ψ be an HS formula. We say that \mathcal{K} models ψ , denoted by $\mathcal{K} \models \psi$, iff $\mathcal{K}, \rho \models \psi$, for all *initial* tracks $\rho \in \text{Trk}_{\mathcal{K}}$.

2.3 The notion of B_k -descriptor

For any finite Kripke structure \mathcal{K} , one can find a corresponding induced abstract interval model $\mathcal{A}_{\mathcal{K}}$, featuring one interval for each track of \mathcal{K} . Since \mathcal{K} has loops (each state must have at least one successor), the number of its tracks, and thus the number of intervals of $\mathcal{A}_{\mathcal{K}}$, is infinite. In [7], 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 having the same structural complexity (i.e., nesting depth of B modality) as φ . Using this 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 [7]. In this paper, we restrict our attention to the model checking problem for the fragment $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ (and the symmetric fragment $\text{HS}[A, \bar{A}, E, \bar{B}, \bar{E}]$).

We start with the definition of some basic notions.

Definition 2.7 Let ψ be an $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula. The B-nesting depth of ψ , denoted by $\text{Nest}_B(\psi)$, is defined by induction on the complexity of the formula:

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

Using Definition 2.7, we can introduce a relation of k -equivalence over tracks.

Definition 2.8 Let \mathcal{K} be a finite Kripke structure and ρ and ρ' be two tracks in $\text{Trk}_{\mathcal{K}}$. We say that ρ and ρ' are k -equivalent if and only if, for every $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula ψ , with $\text{Nest}_B(\psi) = k$, $\mathcal{K}, \rho \models \psi$ if and only if $\mathcal{K}, \rho' \models \psi$.

It can be easily proved that k -equivalence propagates downwards [7].

Proposition 2.9 Let \mathcal{K} be a finite Kripke structure and ρ and ρ' be two tracks in $\text{Trk}_{\mathcal{K}}$. If ρ and ρ' are k -equivalent, then they are h -equivalent, for all $0 \leq h \leq k$.

We now define the key notion of *descriptor* for a track of a Kripke structure [7].

Definition 2.10 Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho \in \text{Trk}_{\mathcal{K}}$, 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:
 - (i) 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' ;
 - (ii) 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' ;
 - (iii) 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 (iii) of Def. 2.10 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, 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 it is a child of

the root of \mathcal{D} . Such a construction is then recursively 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*.

In Fig. 1, we show an example of B_2 -descriptor.

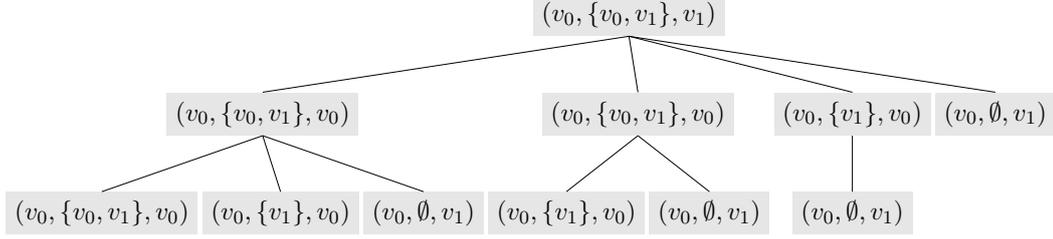


Fig. 1. The figure shows the B_2 -descriptor for the track $\rho = v_0v_1v_0v_0v_0v_0v_1$ of the Kripke structure $\mathcal{K} = (\{p, q\}, \{v_0, v_1\}, \{(v_0, v_0), (v_0, v_1), (v_1, v_0), (v_1, v_1)\}, \mu, v_0)$. It is worth noticing that there exist two distinct prefixes of ρ , that is, the tracks $\rho' = v_0v_1v_0v_0v_0v_0$ and $\rho'' = v_0v_1v_0v_0v_0$, which have the same B_1 -descriptor. Since, according to Definition 2.10, 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 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, but information is enough to decide which $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ formulas are satisfied by the originating track. In [7], the authors prove that, for a finite Kripke structure \mathcal{K} , there is a *finite number* (non-elementary with respect to $|W|$ 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: the k -descriptor equivalence [8].

Definition 2.11 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 (denoted by $\rho \sim_k \rho'$) iff the B_k -descriptors for ρ and ρ' are isomorphic.

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 2.8) [8].

Theorem 2.12 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 $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ with $\text{Nest}_{\text{B}}(\psi) = k$. 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 labelled over descriptor elements is highly redundant: this 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 (at

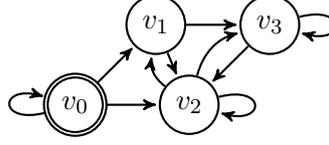


Fig. 2. An example of finite Kripke structure.

least) 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 $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ formula of B -nesting depth up to k .

Definition 3.1 Let $\rho = v_0v_1 \cdots v_n$ be a track of a Kripke structure. The descriptor sequence ρ_{ds} for ρ is $d_0 \cdots d_{n-1}$, where $d_i = \rho_{ds}(i) = (v_0, \text{instates}(v_0 \cdots v_{i+1}), v_{i+1})$, for $0 \leq i < n$. $\text{DElm}(\rho_{ds})$ denotes the set of descriptor elements occurring in ρ_{ds} .

As an example, let us consider the finite Kripke structure of Fig. 2 and the track $\rho = v_0v_0v_0v_1v_2v_1v_2v_3v_3v_2v_2$. The descriptor sequence for ρ is:

$$\rho_{ds} = (v_0, \emptyset, v_0) \boxed{(v_0, \{v_0\}, v_0)} (v_0, \{v_0\}, v_1) (v_0, [v_1], v_2) \\ \boxed{(v_0, [v_2], v_1) (v_0, [v_2], v_2)} (v_0, [v_2], v_3) \boxed{(v_0, [v_3], v_3) (v_0, [v_3], v_2) (v_0, [v_3], v_2)}, \quad (*)$$

where $[v_i] = \{v_0, \dots, v_i\}$ and $\text{DElm}(\rho_{ds}) = \{(v_0, \emptyset, v_0), (v_0, \{v_0\}, v_0), (v_0, \{v_0\}, v_1), (v_0, [v_1], v_2), (v_0, [v_2], v_1), (v_0, [v_2], v_2), (v_0, [v_2], v_3), (v_0, [v_3], v_2), (v_0, [v_3], v_3)\}$.

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, it holds that $d' R_t d''$ if d' and d'' are the descriptor elements of two tracks ρ' and ρ'' , resp., and ρ' is a prefix of ρ'' [8].

Definition 3.2 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, but it is neither reflexive nor symmetric, nor anti-symmetric. It can be easily shown that R_t pairs descriptor elements of increasing prefixes of a track: if $\rho = v_0v_1 \cdots v_n$, 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: 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 $v_{fin} \in S$. 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 $d \not R_t d$. 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. Finally, it can easily be proved that 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 $S' = S''$, and 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: 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 $i, j \in \{1, \dots, s\}$. Due 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 $d' \not R_t d$, or $d' R_t d$ and $d \not R_t d'$.

It is straightforward to check that the descriptor elements of a cluster C are contiguous in ρ_{ds} (they form a subsequence of ρ_{ds}), that is, occurrences of descriptor elements of C are never shuffled with occurrences of descriptor elements not in C .

Definition 3.3 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 .

As we already pointed out, 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. As an example, in the descriptor sequence (*) for the track $\rho = v_0 v_0 v_0 v_1 v_2 v_1 v_2 v_3 v_3 v_2 v_2$ of the Kripke structure in Fig. 2, 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. Moreover, Type-2 elements may have multiple occurrences in a descriptor sequence. Thus, to give a bound on the length of track representatives of B_k -descriptors, we need to somehow limit the number of occurrences of Type-2 elements. 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 [8].

Definition 3.4 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 iff: (i) if $k = 1$: $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$; (ii) if $k \geq 2$: 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 3.4, it follows that two indistinguishable occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$ of the same descriptor element belong to the same subsequence of ρ_{ds} associated with a cluster. The following properties of k -indistinguishability hold [8].

Proposition 3.5 Let $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, be two occurrences of the same descriptor element in a descriptor sequence ρ_{ds} . It holds that: (i) if

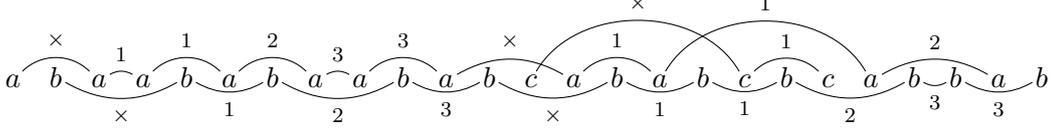


Fig. 3. Let us consider the track $\rho = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2v_2v_3v_2$ of the finite Kripke structure depicted in Fig. 2. Such a track generates the descriptor sequence $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, \{v_1, v_2\}, v_3)abaabababababcbabbab$, where a, b , and c stand for $(v_0, \{v_1, v_2, v_3\}, v_3)$, $(v_0, \{v_1, v_2, v_3\}, v_2)$, 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 thanks to the properties established by Proposition 3.5.

$\rho_{ds}(i)$ and $\rho_{ds}(j)$ are k -indistinguishable, for $k \geq 2$, then they are also $(k - 1)$ -indistinguishable; (ii) if $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are k -indistinguishable and $\rho_{ds}(m) = \rho_{ds}(j)$, for some $i < m < j$, then $\rho_{ds}(m)$ and $\rho_{ds}(j)$ are k -indistinguishable; (iii) if $\rho_{ds}(m) = \rho_{ds}(j)$, for some $i < m < j$, and both the pair $\rho_{ds}(i)$ and $\rho_{ds}(m)$ and the pair $\rho_{ds}(m)$ and $\rho_{ds}(j)$ are k -indistinguishable, then $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are k -indistinguishable.

The fundamental connection between k -indistinguishability of descriptor elements and k -descriptor equivalence of tracks is stated by the next theorem [8].

Theorem 3.6 *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 $\rho(0, i + 1) \sim_k \rho(0, j + 1)$.*

In Fig. 3, we give some examples of k -indistinguishability relations for $k \in \{1, 2, 3\}$.

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.

4 Model checking 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{K}), a *track representative* $\tilde{\rho}$ in \mathcal{K} , 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 $\rho(0, i + 1) \cdot \bar{\rho}$. Indeed, by Theorem 3.6, $\rho(0, i + 1)$ and $\rho(0, j + 1)$ have the same B_k -descriptor, and it is possible to show that, whenever two tracks ρ' and ρ'' have the same B_k -descriptor and $\tilde{\rho}$ is a track such that $(\text{lst}(\rho'), \text{fst}(\tilde{\rho}))$ is an edge of the Kripke structure, then $\rho' \cdot \tilde{\rho}$ and $\rho'' \cdot \tilde{\rho}$ have the same B_k -descriptor [8]. It immediately follows that $\rho = \rho(0, j + 1) \cdot \bar{\rho}$ 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, we have that $\rho(i+1) = \rho(j+1)$, and thus the track $\rho(0, i+1) \cdot \bar{\rho}$ is witnessed in the Kripke structure. By iteratively applying such a *contraction method*, we can find a track, 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.

The next Proposition 4.1 and Theorem 4.2 provide an upper bound to the length of track representatives (their proofs heavily rest on the contraction method) [8].

Proposition 4.1 *Let ρ be a track of $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$, which is associated with a descriptor element d . Then, there is a track representative $\rho' \in \text{Trk}_{\mathcal{K}}$, which is associated with d , such that $|\rho'| \leq 2 + |W|^2$.*

Proposition 4.1 will be used in the unravelling Algorithm 1 as a termination criterion, referred to as *0-termination criterion*: *to get a track representative for every descriptor element with initial state v , witnessed in a finite Kripke structure with set of states W , we can avoid considering tracks longer than $2 + |W|^2$ while exploring the unravelling of the Kripke structure from v .*

Theorem 4.2 *Let ρ be a track of $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\tau(|W|, k) = \min \{1 + (1 + |W|)^{2k+4} + |W|, 1 + (k+3)^{|W|^2+1} + |W|\}$. Then, there exists a track representative $\rho' \in \text{Trk}_{\mathcal{K}}$, associated with the same B_k -descriptor as ρ , such that $|\rho'| \leq \tau(|W|, k)$.*

Theorem 4.2 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{K} to find the track representatives for all witnessed B_k -descriptors. More precisely, in *forward mode* (backward is analogous), for a given $v \in W$ and for every track ρ of \mathcal{K} such that $\text{fst}(\rho) = v$ and $|\rho| \geq 2$, the unravelling algorithm returns a track representative ρ' , with $\text{fst}(\rho') = v$, such that ρ and ρ' have the same B_k -descriptor and $|\rho'| \leq \tau(|W|, k)$. Soundness and completeness are proved in [8].

In the *forward mode* (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. However, the upper bound $\tau(|W|, k)$ on the maximum depth of the unravelling ensures the termination of the algorithm in this mode.

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

if direction = FORW **then**
 Unravel \mathcal{K} starting from v according to $\ll \triangleright \ll$ is an arbitrary order of states
 For every new node *of the unravelling* met during the visit, return the track ρ
 from v to the current node only if:
if $k = 0$ **then**
 Apply the 0-termination criterion
else
 if The last descriptor element d of (the descriptor sequence of) the current
 track ρ is k -indistinguishable from a previous occurrence of d **then**
 do *not* return ρ 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{K} .
 else if direction = BACKW **then**
 Unravel $\overline{\mathcal{K}}$ starting from v according to $\ll \triangleright \overline{\mathcal{K}}$ is \mathcal{K} with transposed edges
 For every new node *of the unravelling* met during the visit, consider the track ρ
 from the current node to v , and recalculate descriptor element indistinguishability
 from scratch (left to right); return the track only if:
 if $k = 0$ **then**
 Apply the 0-termination criterion
 else
 if There exist two k -indistinguishable occurrences of a descriptor element d
 in (the descriptor sequence of) the current track ρ **then** do *not* return ρ
 Do not visit tracks of length greater than $\tau(|W|, k)$

Algorithm 2 ModCheck(\mathcal{K}, ψ)

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

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

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 is $(|\psi| + 1) O(|W| + \text{Nest}_B(\psi)) \tau(|W|, \text{Nest}_B(\psi))$ 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 used to represent a state of \mathcal{K} , a descriptor element, and a counter for k -indistinguishability.

5 Model checking with bounded cycle alternations

In this section, we show how the introduction of a bound to the number of cycle alternations in tracks makes it possible to check formulas of $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ by using *polynomial working space*. To some extent, the proposed approach resembles that

of *bounded model checking* (BMC), where one establishes a bound on the maximum length of considered computations. However, what we bound here is the alternation of different cycles in computations, which is responsible for the exponential length of track representatives. This means that there is not an a priori bound on the length of the tracks (indirectly) checked by the bounded version of the algorithm, the only condition being that their representatives satisfy a suitable constraint on the *number of cyclic subtracks* (which is formally defined in the following). In such a way, we can lower the working space needed by the algorithm—from exponential to polynomial one—at the expense of completeness.

As a warm-up, we focus our attention on a simplified scenario, where every strongly connected component (SCC) of the Kripke structure consists of a (simple) cycle only. Intuitively, in these structures, a track can traverse a cycle many times in a row, but once the cycle (and thus the SCC) is left, it cannot be visited again afterwards by the track. As an example, let us consider a track $\rho = v_0v_1v_2v_3v_1v_2v_3v_1v_2v_3v_1v_2v_3v_1v_4v_5v_4v_5v_6v_7v_7v_7$ of a Kripke structure \mathcal{K} that satisfies the above constraint. In this case, \mathcal{K} has (at least) 3 SCCs, that respectively consist of the sets of states $\{v_1, v_2, v_3\}$, $\{v_4, v_5\}$, and $\{v_7\}$. The descriptor sequence for ρ is $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, [v_2], v_3) \boxed{abcabcabca} (v_0, [v_3], v_4)(v_0, [v_4], v_5) \boxed{(v_0, [v_5], v_4) (v_0, [v_5], v_5)} (v_0, [v_5], v_6)(v_0, [v_6], v_7) \boxed{(v_0, [v_7], v_7) (v_0, [v_7], v_7)}$, where $[v_i] = \{v_1, \dots, v_i\}$, for all i , and $a = (v_0, [v_3], v_1)$, $b = (v_0, [v_3], v_2)$, $c = (v_0, [v_3], v_3)$. The subsequences associated with a cluster are boxed.

A cluster is originated whenever a state of a cycle is visited (at least) twice. Moreover, Type-2 descriptor elements occur in a strictly periodic manner in each subsequence associated with a cluster, due to the constraints of the Kripke structure, admitting only simple cycles as its SCCs. As a consequence, the *second* and the *third* occurrences of a descriptor element d in a subsequence associated with a cluster are 1-indistinguishable, the *third* and the *fourth* are 2-indistinguishable, and so on. Hence, when a track visits a state in a loop for the $(t + 3)$ -th time, with $t \geq 1$, the corresponding occurrence of the descriptor element (the $(t + 2)$ -th one) in the descriptor sequence for that track is t -indistinguishable from the previous occurrence of the same descriptor element (the $(t + 1)$ -th one)⁵. As a consequence, when we consider indistinguishability up to k , any track representative $\tilde{\rho}$ cannot be longer than $1 + (k + 2) \cdot |W|$. The first state occurring in $\tilde{\rho}$ contributes the first addend, and it can be followed by at most $(k + 2) \cdot |W|$ state occurrences, because, otherwise, there would be at least $k + 3$ occurrences of the same state in $\tilde{\rho}$, thus originating a pair of k -indistinguishable occurrences of a descriptor element in $\tilde{\rho}_{ds}$.

If we modify Algorithm 1 in such a way that, while visiting the unravelling of \mathcal{K} , it halts (at the latest) at depth $1 + (k + 2) \cdot |W|$, it immediately follows that the model checking Algorithm 2 uses $(|\psi| + 1) \cdot O(|W| + \text{Nest}_B(\psi)) \cdot O((\text{Nest}_B(\psi) + 2) \cdot |W|)$ bits overall (*polynomial working space*), where ψ is the input formula.

We can now lift such a polynomial space checking procedure from the above special case to the general one, trading completeness for efficiency. In the following,

⁵ Recall that $\rho_{ds}(i) = d$, for $0 \leq i < |\rho_{ds}|$, is the occurrence of the descriptor element d corresponding to $\rho(i + 1)$.

we describe a PSPACE model checking algorithm for the fragment $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ over unrestricted finite Kripke structures.

Definition 5.1 Let \mathcal{K} be a finite Kripke structure and $\rho \in \text{Trk}_{\mathcal{K}}$. An *occurrence of a simple cycle in ρ* is a subtrack $\rho(i, j)$ of ρ , for some $0 \leq i < j < |\rho|$, such that for all $i \leq l < m \leq j$, we have that $\rho(l) = \rho(m)$ if and only if $l = i$ and $m = j$.

If we exclude the starting and ending state occurrences, an occurrence of a simple cycle in a track ρ has no repeated occurrences of the same state of \mathcal{K} .

As an example, the track $\rho = [v_0 v_2 (v_1 v_3 v_0) v_4 \langle v_5 v_1 \rangle v_2 \{v_5\} v_5]$ shows that occurrences of simple cycles in ρ (each of them is delimited by corresponding brackets) may be overlapping, but, by definition, no occurrence can include another one. Moreover, the track $\rho' = v_1 v_2 v_3 (v_4 v_4) v_1 v_2 v_3 [v_5 \{v_5\} v_5] v_1 v_2 v_3$ shows that not all state occurrences necessarily belong to an occurrence of a simple cycle.

We now show how to generalize the notion of occurrence of a simple cycle to allow the cycle to be arbitrarily iterated and followed by a prefix of itself.

Definition 5.2 Let \mathcal{K} be a finite Kripke structure and $\rho \in \text{Trk}_{\mathcal{K}}$. A *cyclic subtrack* is a *maximal* subtrack $\rho(i, j)$ of ρ , for some $0 \leq i < j < |\rho|$, of the form $\rho(i, j) = \rho(i, i')^s \cdot \rho(i, j')$, for some $i \leq j' \leq i' < j$, $s \in \mathbb{N}^+$, and $\rho(i, i' + 1)$ is an occurrence of a simple cycle. We denote the number of cyclic subtracks of ρ by $nc(\rho)$.

We introduce a bound in the model checking algorithm by constraining the number of cyclic subtracks: by setting a bound on $nc(\cdot)$, we can limit the number of alternations of simple cycles in the considered track representatives (up to iterations of each simple cycle). It is worth noticing that, in such a way, we are not imposing a bound on the maximum length of tracks to be taken into consideration by the algorithm (via their representatives), as even a representative $\tilde{\rho}$ with a small value of $nc(\tilde{\rho})$ may represent tracks of unbounded length. As an example, $\tilde{\rho} = v_0 (v_1 v_2 v_3)^4$, where $nc(\tilde{\rho}) = 1$, is the representative, with respect to 2-indistinguishability, for all the tracks $v_0 (v_1 v_2 v_3)^s$, $s \geq 4$.

For any given $\ell \in \mathbb{N}$, the next theorem gives us a bound on the length of track representatives $\tilde{\rho}$ such that $nc(\tilde{\rho}) \leq \ell$.

Theorem 5.3 *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure. Any track representative $\tilde{\rho}$ for some tracks of \mathcal{K} (with respect to k -indistinguishability), with $nc(\tilde{\rho}) \leq \ell$, is no longer than $1 + \ell \cdot (1 + (k + 2) \cdot |W| + |W|) + |W|$.*

Proof. By reasoning as in the case of finite Kripke structures featuring only simple cycles as their SCCs, if a state occurs $k + 3$ times or more in a cyclic subtrack ($k \geq 1$), then the corresponding occurrence of the descriptor element in the descriptor sequence is (at least) k -indistinguishable from the preceding one. Thus, if a track features a cyclic subtrack longer than $1 + (k + 2) \cdot |W|$, then it is *not* a representative. Furthermore, at most $|W|$ consecutive occurrences of states not belonging to any occurrence of a simple cycle may occur; otherwise, at least one state repeats, originating an occurrence of a simple cycle. \square

Let $\ell = O(|W|^c)$ for some constant $c \in \mathbb{N}^+$. By modifying Algorithm 1 in such a way that it stops visiting the unravelling of the Kripke structure at the latest at

depth $1 + \ell \cdot (1 + (k + 2) \cdot |W| + |W|) + |W|$, Algorithm 2 needs $(|\psi| + 1) \cdot O(|W| + \text{Nest}_B(\psi)) \cdot O((\text{Nest}_B(\psi) + 2) \cdot |W|^{c+1})$ bits (*polynomial working space*).

This algorithm is in general *incomplete* because the representatives for some tracks of the Kripke structure may be disregarded (if their $nc()$ exceeds the chosen bound). However, the higher c is, the higher number of representatives is taken into account. Obviously, there exists a *completeness threshold*, that is, a high enough value of the constant c which makes the algorithm complete: when the value we choose for ℓ makes the maximum length of the representatives the algorithm considers no lower than $\tau(|W|, \text{Nest}_B(\psi))$, all the possible behaviors of the system are analyzed. However, even small values of ℓ should be enough to find counterexamples to typical properties of transition systems we are interested in, e.g., mutual exclusion of processes, reachability of states, non-starvation, liveness, and so on.

6 Conclusion and future work

In this paper, we outlined a PSPACE model checking algorithm for the HS fragments $\text{HS}[A, \bar{A}, B, \bar{B}, \bar{E}]$ and $\text{HS}[A, \bar{A}, E, \bar{B}, \bar{E}]$. As in bounded model checking, the algorithm is in general incomplete: it aims not at analyzing all the behaviors of a system, but at finding counterexamples to relevant properties. The idea is to consider only some representatives of computations, for which a bound on the number of their cyclic subtracks holds. At the expense of completeness, we have decreased of an exponential the (spatial) complexity of the algorithm given in [8]. We expect that this new algorithm can be easily implemented and used for practical purposes.

References

- [1] Allen, J. F., *Maintaining knowledge about temporal intervals*, Communications of the ACM **26** (1983), pp. 832–843.
- [2] Biere, A., A. Cimatti, E. M. Clarke, M. Fujita and Y. Zhu, *Symbolic model checking using SAT procedures instead of BDDs*, in: *ACM/IEEE Design Automation Conference*, 1999, pp. 317–320.
- [3] Della Monica, D., V. Goranko, A. Montanari and G. Sciavicco, *Interval temporal logics: a journey*, Bull. of the EATCS **105** (2011), pp. 73–99.
- [4] Halpern, J. Y. and Y. Shoham, *A propositional modal logic of time intervals*, Journal of the ACM **38** (1991), pp. 935–962.
- [5] Lomuscio, A. and J. Michaliszyn, *An epistemic Halpern-Shoham logic*, in: *IJCAI*, 2013, pp. 1010–1016.
- [6] Lomuscio, A. R. and J. Michaliszyn, *Decidability of model checking multi-agent systems against a class of EHS specifications*, in: *ECAI*, 2014, pp. 543–548.
- [7] Molinari, A., 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>.
- [8] Molinari, A., A. Montanari and A. Peron, *A Model Checking Procedure for Interval Temporal Logics based on Track Representatives*, in: *CSL*, 2015.
- [9] Montanari, A., A. Murano, G. Perelli and A. Peron., *Checking interval properties of computations*, in: *TIME*, 2014, pp. 59–68.
- [10] Moszkowski, B., “Reasoning About Digital Circuits,” Ph.D. thesis, Dept. of Computer Science, Stanford University, Stanford, CA (1983).
- [11] Pratt-Hartmann, I., *Temporal prepositions and their logic*, Artificial Intelligence **166** (2005), pp. 1–36.
- [12] Venema, Y., *Expressiveness and completeness of an interval tense logic*, Notre Dame Journal of Formal Logic **31** (1990), pp. 529–547.