University of Udine

Department of Mathematics and Computer Science

PREPRINT

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

Alberto Molinari, Angelo Montanari e Adriano Peron

Preprint nr.: 2/2015

Reports available from:
http://www.dimi.uniud.it/preprints

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

Alberto Molinari and Angelo Montanari
Department of Mathematics and Computer Science
University of Udine
Email: molinari.alberto@gmail.com; angelo.montanari@uniud.it

Adriano Peron
Department of Electronic Engineering and Information Technology
University of Napoli
Email: adrperon@unina.it

Abstract

Model checking is commonly recognised as the most effective tool in 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 long enough tracks of a Kripke structure by equivalent shorter ones.

I. Introduction

Model checking is commonly recognised as the most effective tool in system verification. 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 [1]. 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 logics have been proposed as a more expressive formalism for temporal representation and reasoning than standard point-based ones [2]–[4]. On the positive side, expressiveness of interval temporal logics make them well suited for a number of applications in a variety of fields, including formal verification [5], [6], computational linguistics [7], and planning [8]. On the negative side, undecidability is the rule and decidability the exception for the satisfiability problem of interval temporal logics. Moreover, in the few cases of decidable interval logics, the standard proof machinery, like Rabin's theorem, is usually not applicable.

A prominent position among interval temporal logics is occupied by Halpern and Shoham's modal logic of time intervals $HS[A, \overline{A}, B, \overline{B}, E, \overline{E}]$ (HS, for short) [2]. HS features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen's relations [9]), apart from the equality relation. In [2], it has been shown that the satisfiability problem for HS interpreted over all relevant (classes of) linear orders is highly undecidable. Since then, a lot of work has been done on the satisfiability problem for HS fragments, which showed that undecidability rules over them [10]–[12]. However, meaningful exceptions exist, including the interval logic of temporal neighbourhood and the temporal logic of sub-intervals [13]–[16].

In this paper, we focus our attention on the model checking problem for interval temporal logics. While their satisfiability problem has been extensively and systematically investigated in the literature [17], a little work has been done on model checking. In the classical formulation of the model checking problem, systems are usually modelled as (finite) labelled state-transition

graphs, or 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 interpret each finite path of a Kripke structure as an interval, and to suitably define its labelling on the basis of the labelling of the states that compose it.

In [18], [19], Lomuscio and Michaliszyn address the model checking problem for epistemic extensions of some HS fragments. In [18], they focus their attention on the fragment $HS[B, E, D]$ 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. These results do not come as a surprise as they trade expressiveness for efficiency: modalities B, D, and E allow one to access only sub-intervals of the initial one, whose number is quadratic in the length (number of states) of the initial interval. In [19], 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 HS fragment $HS[A, \overline{B}, L]$ extended with epistemic modalities is decidable with a non-elementary upper bound.

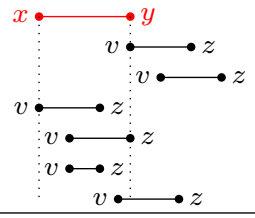
In [20], [21], Montanari et al. outline a general characterization of the model checking problem for full HS, interpreted over finite Kripke structures (under the homogeneity assumption [22]). Their semantic assumptions differ from those made in [18], making it difficult to compare the two research contributions. In both cases, formulas of interval temporal logic are evaluated over finite paths/tracks obtained from the unravelling of a finite Kripke structure. However, in [20] a proposition letter holds over an interval (track) if and only if it holds over all its states (homogeneity principle), while in [18] truth of proposition letters is defined over pairs of states (the endpoints of tracks/intervals). In [20], 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 the non-elementary decidability of the model checking problem for full HS against finite Kripke structures. In [21], Molinari et al. work out such a proposal in all its technical details, and they prove that the problem is EXPSPACE-hard.

In this paper, we prove that the model checking problem for two large HS fragments, namely, $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ and $HS[A, \overline{A}, E, \overline{E}, \overline{B}]$, 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).

The rest of the paper is organised as follows. In Section II, we introduce the considered fragments of HS, and we provide some background knowledge. In Section III 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 IV, we prove a small model theorem, showing that we can select a track representative of bounded length from each equivalence class. In Section V, we outline a model checking procedure, and we analyse its soundness, completeness, and complexity. In addition, we provide a lower bound to the complexity of the problem. Conclusions provide a short assessment of the work and outline future work directions. Due to space limitations, most of the proofs have been moved to an appendix.

TABLE I
ALLEN'S INTERVAL RELATIONS AND CORRESPONDING HS MODALITIES.

Allen's relation	HS	Definition w.r.t. interval structures	Example
MEETS	$\langle \mathrm{A} \rangle$	$[x,y]\mathcal{R}_A[v,z] \iff y = v$	
BEFORE	$\langle \mathrm{L} \rangle$	$[x,y]\mathcal{R}_L[v,z] \iff y < v$	
STARTED-BY	$\langle \mathrm{B} \rangle$	$[x,y]\mathcal{R}_B[v,z] \iff x = v \wedge z < y$	
FINISHED-BY	$\langle \mathrm{E} \rangle$	$[x,y]\mathcal{R}_E[v,z] \iff y = z \wedge x < v$	
CONTAINS	$\langle \mathrm{D} \rangle$	$[x,y]\mathcal{R}_D[v,z] \iff x < v \wedge z < y$	
OVERLAPS	$\langle \mathrm{O} \rangle$	$[x,y]\mathcal{R}_O[v,z] \iff x < v < y < z$	



II. BACKGROUND KNOWLEDGE

A. The interval temporal logic HS

An interval algebra to reason about intervals and their relative order was first proposed by Allen in [9]; then, a systematic logical study of interval representation and reasoning was done by Halpern and Shoham, who introduced the interval temporal logic HS featuring one modality for each Allen interval relation [2]. Table I depicts 6 of the 13 possible binary ordering relations between a pair of intervals, together with the corresponding HS (existential) modality. The other 7 are the equality and the 6 inverse relations (given a generic binary relation $\mathcal{R}$, the inverse relation $\overline{\mathcal{R}}$ is such that $b\overline{\mathcal{R}}a$ if and only if $a\mathcal{R}b$).

The language of HS features a set of proposition letters $\mathcal{AP}$, the Boolean connectives $\neg$ and $\wedge$, the logical constants $\top$ and $\bot$ (respectively *true* and *false*), and a temporal modality for each of the (non trivial) Allen's relations, namely, $\langle \mathrm{A} \rangle, \langle \mathrm{L} \rangle, \langle \mathrm{B} \rangle, \langle \mathrm{E} \rangle, \langle \mathrm{D} \rangle, \langle \mathrm{O} \rangle, \langle \overline{\mathrm{A}} \rangle, \langle \overline{\mathrm{L}} \rangle, \langle \overline{\mathrm{B}} \rangle, \langle \overline{\mathrm{E}} \rangle, \langle \overline{\mathrm{D}} \rangle$, and $\langle \overline{\mathrm{O}} \rangle$,

HS formulas are defined by the following grammar:

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle \psi \mid \langle \overline{X} \rangle \psi, \quad \text{with } p \in \mathcal{AP}$$

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

We will assume the strict semantics of HS: only intervals made of at least two points are allowed (no point-intervals). Under this assumption, all HS modalities can be expressed in terms of modalities $\langle \mathrm{A} \rangle$, $\langle \mathrm{B} \rangle, \langle \mathrm{E} \rangle$, and the transposed modalities $\langle \overline{\mathrm{A}} \rangle, \langle \overline{\mathrm{B}} \rangle, \langle \overline{\mathrm{E}} \rangle$ as follows:

$$\begin{array}{ll} \langle \mathrm{L} \rangle \psi \equiv \langle \mathrm{A} \rangle \langle \mathrm{A} \rangle \psi & \langle \overline{\mathrm{L}} \rangle \psi \equiv \langle \overline{\mathrm{A}} \rangle \langle \overline{\mathrm{A}} \rangle \psi \\ \langle \mathrm{D} \rangle \psi \equiv \langle \mathrm{B} \rangle \langle \mathrm{E} \rangle \psi \equiv \langle \mathrm{E} \rangle \langle \mathrm{B} \rangle \psi & \langle \mathrm{O} \rangle \psi \equiv \langle \mathrm{E} \rangle \langle \overline{\mathrm{B}} \rangle \psi \\ \langle \overline{\mathrm{D}} \rangle \psi \equiv \langle \overline{\mathrm{B}} \rangle \langle \overline{\mathrm{E}} \rangle \psi \equiv \langle \overline{\mathrm{E}} \rangle \langle \overline{\mathrm{B}} \rangle \psi & \langle \overline{\mathrm{O}} \rangle \psi \equiv \langle \mathrm{B} \rangle \langle \overline{\mathrm{E}} \rangle \psi \end{array}$$

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

HS can be viewed as a multi-modal logic with six primitive modalities, namely, $\langle \mathrm{A} \rangle$, $\langle \mathrm{B} \rangle$, $\langle \mathrm{E} \rangle$, and their inverses. Accordingly, HS 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 intervals.

Definition 1: An *abstract interval model* is a tuple $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where:

- $\mathcal{AP}$ is a finite set of proposition letters;
- $\mathbb{I}$ is a possibly infinite set of atomic objects (worlds);
- $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, $E_{\mathbb{I}}$ are three binary relations over $\mathbb{I}$;

- $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is a (total) labeling function, which assigns a set of proposition letters to each world.

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:

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

Satisfiability and *validity* are defined in the usual way: an HS formula ψ is satisfiable if there exists an interval model $\mathcal{A}$ and a world (interval) I such that $\mathcal{A}, I \models \psi$. Moreover, ψ is valid, denoted as $\models \psi$, if $\mathcal{A}, I \models \psi$ for all worlds (intervals) I of any interval model $\mathcal{A}$.

B. Kripke structures and abstract interval models

Finite state systems are usually modelled as finite Kripke structures. In the following, 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: (Finite Kripke structure) 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, $\mu : W \mapsto 2^{\mathcal{AP}}$ a total labelling function, and $w_0 \in W$ is the initial state.

For all $w \in W$, $\mu(w)$ captures 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.

Example 1: Figure 1 below 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 $\mu(v_1) = \{q\}$.

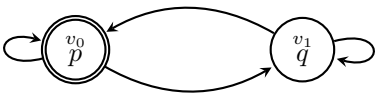


Fig. 1. The Kripke structure $\mathcal{K}_{Equiv}$.

Definition 3: (Track over $\mathcal{K}$) 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 $i \in \{0, \cdots, n-1\}$, $(v_i, v_{i+1}) \in \delta$.

Let $\mathrm{Trk}_\mathcal{K}$ be the (possibly infinite) set of all tracks over a finite Kripke structure $\mathcal{K}$. For any track $\rho = v_0 \cdots v_n \in \mathrm{Trk}_\mathcal{K}$, we define: $|\rho| = n + 1$, $\rho(i) = v_i$, $\mathrm{states}(\rho) = \{v_0, \cdots, v_n\} \subseteq W$, $\mathrm{intstates}(\rho) = \{v_1, \cdots, v_{n-1}\} \subseteq W$, $\mathrm{fst}(\rho) = v_0$ and $\mathrm{lst}(\rho) = v_n$; moreover $\rho(i, j) = v_i \cdots v_j$ is a subtrack of ρ for $0 \leq i < j \leq |\rho| - 1$. Finally, $\mathrm{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper prefixes of ρ, and $\mathrm{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$ 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 $\mathrm{fst}(\rho) = w_0$, where w_0 is the initial state of $\mathcal{K}$, ρ is said to be an *initial track*. In the following, we will sometimes denote by ρ^n the track obtained by concatenating n copies of a given track ρ.

An abstract interval model (over $\mathrm{Trk}_\mathcal{K}$) 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: (Abstract interval model induced by $\mathcal{K}$) The abstract interval model induced by a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is the abstract interval model $\mathcal{A}_\mathcal{K} = (\mathcal{AP}, \mathbb{I}, A_\mathbb{I}, B_\mathbb{I}, E_\mathbb{I}, \sigma)$, where:

- $\mathbb{I} = \mathrm{Trk}_\mathcal{K}$,
- $A_\mathbb{I} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \mathrm{lst}(\rho) = \mathrm{fst}(\rho')\}$,
- $B_\mathbb{I} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Pref}(\rho)\}$,
- $E_\mathbb{I} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Suff}(\rho)\}$, and
- $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ is such that for all $\rho \in \mathbb{I}$,

$$\sigma(\rho) = \bigcap_{w \in \mathrm{states}(\rho)} \mu(w).$$

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 $v_0, \cdots, v_n$ 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 $\mathcal{K}$ be a finite Kripke structure, ρ be a track in $\mathrm{Trk}_\mathcal{K}$, ψ be an HS formula. We say that the pair $(\mathcal{K}, \rho)$ satisfies ψ, denoted by $\mathcal{K}, \rho \models \psi$, if and only if it holds that $\mathcal{A}_\mathcal{K}, \rho \models \psi$.

We are now ready to formally state the *model checking problem* for HS over finite Kripke structures.

Definition 6: (Model checking) Let $\mathcal{K}$ be a finite Kripke structure and ψ be an HS formula. We say that $\mathcal{K}$ models ψ, denoted by $\mathcal{K} \models \psi$, if and only if

$$\text{for all } \textit{initial} \text{ tracks } \rho \in \mathrm{Trk}_\mathcal{K}, \text{ it holds that } \mathcal{K}, \rho \models \psi.$$

Here are some examples of meaningful properties of tracks that can be expressed in HS. To start with, we observe that the formula $[B]\bot$ can be used to select all and only the tracks of length 2. Indeed, given any ρ with $|\rho| = 2$, independently of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models [B]\bot$, because ρ has not (strict) prefixes. On the other hand, it holds that $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle \top$ if (and only if) $|\rho| > 2$. Modality $\langle \mathrm{B} \rangle$ (or, equivalently, $\langle \mathrm{E} \rangle$) can be used to constrain the length of an interval to be greater than, less than, or equal to any value k. Let us denote k nested applications of $\langle \mathrm{B} \rangle$ by $\langle \mathrm{B} \rangle^k$. It holds that $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle^k \top$ if and only if $|\rho| \geq k+2$. Analogously, $\mathcal{K}, \rho \models [B]^k \bot$ if and only if $|\rho| \leq k+1$. Let $\ell(k)$ be a shorthand for $[B]^{k-1}\bot \wedge \langle \mathrm{B} \rangle^{k-2} \top$. It holds that $\mathcal{K}, \rho \models \ell(k)$ if and only if $|\rho| = k$. Modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ can also be exploited to distinguish between tracks encompassing a different number of iterations of a given loop. Finally, modalities $\langle \mathrm{A} \rangle$ and $\langle \overline{\mathrm{A}} \rangle$ can be used to distinguish between tracks that start or end at different states.

C. The notion of B_k-descriptor

For any given 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 $\mathcal{K}$ have 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 [21], 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 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 [21]. In this paper, we restrict our attention to the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ (and

the symmetric fragment $HS[A, \overline{A}, E, \overline{E}, \overline{B}]$) and we show that the model checking problem for it has a lower complexity.

We now start with the definition of some basic notions. The first one is the notion of B-nesting depth of an HS formula.

Definition 7: (B-nesting depth of an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula) Let ψ be an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula. The B-nesting depth of ψ, denoted by $\mathrm{Nest_B}(\psi)$, is defined by induction on the structure complexity of the formula as follows:

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

Making use of the notion of B-nesting depth of a formula, we can define a relation of k-equivalence over tracks.

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

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

Proposition 1: Let $\mathcal{K}$ be a finite Kripke structure and ρ and ρ' be two tracks in $\mathrm{Trk}_{\mathcal{K}}$. If ρ and ρ' are k-equivalent, then they are h-equivalent, for all $0 \leq h \leq k$.

We are now ready to introduce the notion of *descriptor* for a track of a Kripke structure, which will play a fundamental role hereafter.

Definition 9: Let $\mathcal{K}$ be a finite Kripke structure, ρ be a track in $\mathrm{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. The B_k-descriptor for ρ is a labelled tree of depth k, $\mathcal{D} = (V, E, \lambda)$, 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 $\mathcal{D} = (\mathrm{root}(\mathcal{D}), \emptyset, \lambda)$, where $\lambda(\mathrm{root}(\mathcal{D})) = (\mathrm{fst}(\rho), \mathrm{intstates}(\rho), \mathrm{lst}(\rho))$;
- for $k > 0$, the B_k-descriptor for ρ is the tree $\mathcal{D} = (V, E, \lambda)$, where $\lambda(\mathrm{root}(\mathcal{D})) = (\mathrm{fst}(\rho), \mathrm{intstates}(\rho), \mathrm{lst}(\rho))$, which satisfies the following conditions:
  1) for each prefix ρ' of ρ, there exists $v \in V$ such that $(\mathrm{root}(\mathcal{D}), v) \in E$ and the subtree rooted in v is the B_{k-1}-descriptor for ρ';
  2) for each vertex $v \in V$ such that $(\mathrm{root}(\mathcal{D}), v) \in E$, 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 $(\mathrm{root}(\mathcal{D}), v'), (\mathrm{root}(\mathcal{D}), v'') \in E$, 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 9 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 $\mathrm{root}(\mathcal{D})$. 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 $(\mathrm{fst}(\rho), \mathrm{intstates}(\rho), \mathrm{lst}(\rho))$. Then each prefix ρ' of ρ is associated with some subtree, whose root is labelled with $(\mathrm{fst}(\rho'), \mathrm{intstates}(\rho'), \mathrm{lst}(\rho'))$ and it 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*. The following lemma holds.

Lemma 1: For all $k \in \mathbb{N}$, there exists a finite number of possible B_k-descriptors.

Proof: For $k = 0$, there are at most $|W| \cdot 2^{|W|} \cdot |W|$ pairwise distinct B_0-descriptors. As for the inductive step, let us assume h to be the number of pairwise distinct B-descriptors of depth at most k. The number of B_{k+1}-descriptors is at most $|W| \cdot 2^{|W|} \cdot |W| \cdot 2^h$ (there are at most $|W| \cdot 2^{|W|} \cdot |W|$ possible choices for the root, which can have any subset of the h B-descriptors of depth at most k as subtrees). Moreover, by the König's lemma, they are all finite, because their depth is $k + 1$ and the root has a finite number of children (no two subtrees of the root can be isomorphic). ■

Lemma 1 provides an upper bound to the number of distinct B_k-descriptors, and thus to the number of nodes of each B_{k+1}-descriptor, for $k \in \mathbb{N}$, which is *not* elementary with respect to $|W|$ and k. As a matter of fact, this is a very rough upper bound, as some descriptors may not have depth $k + 1$ and some others might not even fulfil the definition of descriptor.

In general, B-descriptors do not convey enough information to determine which track they were built from (this will be clear shortly). However, they can be exploited to determine which $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formulas are satisfied by the track from which they have been built: to check satisfiability of proposition letters, they keep information about initial, final, and internal states of the track; to deal with $\langle \mathrm{A} \rangle \psi$ and $\langle \overline{\mathrm{A}} \rangle \psi$ formulas they store the final and initial states of the track; to deal with $\langle \mathrm{B} \rangle \psi$ formulas, the B-descriptor keeps information about all the prefixes of the track, and no additional information is needed for $\langle \overline{\mathrm{B}} \rangle \psi$ and $\langle \overline{\mathrm{E}} \rangle \psi$ formulas.

Example 2: In Figure 2, we show the B_2-descriptor for the track $\rho = v_0v_1v_0v_0v_0v_0v_1$ of $\mathcal{K}_{Equiv}$. It is worth noticing that there exist two distinct prefixes of track 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 9, 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.

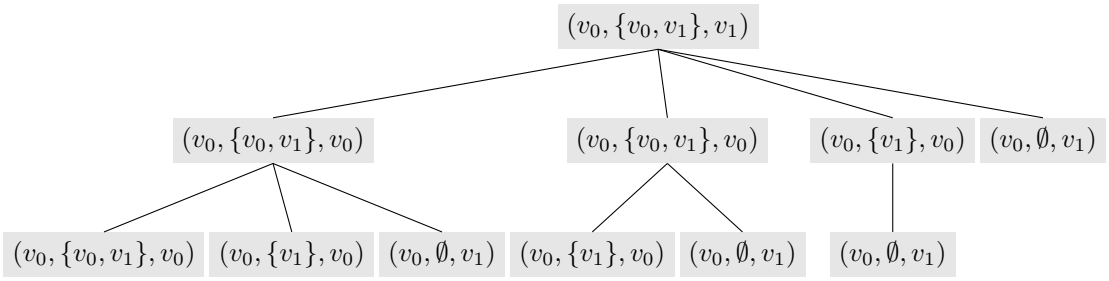


Fig. 2. The B_2-descriptor for the track $v_0v_1v_0v_0v_0v_1$ of $\mathcal{K}_{Equiv}$.

We focus now our attention on the relationships between the tracks obtained from the unravelling of a finite Kripke structure and their B_k-descriptors. A key observation is that, even though the number of tracks of a finite Kripke structure $\mathcal{K}$ is infinite, for any $k \in \mathbb{N}$, the set of B_k-descriptors for its tracks is finite. Thus, at least one B_k-descriptor must be the B_k-descriptor of infinitely many tracks. B_k-descriptors naturally induce an equivalence relation of finite index over the set of tracks of a finite Kripke structure, that we call k-descriptor equivalence relation.

Definition 10: Let $\mathcal{K}$ be a finite Kripke structure, ρ, ρ' be two tracks in $\mathrm{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. We say that ρ and ρ' are k-descriptor equivalent, denoted by $\rho \sim_k \rho'$, if and only if the B_k-descriptors for ρ and ρ' coincide.

Theorem 1 will prove that, for any given pair of tracks $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$, 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, equivalently, the equivalence relation $\sim_k$ has a finite index), there exists

always a finite number of B_k-descriptors that "satisfy" an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula ψ with $\mathrm{Nest_B}(\psi) = k$ (this can be formally proved by a quotient construction [21]). This fact will be of fundamental importance throughout the next section.

Before finally getting to Theorem 1, we need the following lemma.

Lemma 2: Let $k \in \mathbb{N}$, $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure and ρ_1, ρ_1', ρ_2, ρ_2' be tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that: $(\mathrm{lst}(\rho_1), \mathrm{fst}(\rho_1')) \in \delta$, $(\mathrm{lst}(\rho_2), \mathrm{fst}(\rho_2')) \in \delta$, $\rho_1 \sim_k \rho_2$ and $\rho_1' \sim_k \rho_2'$. Then $\rho_1 \cdot \rho_1' \sim_k \rho_2 \cdot \rho_2'$.

The next propositions immediately follow:

Proposition 2: (Right extension) Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, ρ and ρ' be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$. If $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ is such that $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$, then $\rho \cdot \overline{\rho} \sim_k \rho' \cdot \overline{\rho}$.

Proposition 3: (Left extension) Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, ρ and ρ' be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$. If $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ is such that $(\mathrm{lst}(\overline{\rho}), \mathrm{fst}(\rho)) \in \delta$, then $\overline{\rho} \cdot \rho \sim_k \overline{\rho} \cdot \rho'$.

The former proposition states that if we extend the two tracks ρ and ρ' having the same B_k-descriptor "to the right" with the same track $\overline{\rho}$ in $\mathrm{Trk}_{\mathcal{K}}$, with $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$, then the resulting tracks $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$ (both belonging to $\mathrm{Trk}_{\mathcal{K}}$) have the same B_k-descriptor as well. The latter proposition symmetrically deals with the extension of the two tracks ρ and ρ' "to the left". In these Propositions 2 and 3, $|\overline{\rho}| \geq 2$; however both continue to hold if $|\overline{\rho}| = 1$.

Theorem 1: Let $\mathcal{K}$ be a finite Kripke structure, ρ and ρ' two tracks in $\mathrm{Trk}_{\mathcal{K}}$, $\mathcal{A}_{\mathcal{K}}$ the abstract interval model induced by $\mathcal{K}$ and ψ a formula of $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ with $\mathrm{Nest_B}(\psi) = k$. If $\rho \sim_k \rho'$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \psi \iff \mathcal{A}_{\mathcal{K}}, \rho' \models \psi$.

The proof can be found in [21].

III. 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. As an 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. This property immediately follows from the fact that 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 representative track whose length is (exponentially) bounded in both the size of W (the set of states of the Kripke structure) and k.

In order to fix such a bound on the length of track representatives, 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 the prefixes of that track. 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 $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula of B-nesting depth up to k.

Definition 11: Let $\rho = v_0 v_1 \dots v_n$ be a track of a finite Kripke structure. The descriptor sequence ρ_{ds} for ρ is $d_0 \dots d_{n-1}$, where $d_i = \rho_{ds}(i) = (v_0, \mathrm{intstates}(v_0 \dots v_{i+1}), v_{i+1})$, for $i \in \{0, \dots, n-1\}$. We denote the set of descriptor elements occurring in ρ_{ds} by $DElm(\rho_{ds})$.

As an example, let us consider the finite Kripke structure of Figure 3 and the track $\rho = v_0 v_0 v_0 v_1 v_2 v_1 v_2 v_3 v_3 v_2 v_2$. The descriptor sequence for ρ is:

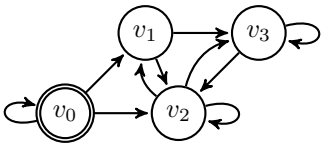


Fig. 3. An example of finite Kripke structure.

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

where $\Delta = \{v_0, v_1, v_2, v_3\}$ and

$$DElm(\rho_{ds}) = \{(v_0, \emptyset, v_0), (v_0, \{v_0\}, v_0), (v_0, \{v_0\}, v_1),$$
$$(v_0, \{v_0, v_1\}, v_2), (v_0, \{v_0, v_1, v_2\}, v_1), (v_0, \{v_0, v_1, v_2\}, v_2),$$
$$(v_0, \{v_0, v_1, v_2\}, v_3), (v_0, \Delta, v_2), (v_0, \Delta, v_3)\}.$$

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

Definition 12: 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,

$$d' \,\mathrm{R_t}\, d'' \text{ if (and only if) } S' \cup \{v'_{fin}\} \subseteq S''.$$

It can be easily checked that the relation $\mathrm{R_t}$ is transitive. For all triple of descriptor elements d', d'', d''', if $d' \,\mathrm{R_t}\, d''$ and $d'' \,\mathrm{R_t}\, d'''$, then $S' \cup \{v'_{fin}\} \subseteq S''$ and $S'' \cup \{v''_{fin}\} \subseteq S'''$. It immediately follows that $S' \cup \{v'_{fin}\} \subseteq S'''$, and thus $d' \,\mathrm{R_t}\, d'''$.

It is worth noticing that $\mathrm{R_t}$ is neither an equivalence relation, nor a quasiorder, since $\mathrm{R_t}$ is neither reflexive (e.g., $(v_0, \{v_0\}, v_1) \not\mathrm{R_t} (v_0, \{v_0\}, v_1)$), nor symmetric (e.g., $(v_0, \{v_0\}, v_1) \,\mathrm{R_t} (v_0, \{v_0, v_1\}, v_1)$ and $(v_0, \{v_0, v_1\}, v_1) \not\mathrm{R_t} (v_0, \{v_0\}, v_1)$), nor antisymmetric (e.g., $(v_0, \{v_1, v_2\}, v_1) \,\mathrm{R_t} (v_0, \{v_1, v_2\}, v_2)$ and $(v_0, \{v_1, v_2\}, v_2) \,\mathrm{R_t}\, (v_0, \{v_1, v_2\}, v_1)$, but the two elements are distinct).

The following proposition shows that $\mathrm{R_t}$ associates descriptor elements of increasing prefixes of the same track.

Proposition 4: Let ρ_{ds} be the descriptor sequence for the track $\rho = v_0 v_1 \cdots v_n$. Then, $\rho_{ds}(i) \,\mathrm{R_t}\, \rho_{ds}(j)$ for all $0 \leq i < j < n$.

Proof: $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are associated with $\rho_1 = v_0 \cdots v_{i+1}$ and $\rho_2 = v_0 \cdots v_{i+1} \cdots v_{j+1}$, respectively, and thus $\mathrm{intstates}(\rho_1) \cup \{v_{i+1}\} \subseteq \mathrm{intstates}(\rho_2)$. ■

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

Definition 13: 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 of Type-1 if and only if $\mathrm{R_t}$ is not reflexive in d: (i) if $d \not\mathrm{R_t}\, d$, then $S \cup \{v_{fin}\} \not\subseteq S$, and thus $v_{fin} \notin S$, and (ii) if $v_{fin} \notin S$, then $d \not\mathrm{R_t}\, d$. It follows that a Type-1 descriptor element cannot occur more than once in a descriptor

sequence. On the contrary, 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 5: If both $d' \,\mathrm{R_t}\, d''$ and $d'' \,\mathrm{R_t}\, d'$, with $d' = (v_{in}, S', v'_{fin})$ 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.

Proof: It holds that $S' \cup \{v'_{fin}\} \subseteq S'' \subseteq S'' \cup \{v''_{fin}\} \subseteq S'$ and $S'' \cup \{v''_{fin}\} \subseteq S' \subseteq S' \cup \{v'_{fin}\} \subseteq S''$. ■

We are now ready to provide 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 $\mathrm{R_t}$ is symmetric, separated by occurrences of Type-1 descriptor elements. Such a characterization can be formalized by means of the notion of cluster.

Definition 14: A cluster $\mathcal{C}$ of (Type-2) descriptor elements is a maximal set of descriptor elements $\{d_1, \ldots, d_s\} \subseteq DElm(\rho_{ds})$ such that $d_i \,\mathrm{R_t}\, d_j$ and $d_j \,\mathrm{R_t}\, d_i$ for all $i, j \in \{1, \ldots, s\}$.

Thanks to maximality, clusters are pairwise disjoint: if $\mathcal{C}$ and $\mathcal{C}'$ are distinct clusters, $d \in \mathcal{C}$ and $d' \in \mathcal{C}'$, either $d \,\mathrm{R_t}\, d'$ and $d' \not\mathrm{R_t}\, d$, or $d' \,\mathrm{R_t}\, d$ and $d \not\mathrm{R_t}\, d'$.

Definition 15: Let ρ_{ds} be a descriptor sequence and $\mathcal{C}$ be one of its clusters. The subsequence of ρ_{ds} associated with $\mathcal{C}$ is a subsequence $\rho_{ds}(i, j)$ such that $\rho_{ds}(i') \in \mathcal{C}$ iff $i \leq i' \leq j < |\rho_{ds}|$.

As an example, with reference to 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 finite Kripke structure in Figure 3, in (*) the subsequences associated with clusters are surrounded by boxes. It is worth observing that:

- the descriptor elements of a cluster $\mathcal{C}$ are contiguous (they form a subsequence), that is, occurrences of descriptor elements in $\mathcal{C}$ are never shuffled with occurrences of descriptor elements not belonging to $\mathcal{C}$;
- two subsequences associated with two distinct clusters $\mathcal{C}$ and $\mathcal{C}'$ in a descriptor sequence must be separated by at least one occurrence of a Type-1 descriptor element (intuitively, in order to "leave" a cluster and to enter another one, a new state—not belonging to the set of already met states—must occur in the track). Type-1 descriptor elements thus act as "separators".

While $\mathrm{R_t}$ 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.

Definition 16: 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 16, it immediately 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}. (In general, it is always the case that $DElm(\rho_{ds}(0, i-1)) \subseteq DElm(\rho_{ds}(0, j-1))$, for $i < j$.) Moreover, 1-indistinguishability guarantees that $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$. From this, it easily follows that the two first occurrences of a descriptor element are not 1-indistinguishable.

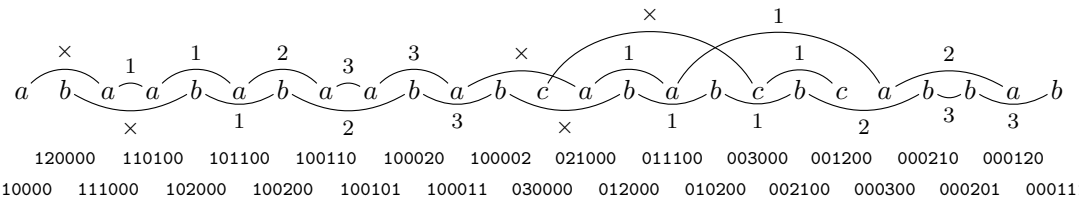


Fig. 4. The track $\rho = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2v_2v_3v_2$ of the finite Kripke structure depicted in Figure 3 generates the descriptor sequence $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, \{v_1, v_2\}, v_3)abaababaababcababcbcabbab$, 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 $\mathcal{C} = \{a, b, c\}$. 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 7 and Proposition 8. At the bottom of the figure, for each position, we report the associated configuration: $c(3) = (2, 1, 0, 0, 0, 0)$, $c(4) = (1, 2, 0, 0, 0, 0)$, and so on.

Proposition 6 and Proposition 7 state some basic properties of the k-indistinguishability relation.

Proposition 6: 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 $(k-1)$-indistinguishable.

Proposition 7: 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 the notions of k-indistinguishability of descriptor elements and k-descriptor equivalence of tracks.

Theorem 2: 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 following proposition easily follows from Theorem 2.

Proposition 8: 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.

IV. 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 (witnessed) B_k-descriptor $\mathcal{D}_{B_k}$, a *track representative* $\tilde{\rho}$, witnessed in the considered finite Kripke structure, 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 exist two occurrences of the same descriptor element $\rho_{ds}(i)$ and

$\rho_{ds}(j)$, with $i < j$, which are k-indistinguishable (we let $\rho = \rho(0, j+1)\cdot\overline{\rho}$ and $\overline{\rho} = \rho(j+2, |\rho|-1)$), then we can replace the track ρ by the k-descriptor equivalent, shorter track $\rho(0, i+1)\cdot\overline{\rho}$. Indeed, by Theorem 2, $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same B_k-descriptor and thus, by Proposition 2, $\rho = \rho(0, j+1)\cdot\overline{\rho}$ and $\rho(0, i+1)\cdot\overline{\rho}$ 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 thus the track $\rho(0, i+1)\cdot\overline{\rho}$ 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. In the rest of the section, we shall consider the problem of establishing 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 preliminary 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 9: For each track ρ of $\mathcal{K}$, with descriptor element d, there exists a track ρ' of $\mathcal{K}$, with the same descriptor element, such that $|\rho'| \leq 2 + |W|^2$.

Proposition 9 will be used in the unravelling algorithm reported in Figure 6 as a termination criterion (referred to as *0-termination criterion*) for the unravelling a finite Kripke structure when it is not necessary to observe multiple occurrences of the same descriptor element.

Definition 17 (0-termination criterion): To get a track representative for all descriptor elements, witnessed in a finite Kripke structure with set of states W and with initial state v, we can avoid to consider 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, in order to bound the length of ρ_{ds}, one has to bound the length of subsequences of ρ_{ds} associated with clusters of Type-2 descriptor elements. Since these subsequences are separated by Type-1 descriptor elements, at most $|W|$ of them, related to distinct clusters, can occur in any descriptor sequence. Finally, for any cluster $\mathcal{C}$, it holds that $|\mathcal{C}| \leq |W|$, because all (Type-2) descriptor elements of $\mathcal{C}$ 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 $\mathcal{C}$, 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.

Given the subsequence associated with a cluster $\mathcal{C}$, we sequentially scan it, 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 $Q_{-2}()$, $Q_{-1}()$, $Q_0()$, $Q_1()$, $Q_2()$, …, $Q_s()$. Array elements are sets of descriptor elements of $\mathcal{C}$. Given an index i, the sets at position i, $Q_{-2}(i)$, $Q_{-1}(i)$, $Q_0(i)$, $Q_1(i)$, $Q_2(i)$, …, $Q_s(i)$ store information about indistinguishabilty for multiple occurrences of descriptor elements in the subsequence up to position $i > u$. To exemplify, if we assume that the scan function finds an occurrence of the descriptor element $d \in \mathcal{C}$ at position i, that is, $\rho_{ds}(i) = d$, we have:

1) $Q_{-2}(i)$ contains all descriptor elements of $\mathcal{C}$ which have never occurred in $\rho_{ds}(u, i)$;

$$f(\rho_{ds}, u) = \big(\mathcal{C} \setminus \{d\}, \{d\}, \emptyset, \cdots, \emptyset\big) \text{ with } \rho_{ds}(u) = d;$$

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

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

Fig. 5. Definition of the scan function f.

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 $\mathcal{C} \setminus \{d\}$ and $Q_0(u)$, $Q_1(u)\ldots$ are empty sets.

In general, arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$, $Q_1()$, $Q_2()$, …, $Q_s()$ satisfy the following constraints:

- for all i, $\bigcup_{m=-2}^{s} Q_m(i) = \mathcal{C}$;
- for all i and all $m \neq m'$, $Q_m(i) \cap Q_{m'}(i) = \emptyset$.

Intuitively, at every position i, $Q_{-2}(i)$, $Q_{-1}(i)$, …, $Q_s(i)$ describe a *state* of the scanning process of the subsequence. The change of the state produced by the transition from position $i-1$ to 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 $\big(Q_{-2}(i), Q_{-1}(i), Q_0(i), \ldots, Q_s(i)\big)$.

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 all elements in $Q_{z''}(i-1)$ for all $z'' \geq z'$ belong to $Q_{z'}(i)$. Consider, for instance, this scenario: in a subsequence of ρ_{ds} associated with some cluster $\mathcal{C}$, $\rho_{ds}(h) = \rho_{ds}(i) = d \in \mathcal{C}$ and $\rho_{ds}(h') = \rho_{ds}(i') = d' \in \mathcal{C}$ 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. We will prove that when a descriptor subsequence is scanned, configurations never repeat, that is, the sequence of configurations is strictly decreasing according to the lexicographical order $>_{lex}$. This property will allow us to establish the bound to the length of track representatives.

Definition 18: 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:

$$c(i) = (|Q_{-2}(i)|, |Q_{-1}(i)|, |Q_0(i)|, |Q_1(i)|, \cdots, |Q_s(i)|),$$

where $f(\rho_{ds}, i) = (Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \cdots, Q_s(i))$.
An example of a configuration sequence is given in Figure 4.

Theorem 3: 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 $\mathcal{C}$. For all $u < i \leq v$, if $\rho_{ds}(i) = d$, then it holds that $d \in Q_s(i-1)$, $d \in Q_{s+1}(i)$, and $c(i-1) >_{lex} c(i)$, for some $s \in \{-2, -1, 0\} \cup \mathbb{N}$.

The proof of Theorem 3 is given in the Appendix.

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 3 guarantees that, while scanning a subsequence, configurations are never repeated. Such a property allows us to fix an upper bound to the length of a track, exceeding which 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 10: 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 10 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 10 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 $\mathcal{C}_1, \mathcal{C}_2, \ldots$, 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}: $|\rho| \leq 1 + (|\mathcal{C}_1| + 1)^{k+2} + (|\mathcal{C}_2| + 1)^{k+2} + \cdots + (|\mathcal{C}_s| + 1)^{k+2} + |W|$, 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 have two possible upper bounds:

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

and

$$|\rho| \leq 1 + (k+3)^{|C_1|} + (k+3)^{|C_2|} + \cdots + (k+3)^{|C_s|} + |W| \leq$$
$$1 + (k+3)^{|C_1|+|C_2|+\cdots+|C_s|} + |W| \leq$$
$$1 + (k+3)^{|DElm(\rho_{ds})|} + |W| \leq 1 + (k+3)^{|W|^2+1} + |W|.$$

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

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

Theorem 4: Let $\mathcal{K}$ be a finite Kripke structure and ρ be a track in $Trk_{\mathcal{K}}$. If $|\rho| > \tau(|W|, k)$, there exists another track in $\mathrm{Trk}_{\mathcal{K}}$, whose length is less than or equal to $\tau(|W|, k)$, which has the same B_k-descriptor as ρ.

Proof: (Sketch) If $|\rho| > \tau(|W|, k)$, then there exists at least one subsequence of ρ_{ds}, associated with some cluster $\mathcal{C}$, which contains at least one pair of k-indistinguishable occurrences of a descriptor element $d \in \mathcal{C}$, say $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $j < i$. By Theorem 2, the two tracks associated with $\rho_{ds}(0, j)$ and $\rho_{ds}(0, i)$, say $\tilde{\rho}_1$ and $\tilde{\rho}_2$, have the same B_k-descriptor. Now, let us rewrite the track ρ as the concatenation $\tilde{\rho}_2 \cdot \overline{\rho}$ for some $\overline{\rho}$. By Proposition 2, the tracks $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ and $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ have the same B_k-descriptor. Since $\mathrm{lst}(\tilde{\rho}_1) = \mathrm{lst}(\tilde{\rho}_2)$ ($\rho_{ds}(j)$ and $\rho_{ds}(i)$ are occurrences of the same descriptor element d), $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ is a track of $\mathcal{K}$ shorter than ρ. If $|\rho'| \leq \tau(|W|, k)$, we have proved the thesis; otherwise, we can iterate the process by applying the above contraction to ρ'. ■

Theorem 4 allows us to specify a termination criterion to bound the depth of the unravelling of a finite Kripke structure, while searching for track representatives for witnessed B_k-descriptors.

Definition 19 ($(k \geq 1)$-termination criterion): For any given $k \geq 1$, to get a track representative for every B_k-descriptor with a given initial state v and witnessed in a finite Kripke structure with set of states W, we can avoid to take into consideration tracks longer than $\tau(|W|, k)$ while exploring the unravelling of the structure from v.

In Figure 6, we outline an *unravelling algorithm*, which search the unravelling of the input Kripke structure $\mathcal{K}$ 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 d in ρ_{ds}.

The following theorem proves soundness and completeness of the unravelling algorithm in Figure 6.

Theorem 5: Let $\mathcal{K}$ be a finite Kripke structure, v be a state in W, and $k \in \mathbb{N}$. For every track ρ of $\mathcal{K}$, with $\mathrm{fst}(\rho) = v$ and $|\rho| \geq 2$, the unravelling algorithm returns a track ρ' of $\mathcal{K}$, with $\mathrm{fst}(\rho') = v$, such that ρ and ρ' have the same B_k-descriptor and $|\rho'| \leq \tau(|W|, k)$.

The proof of Theorem 5 is given in the Appendix.

It basically shows how a "contracted variant" of a track ρ is (indirectly) computed by the algorithm in Figure 6.

As an example, in place of the track ρ of Figure 4, the algorithm returns the following contracted track:

$$\rho' = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2.$$

It can be easily checked that ρ' does not contain any pair of 3-indistinguishable occurrences of a descriptor element and that ρ and ρ' have the same B_3-descriptor.

In the forward modality, 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,

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

Fig. 6. `Unrav`$(\mathcal{K}, v, k,$ direction$)$

suggesting an easy termination criterion for stopping the unravelling of tracks. In the backward modality, such a straightforward criterion cannot be adopted, because tracks are explored right to left (the opposite direction with respect to 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.

Luckily, a heuristic is applicable when dealing with the backward modality: if the descriptor sequence ρ_{ds} for ρ contains a pair of k-indistinguishable occurrences $\rho_{ds}(j)$ and $\rho_{ds}(i)$ of the same descriptor element, with $j < i$, it is possible to skip the exploration of tracks of the form $\overline{\rho} \cdot \rho$, for any $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$. Since $\rho(0, j+1)$ and $\rho(0, i+1)$ have the same B_k-descriptor, by Proposition 3, for any $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $(\mathrm{lst}(\overline{\rho}), \mathrm{fst}(\rho))$ is an edge of $\mathcal{K}$, $\overline{\rho} \cdot \rho(0, i+1)$ and $\overline{\rho} \cdot \rho(0, j+1)$ have the same B_k-descriptor and thus $\overline{\rho} \cdot \rho$ still features the same pair of k-indistinguishable occurrences. Then, the exploration can continue from $\overline{v} \cdot \rho(1, |\rho| - 1)$, where $\overline{v}$ is the minimum state (with respect to the arbitrarily chosen order of nodes $\ll$) greater than $\rho(0)$ such that $(\overline{v}, \rho(1))$ is an edge of $\mathcal{K}$.

V. The model checking algorithm

Building on the unravelling algorithm in Figure 6, we can easily define the model checking procedure `ModCheck`$(\mathcal{K}, \psi)$, whose pseudocode is reported in Figure 7. In particular

1: $k \leftarrow \mathrm{Nest_B}(\psi)$
2: $u \leftarrow New(\texttt{Unrav}(\mathcal{K}, \texttt{init\_state}(\mathcal{K}), k, \text{FORWARD}))$
3: **while** u.`hasMoreTracks()` **do**
4: $\quad \tilde{\rho} \leftarrow u$.`getNextTrack()`
5: $\quad$ **if** $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 0$ **then**
6: $\quad\quad$ **return** 0: "$\mathcal{K}, \tilde{\rho} \not\models \psi$" ▷ Counterexample
7: **return** 1: "$\mathcal{K} \models \psi$"

Fig. 7. $\texttt{ModCheck}(\mathcal{K}, \psi)$

u.`hasMoreTracks()`, in the guard of the while-loop, is true if and only if not all tracks have already been returned by u, which is an instance of the unravelling algorithm; u.`getNextTrack()` returns the next track from u.

$\texttt{ModCheck}(\mathcal{K}, \psi)$ exploits the procedure $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho})$, shown in Figure 8, which checks a formula ψ of B-nesting depth k against a track $\tilde{\rho}$ of the Kripke structure $\mathcal{K}$.

Before proving the correctness of the model checking procedure, we first assess a correctness result for the auxiliary procedure `Check` (the proof is given in the Appendix).

Lemma 3: Let ψ be an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula, with $\mathrm{Nest_B}(\psi) = k$, $\mathcal{K}$ be a Kripke structure, and $\tilde{\rho}$ be a track in $\mathrm{Trk}_{\mathcal{K}}$. The procedure $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho})$ returns 1 if and only if $\mathcal{K}, \tilde{\rho} \models \psi$.

Notice that an optimization step could be introduced at line 32 of Figure 8, before calling `Check` recursively on a prefix of $\tilde{\rho}$: if a prefix $\hat{\rho}_1$ has the same B_{k-1}-descriptor of the current prefix $\hat{\rho}_2$ of $\tilde{\rho}$, and it is shorter than $\hat{\rho}_2$ (it is possible to check the requirement by exploiting descriptor element indistinguishability), and `Check` has already tested $\hat{\rho}_1$, it is possible to skip the call on $\hat{\rho}_2$. Moreover, instead of checking $\hat{\rho}_2 \cdot \rho$, a prefix of $\tilde{\rho}$ for some ρ, it is possible to check $\hat{\rho}_1 \cdot \rho$ (since, by the right extension proposition 2, they have the same B_{k-1}-descriptor).

The following theorem assesses the correctness of the model checking procedure.

Theorem 6: Let ψ be an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula and $\mathcal{K}$ be a finite Kripke structure. $\texttt{ModCheck}(\mathcal{K}, \psi) = 1$ if and only if $\mathcal{K} \models \psi$.

Proof: If $\mathcal{K} \models \psi$, then for all $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\rho) = w_0$ is the initial state of $\mathcal{K}$, we have $\mathcal{K}, \rho \models \psi$. By Lemma 3, it follows that $\texttt{Check}(\mathcal{K}, \mathrm{Nest_B}(\psi), \psi, \rho)$ returns 1. Now, the unravelling procedure returns a subset of the initial tracks. This implies that $\texttt{ModCheck}(\mathcal{K}, \psi)$ returns 1.
If $\texttt{ModCheck}(\mathcal{K}, \psi)$ returns 1, then for any track ρ with $\mathrm{fst}(\rho) = w_0$ returned by the unravelling algorithm, $\texttt{Check}(\mathcal{K}, \mathrm{Nest_B}(\psi), \psi, \rho)$ returns 1 and, by Lemma 3, $\mathcal{K}, \rho \models \psi$. Assume now that a track $\tilde{\rho}$ with $\mathrm{fst}(\tilde{\rho}) = w_0$, is not returned by the unravelling algorithm. By Theorem 5, there is a track $\overline{\rho}$, with $\mathrm{fst}(\overline{\rho}) = w_0$, which is returned in place of $\tilde{\rho}$ and $\overline{\rho}$ has the same B_k-descriptor of $\tilde{\rho}$ (with $k = \mathrm{Nest_B}(\psi)$). Since $\mathcal{K}, \tilde{\rho} \models \psi \iff \mathcal{K}, \overline{\rho} \models \psi$ (by Theorem 1) and $\mathcal{K}, \overline{\rho} \models \psi$, we get that $\mathcal{K}, \tilde{\rho} \models \psi$. So all tracks starting with state w_0 model ψ, implying that $\mathcal{K} \models \psi$. ■

Finally, we observe that the model checking algorithm `ModCheck` is in EXPSPACE. Indeed, `ModCheck` 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. Since there are at most $|\psi|$ (where ψ is the input formula) simultaneously active calls to `Check`, the total space needed by the considered algorithms is $(|\psi| + 1)\, O(|W| + \mathrm{Nest_B}(\psi))\tau(|W|, \mathrm{Nest_B}(\psi))$ bits overall, where $\tau(|W|, \mathrm{Nest_B}(\psi))$ is the maximum length of track representatives, and $O(|W| + \mathrm{Nest_B}(\psi))$ bits are needed to represent a state of $\mathcal{K}$, a descriptor element, and a counter for k-indistinguishability.

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

We conclude this section by proving that the model checking problem for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formulas, interpreted over finite Kripke structures, is NEXP-hard when a suitable encoding of

```
1: if ψ = ⊤ then
2:     return 1
3: else if ψ = ⊥ then
4:     return 0
5: else if ψ = p ∈ 𝒜𝒫 then
6:     if p ∈ ⋂_{s∈states(ρ̃)} μ(s) then
7:         return 1
8:     else
9:         return 0
10: else if ψ = ¬φ then
11:     return 1 − Check(𝒦, k, φ, ρ̃)
12: else if ψ = φ1 ∧ φ2 then
13:     if Check(𝒦, k, φ1, ρ̃) = 0 then
14:         return 0
15:     else
16:         return Check(𝒦, k, φ2, ρ̃)
17: else if ψ = ⟨A⟩ φ then
18:     u ← New(Unrav(𝒦, lst(ρ̃), k, FORWARD))
19:     while u.hasMoreTracks() do
20:         ρ ← u.getNextTrack()
21:         if Check(𝒦, k, φ, ρ) = 1 then
22:             return 1
23:     return 0
24: else if ψ = ⟨Ā⟩ φ then
25:     u ← New(Unrav(𝒦, fst(ρ̃), k, BACKWARD))
26:     while u.hasMoreTracks() do
27:         ρ ← u.getNextTrack()
28:         if Check(𝒦, k, φ, ρ) = 1 then
29:             return 1
30:     return 0
31: else if ψ = ⟨B⟩ φ then
32:     for each ρ̄ prefix of ρ̃ do
33:         if Check(𝒦, k − 1, φ, ρ̄) = 1 then
34:             return 1
35:     return 0
36: else if ψ = ⟨B̄⟩ φ then
37:     for each v ∈ W such that (lst(ρ̃), v) is an edge of 𝒦 do
38:         if Check(𝒦, k, φ, ρ̃ · v) = 1 then
39:             return 1
40:         u ← New(Unrav(𝒦, v, k, FORWARD))
41:         while u.hasMoreTracks() do
42:             ρ ← u.getNextTrack()
43:             if Check(𝒦, k, φ, ρ̃ · ρ) = 1 then
44:                 return 1
45:     return 0
46: else if ψ = ⟨Ē⟩ φ then
47:     ···                                   ▷ Symmetric to ψ = ⟨B̄⟩ φ.
```

Fig. 8. Check$(\mathcal{K}, k, \psi, \tilde{\rho})$

formulas is exploited. Such an encoding is succinct in the sense that the following binary-encoded shorthands are allowed: $\langle \mathrm{B} \rangle^k \psi$ stands for k repetitions of $\langle \mathrm{B} \rangle$ before ψ, where k is represented in binary. The same can be done for all other HS modalities. Moreover, $\bigwedge_{i=l,\cdots,r} \psi(i)$ denotes a conjunction of formulas which contain some occurrences of the index i as exponents (l and r are binary encoded naturals), for example $\bigwedge_{i=1,\cdots,5} \langle \mathrm{B} \rangle^i \top$.

We finally denote by expand(ψ) the expanded form of ψ: all exponents k have to be eliminated from ψ by explicitly repeating k times each HS modality with such an exponent, and big conjunctions must be replaced by conjunctions of formulas without indexes.

In the following theorem we will prove that the model checking problem for succinct $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formulas is NEXP-hard, otherwise—if formulas are not succinctly encoded—it is NP-hard. The result is obtained by reducing 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 $k \in \mathbb{N}^+$ is a constant. To reduce the problem we suitably define a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula ψ such that $\mathcal{K} \models \psi$ if and only if M accepts its input string $c_0 c_1 \cdots c_{n-1}$.

Theorem 7: The model checking problem for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formulas against Kripke structures is NEXP-hard, if formulas are succinctly encoded; otherwise the problem is NP-hard.

If for a succinct $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula ψ, $|\operatorname{expand}(\psi)| \leq 2^{|\psi|^c}$ for some constant $c \in \mathbb{N}^+$, then the model checking algorithm still runs in exponential space with respect to the succinct input formula ψ—by preliminarily expanding ψ to expand(ψ)—as $\tau(|W|, \mathrm{Nest_B}(\operatorname{expand}(\psi))$ is exponential in $|W|$ and $|\psi|$. Indeed, it's not difficult to show that all succinct formulas ψ are such that $|\operatorname{expand}(\psi)| \leq 2^{|\psi|^c}$. Thus we have shown that the model checking problem for succinct $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formulas is between NEXP and EXPSPACE.

VI. Conclusion and future work

In this paper, we devised an EXPSPACE model checking algorithm for the HS fragments $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ and $HS[A, \overline{A}, E, \overline{E}, \overline{B}]$. In addition, we 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). The proposed algorithm rests on a contraction method that allows us to restrict the verification of the formula to a finite subset of tracks of bounded size, called track representatives.

As for the other HS fragments, we showed that $HS[A, \overline{A}, \overline{B}, \overline{E}]$ is in PSPACE, and we conjecture that it and the "orthogonal" fragment $HS[A, \overline{A}, B, E]$ are PSPACE-complete. Another interesting fragment is $HS[A, \overline{A}]$ (the logic of temporal neighbourhood): it can be easily shown that it is NP-hard, but we can only think of PSPACE model checking algorithms.

Last but not least, it is worth exploring the model checking problem for HS and its fragments under other semantic interpretations (relaxing the homogeneity assumption).

References

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

[2] J. Y. Halpern and Y. Shoham, "A propositional modal logic of time intervals," *Journal of the ACM*, vol. 38, no. 4, pp. 935–962, 1991.

[3] Y. Venema, "Expressiveness and completeness of an interval tense logic," *Notre Dame Journal of Formal Logic*, vol. 31, no. 4, pp. 529–547, 1990.

[4] ——, "A modal logic for chopping intervals," *Journal of Logic and Computation*, vol. 1, no. 4, pp. 453–476, 1991.

[5] Z. Chaochen and M. R. Hansen, *Duration Calculus - A Formal Approach to Real-Time Systems*, ser. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[6] B. Moszkowski, "Reasoning about digital circuits," Ph.D. dissertation, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.

[7] I. Pratt-Hartmann, "Temporal prepositions and their logic," *Artificial Intelligence*, vol. 166, no. 1-2, pp. 1–36, 2005.

[8] H. Bowman and S. J. Thompson, "A decision procedure and complete axiomatization of finite interval temporal logic with projection," *Journal of Logic and Computation*, vol. 13, no. 2, pp. 195–239, 2003.

[9] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.
[10] 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*, vol. 71, no. 1-3, pp. 41–83, 2014.
[11] K. Lodaya, "Sharpening the undecidability of interval temporal logic," in *Proc. of ASIAN*, ser. LNCS 1961, 2000, pp. 290–298.
[12] J. Marcinkowski and J. Michaliszyn, "The undecidability of the logic of subintervals," *Fundamenta Informaticae*, vol. 131, no. 2, pp. 217–240, 2014.
[13] 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*, vol. 20, no. 1, pp. 133–166, 2010.
[14] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, "Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions," *Annals of Pure and Applied Logic*, vol. 161, no. 3, pp. 289–304, 2009.
[15] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco, "What's decidable about Halpern and Shoham's interval logic? The maximal fragment $\mathsf{ABB\overline{B}L}$," in *LICS'11*. IEEE Comp. Society Press, 2011, pp. 387–396.
[16] A. Montanari, G. Puppis, and P. Sala, "Maximal decidable fragments of Halpern and Shoham's modal logic of intervals," in *ICALP'10 (2)*, ser. LNCS 6199, 2010, pp. 345–356.
[17] D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco, "Interval temporal logics: a journey," *Bull. of the EATCS*, vol. 105, pp. 73–99, 2011.
[18] A. Lomuscio and J. Michaliszyn, "An epistemic Halpern-Shoham logic," in *Proc. of IJCAI*, 2013.
[19] A. R. Lomuscio and J. Michaliszyn, "Decidability of model checking multi-agent systems against a class of EHS specifications," in *Proc. of ECAI*, 2014, pp. 543–548.
[20] A. Montanari, A. Murano, G. Perelli, and A. Peron., "Checking interval properties of computations," in *Proc. of TIME*, 2014, pp. 59–68.
[21] A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron, "Checking Interval Properties of Computations," Dept. of Mathematics and Computer Science, University of Udine, Tech. Rep. 2015/01, 2015.
[22] P. Roeper, "Intervals and tenses," *Journal of Philosophical Logic*, vol. 9, pp. 451–469, 1980.
[23] C. H. Papadimitriou, *Computational complexity*. Addison-Wesley, 1994.

Appendix

A. Proof of Proposition 1

Proof: Let us assume that $\mathcal{K}, \rho \models \psi$, with $0 \leq \mathrm{Nest}_{\mathrm{B}}(\psi) \leq k$. Consider the formula $\langle \mathrm{B} \rangle^k \top$, whose B-nesting depth is equal to k. It trivially holds that either $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle^k \top$ or $\mathcal{K}, \rho \models \neg \langle \mathrm{B} \rangle^k \top$. In the first case, we have that $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle^k \top \wedge \psi$. Since $\mathrm{Nest}_{\mathrm{B}}\left(\langle \mathrm{B} \rangle^k \top \wedge \psi\right) = k$, from the hypothesis, it immediately follows that $\mathcal{K}, \rho' \models \langle \mathrm{B} \rangle^k \top \wedge \psi$, and thus $\mathcal{K}, \rho' \models \psi$. The other case can be dealt with in a symmetric way. ■

B. Proof of Lemma 2

In the proof, we will exploit the fact that if two tracks in $\mathrm{Trk}_{\mathcal{K}}$ have the same B_{k+1}-descriptor, then they also have the same B_k-descriptor. The latter can indeed be obtained from the former by removing the nodes at depth $k+1$ (leaves) and then deleting isomorphic subtrees possibly originated by the removal.

Proof: By induction on k.

$k = 0$: let's assume ρ_1 and ρ_2 are associated with the descriptor element (v_{in}, S, v_{fin}) and ρ'_1 and ρ'_2 with (v'_{in}, S', v'_{fin}). Thus $\rho_1 \cdot \rho'_1$ and $\rho_2 \cdot \rho'_2$ are represented by $(v_{in}, S \cup \{v_{fin}, v'_{in}\} \cup S', v'_{fin})$.

$k > 0$: let $\mathcal{D}_{B_k}$ be $\rho_1 \cdot \rho'_1$'s descriptor and $\mathcal{D}'_{B_k}$ be $\rho_2 \cdot \rho'_2$'s descriptor: their roots are the same as for $k = 0$; let's now consider a prefix ρ of $\rho_1 \cdot \rho'_1$:

- if ρ is a proper prefix of ρ_1, since ρ_1 and ρ_2 have the same B_k-descriptor, there exists a prefix $\overline{\rho}$ of ρ_2 associated with the same subtree of ρ of depth $k-1$ in ρ_1's (ρ_2's) descriptor;
- ρ_1 and ρ_2 have the same B_{k-1}-descriptor because they have the same B_k-descriptor;
- if ρ is a proper prefix of $\rho_1 \cdot \rho'_1$ such that $\rho = \rho_1 \cdot \tilde{\rho}_1$ for some prefix $\tilde{\rho}_1$ of ρ'_1, then two cases have to be taken into account:
  - if $|\tilde{\rho}_1| = 1$, then $\tilde{\rho}_1 = v'_{in}$; but also $\mathrm{fst}(\rho'_2) = v'_{in}$. Let's now consider the B_{k-1}-descriptors for $\rho_1 \cdot v'_{in}$ and $\rho_2 \cdot v'_{in}$: the labels of the roots are the same, $(v_{in}, S \cup \{v_{fin}\}, v'_{in})$, then the subtrees of depth $k-2$ are exactly the same as in ρ_1 and ρ_2's B_{k-1}-descriptor's, (possibly) with the addition of ρ_1's B_{k-2}-descriptor (which is equal to ρ_2's). Thus $\rho_1 \cdot v'_{in}$ and $\rho_2 \cdot v'_{in}$ have the same B_{k-1}-descriptor;
  - otherwise, since $\tilde{\rho}_1$ is a prefix of ρ'_1 of length at least 2, and ρ'_1 and ρ'_2 have the same B_k-descriptor, there exists a prefix $\tilde{\rho}_2$ of ρ'_2 associated with the same subtree of depth $k-1$ as $\tilde{\rho}_1$ (in ρ'_1's B_k-descriptor). Hence, by inductive hypothesis, $\rho_1 \cdot \tilde{\rho}_1$ and $\rho_2 \cdot \tilde{\rho}_2$ have the same B_{k-1}-descriptor.

Therefore we have shown that for any proper prefix of $\rho_1 \cdot \rho'_1$ there exists a proper prefix of $\rho_2 \cdot \rho'_2$ with the same B_{k-1}-descriptor. The inverse may be shown by symmetry. Thus $\mathcal{D}_{B_k}$ is equal to $\mathcal{D}'_{B_k}$. ■

C. Proof of Proposition 6

Proof: The proof is by induction on $k \geq 2$.

Base case. Let $\rho_{ds}(i)$ and $\rho_{ds}(j)$ be two 2-indistinguishable occurrences of a descriptor element d. By definition, for any $\rho_{ds}(i')$, with $i \leq i' < j$, an occurrence of the descriptor element $d' = \rho_{ds}(i')$ must occur before position i, and thus $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$. It immediately follows that $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are 1-indistinguishable.

Inductive step ($k \geq 3$). By definition, 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. By the inductive hypothesis, $\rho_{ds}(\ell)$ and $\rho_{ds}(\ell')$ are $(k-2)$-indistinguishable, which implies that $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are $(k-1)$-indistinguishable. ■

D. Proof of Proposition 7

Proof: The proof is by induction on $k \geq 1$.

Base case. Since $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0,\ m-1))$ and $DElm(\rho_{ds}(0, i-1)) \subseteq DElm(\rho_{ds}(0, j-1)) \subseteq DElm(\rho_{ds}(0, m-1))$, then $DElm(\rho_{ds}(0,\ i-1)) = DElm(\rho_{ds}(0, m-1)) = DElm(\rho_{ds}(0,\ j-1))$.

Inductive step ($k \geq 2$). By hypothesis, all occurrences $\rho_{ds}(i')$, with $i \leq i' < m$, are $(k-1)$-indistinguishable from some occurrence of the same descriptor element before i. In particular, this is true for all occurrences $\rho_{ds}(j')$, with $j \leq j' < m$. The thesis trivially follows. ■

E. Proof of Theorem 2

Proof: Let us assume that $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are k-indistinguishable. We prove by induction on $k \geq 1$ that $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same B_k-descriptor.

Base case ($k = 1$). Since $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are occurrences of the same descriptor element, the B_1-descriptors for $\rho(0, i+1)$ and $\rho(0, j+1)$ have roots labelled by the same descriptor element. Moreover, the children of these B_1-descriptors are in one to one correspondence since, by 1-indistinguishability, $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$.

Inductive step ($k \geq 2$). Since all the prefixes of $\rho(0, i+1)$ are also prefixes of $\rho(0, j+1)$, we just need to consider the prefixes $\rho(0, t)$ with $i+1 \leq t \leq j$. By definition, any occurrence $\rho_{ds}(i')$ with $i \leq i' < j$, is $(k-1)$-indistinguishable from another occurrence $\rho_{ds}(i'')$, with $i'' < i$, of the same descriptor element. By the inductive hypothesis, $\rho(0, i'+1)$ and $\rho(0, i''+1)$ have the same B_{k-1}-descriptor. It follows that, for any proper prefix of $\rho(0, j+1)$ (of length at least 2), there exists a proper prefix of $\rho(0, i+1)$ with the same B_{k-1}-descriptor, which implies that the tracks $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same B_k-descriptor.

Conversely, we prove, by induction on $k > 1$, that if $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* k-indistinguishable, then the B_k-descriptors of $\rho(0, i+1)$ and $\rho(0, j+1)$ are different. We assume $\rho_{ds}(i)$ and $\rho_{ds}(j)$ to be occurrences of the same descriptor element (if this was not the case, the thesis would trivially follow, since the roots of the B_k-descriptors for $\rho(0, i+1)$ and $\rho(0, j+1)$ would be labelled with different descriptor elements).

Base case ($k = 1$). If $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* 1-indistinguishable, then $DElm(\rho_{ds}(0, i-1)) \subset DElm(\rho_{ds}(0, j-1))$. Hence, there exists $d \in DElm(\rho_{ds}(0,\ j-1))$ such that $d \notin DElm(\rho_{ds}(0, i-1))$, and the B_1-descriptor for $\rho(0, j+1)$ has a leaf labelled by d which is not present in the B_1-descriptor for $\rho(0, i+1)$.

Inductive step ($k \geq 2$). If $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* k-indistinguishable, then there exists (at least) one occurrence $\rho_{ds}(i')$, with $i \leq i' < j$, of a descriptor element d which is *not* $(k-1)$-indistinguishable from any occurrence of d before position i. By the inductive hypothesis, $\rho(0, i'+1)$ has a B_{k-1}-descriptor which is not isomorphic to any B_{k-1}-descriptor associated with proper prefixes of $\rho(0, i+1)$. Thus, in the B_k-descriptor for $\rho(0, j+1)$ there exists a subtree of depth $k-1$ such that there is not an isomorphic subtree of depth $k-1$ in the B_k-descriptor for $\rho(0, i+1)$. ■

F. Proof of Proposition 9

Proof: By induction on the length $\ell(\geq 2)$ of ρ.

Base case ($\ell = 2$). The track ρ satisfies the condition $l \leq 2 + |W|^2$.

Inductive step ($\ell > 2$). We distinguish two cases. If ρ_{ds} has not duplicated occurrences of the same descriptor element, $|\rho_{ds}| \leq 1 + |W|^2$, since $|DElm(\rho_{ds})| \leq 1 + |W|^2$, and thus ρ satisfies the condition $l \leq 2 + |W|^2$. If $\rho_{ds}(i) = \rho_{ds}(j)$, for some $0 \leq i < j < |\rho_{ds}|$, $\rho(0, i+1)$ and $\rho(0, j+1)$ are associated with the same descriptor element. Now, $\rho' = \rho(0, i+1) \cdot \rho(j+2, |\rho| - 1)$ is a track of $\mathcal{K}$ since $\rho(i+1) = \rho(j+1)$, and, by Proposition 2, $\rho = \rho(0, j+1) \cdot \rho(j+2, |\rho| - 1)$ and ρ' have the same descriptor element. By the inductive hypothesis, there is a track ρ'' of $\mathcal{K}$ associated with the same descriptor element of ρ' (and ρ) with $|\rho''| \leq 2 + |W|^2$. ■

G. Proof of Theorem 3

Proof: The proof is by induction on $i \geq u+1$.
(Case $i = u+1$) We consider two cases:

1) if $\rho_{ds}(u) = \rho_{ds}(u+1) = d \in \mathcal{C}$, then we have $Q_{-2}(u) = \mathcal{C} \setminus \{d\}$, $Q_{-1}(u) = \{d\}$, $\emptyset = Q_0(u) = Q_1(u) = \cdots = Q_s(u)$. Moreover, it holds $Q_{-2}(u+1) = \mathcal{C} \setminus \{d\}$, $Q_{-1}(u) = \emptyset$, $Q_0(u) = \{d\}$ and $\emptyset = Q_1(u) = Q_2(u) = \cdots = Q_s(u)$. $c(u) >_{lex} c(u+1)$ and the thesis follows.
2) if $d, d' \in \mathcal{C}$, $d \neq d'$ and $\rho_{ds}(u) = d$, $\rho_{ds}(u+1) = d'$, then we have $Q_{-2}(u) = \mathcal{C} \setminus \{d\}$, $Q_{-1}(u) = \{d\}$, $\emptyset = Q_0(u) = Q_1(u) = \cdots = Q_s(u)$. Moreover, it holds $Q_{-2}(u+1) = \mathcal{C} \setminus \{d, d'\}$, $Q_{-1}(u) = \{d, d'\}$, $\emptyset = Q_0(u) = Q_1(u) = \cdots = Q_s(u)$ and $c(u) >_{lex} c(u+1)$, implying the thesis.

(Case $i > u+1$) In the following we say that $\rho_{ds}(\ell)$ and $\rho_{ds}(m)$ ($\ell < m$) are consecutive occurrences of a descriptor element d if there are no other occurrences of d in $\rho_{ds}(\ell+1, m-1)$. We consider the following cases:

1) If $\rho_{ds}(i)$ is the first occurrence of $d \in \mathcal{C}$, then $d \in Q_{-2}(i-1)$, $d \in Q_{-1}(i)$ and $c(i-1) >_{lex} c(i)$.
2) If $\rho_{ds}(i)$ is the second occurrence of $d \in \mathcal{C}$, according to the definition, $\rho_{ds}(i)$ can not be 1-indistinguishable from the previous occurrence of d. So $d \in Q_{-1}(i-1)$ ($\rho_{ds}(u, i-1)$ contains the first occurrence of d) and $d \in Q_0(i)$, proving that $c(i-1) >_{lex} c(i)$.
3) If $\rho_{ds}(i)$ is at least the third occurrence of $d \in \mathcal{C}$, but $\rho_{ds}(i)$ is *not* 1-indistinguishable from the immediately preceding occurrence of d $\rho_{ds}(i')$, (with $i' < i$), then $DElm(\rho_{ds}(u, i'-1)) \subset DElm(\rho_{ds}(u, i-1))$. Hence, there exists a first occurrence of some $d' \in \mathcal{C}$ in $\rho_{ds}(i'+1, i-1)$, say $\rho_{ds}(j) = d'$, for $i'+1 \leq j \leq i-1$. Thus $d \in Q_{-1}(j), \cdots, d \in Q_{-1}(i-1)$ and $d \in Q_0(i)$, proving that $c(i-1) >_{lex} c(i)$.
4) In the remaining cases we assume that $\rho_{ds}(i)$ is *at least the third occurrence* of $d \in \mathcal{C}$. If $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are both occurrences of $d \in \mathcal{C}$ and $\rho_{ds}(i-1)$ is t-indistinguishable, for some $t > 0$, and not $(t+1)$-indistinguishable, from the immediately preceding occurrence of d, then $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are exactly $(t+1)$-indistinguishable. So $d \in Q_t(i-1)$ and $d \in Q_{t+1}(i)$, implying that $c(i-1) >_{lex} c(i)$ (as a particular case, if $\rho_{ds}(i-1)$ and the immediately preceding occurrence are not 1-indistinguishable, then $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are at most 1-indistinguishable).
5) If $\rho_{ds}(i)$ is exactly 1-indistinguishable from the immediately preceding occurrence of d, $\rho_{ds}(j)$ (with $j < i-1$), then $DElm(\rho_{ds}(u, j-1)) = DElm(\rho_{ds}(u, i-1))$, and there are no first occurrences of any $d' \in \mathcal{C}$ in $\rho_{ds}(j, i-1)$. If $\rho_{ds}(j)$ is not 1-indistinguishable from its previous occurrence of d, it immediately follows that $d \in Q_0(j), \cdots, d \in Q_0(i-1)$ and $d \in Q_1(i)$, implying that $c(i-1) >_{lex} c(i)$.
   Otherwise, there exists $j < i' < i$ such that $\rho_{ds}(i') = d'' \in \mathcal{C}$ is not 1-indistinguishable from any occurrence of d'' before j (as a matter of fact, if this was not the case, $\rho_{ds}(i)$ and $\rho_{ds}(j)$ would be 2-indistinguishable); in particular, $\rho_{ds}(i')$ is not 1-indistinguishable from the last occurrence of d'' before j, say $\rho_{ds}(j')$, for some $j' < j$ (such a j' exists since there are no first occurrences in $\rho_{ds}(j+1, i-1)$). Now, if by contradiction every pair of consecutive occurrences of d'' in $\rho_{ds}(j', i')$ were 1-indistinguishable, then by property 8 $\rho_{ds}(j')$ and $\rho_{ds}(i')$ would be 1-indistinguishable. Thus, a pair of consecutive occurrences of d'' exists, where the second element in the pair is $\rho_{ds}(\ell) = d''$ with $j < \ell < i$, such that they are not 1-indistinguishable. By inductive hypothesis, $d'' \in Q_{-1}(\ell-1)$ and $d'' \in Q_0(\ell)$. Therefore, $d \in Q_0(\ell), \cdots, d \in Q_0(i-1)$ (recall that there are no first occurrences between j and i) and $d \in Q_1(i)$, proving that $c(i-1) >_{lex} c(i)$.
6) If $\rho_{ds}(j) = d \in \mathcal{C}$ is at most t-indistinguishable (for some $t \geq 1$) from a preceding occurrence of d and $\rho_{ds}(j)$ and $\rho_{ds}(i) = d$ (with $j < i-1$) are consecutive occurrences of d and they

are $(t+1)$-indistinguishable (by definition of indistinguishability $\rho_{ds}(j)$ and $\rho_{ds}(i)$ cannot be more than $(t+1)$-indistinguishable), any occurrence of $d' \in \mathcal{C}$ in $\rho_{ds}(j+1, i-1)$ is (at least) t-indistinguishable from another occurrence of d' before j. By property 7, all pairs of consecutive occurrences of d' in $\rho_{ds}(j+1, i-1)$ are (at least) t-indistinguishable, hence $d \in Q_t(j)$, $\cdots$, $d \in Q_t(i-1)$ and finally $d \in Q_{t+1}(i)$, proving that $c(i-1) >_{lex} c(i)$.

7) If $\rho_{ds}(j) = d \in \mathcal{C}$ is at most t-indistinguishable (for some $t \geq 1$) from a preceding occurrence of d, and $\rho_{ds}(j)$ and $\rho_{ds}(i) = d$ (with $j < i-1$) are consecutive occurrences of d which are at most $\bar{t}$-indistinguishable, for $1 \leq \bar{t} \leq t$, we preliminary observe that $DElm(\rho_{ds}(u, j-1)) = DElm(\rho_{ds}(u, i-1))$. Then, if a $d'' \in \mathcal{C}, d \neq d''$ occurs in $\rho_{ds}(j+1, i-1)$ which is not 1-indistinguishable from any occurrence of d'' before j, $\bar{t} = 1$ and we are again in case 5. Otherwise all the occurrences of descriptor elements in $\rho_{ds}(j+1, i-1)$ are (at least) 1-indistinguishable from other occurrences before j. Moreover, there exists $j < i' < i$ such that $\rho_{ds}(i') = d' \in \mathcal{C}, d \neq d'$ and it is at most $(\bar{t}-1)$-indistinguishable from another occurrence of d' before j. Analogously to the case 5, $\rho_{ds}(i')$ must be $(\bar{t}-1)$-indistinguishable from the last occurrence of d' before j, say $\rho_{ds}(j')$ with $j' < j$ (it's a consequence of property 7). But two consecutive occurrences of d' in $\rho_{ds}(j', i')$ must then be at most $(\bar{t}-1)$-indistinguishable (if all pairs of occurrences of d' in $\rho_{ds}(j', i')$ were $\bar{t}$-indistinguishable, $\rho_{ds}(i')$ and $\rho_{ds}(j')$ would be $\bar{t}$-indistinguishable as well) where the second occurrence is $\rho_{ds}(\ell) = d'$ for some $j < \ell \leq i'$. By applying the inductive hypothesis, we have $d' \in Q_{\bar{t}-2}(\ell-1)$ and $d' \in Q_{\bar{t}-1}(\ell)$. As a consequence, we have $d \in Q_{\bar{t}-1}(\ell)$, $\cdots$, $d \in Q_{\bar{t}-1}(i-1)$ (all descriptor elements in $\rho_{ds}(j, i)$ are at least $(\bar{t}-1)$-indistinguishable from other occurrences before j) and finally $d \in Q_{\bar{t}}(i)$, implying that $c(i-1) >_{lex} c(i)$.

■

It is worth pointing out that, from the proof of the theorem, it follows that the definition of f is in fact redundant: cases (c) and (e) never happen.

H. Proof of Theorem 5

Proof: The proofs for the forward and backward directions are quite similar. We give the proof for one direction (the forward one), and we omit the proof for the other direction.

If $k = 0$ the thesis follows immediately by definition 17. So let's assume $k \geq 1$. The proof is by induction on $\ell = |\rho|$.

(Case $\ell = 2$) $\rho_{ds} = (\text{fst}(\rho), \emptyset, \text{lst}(\rho))$, an the only descriptor element of the sequence is Type-1. Thus ρ itself is returned by the algorithm.

(Case $\ell > 2$) If in ρ_{ds} there are no pairs of k-indistinguishable occurrences of some descriptor element, the termination criterion of algorithm in figure 6 can never be applied. Thus ρ itself is returned (as soon as it is visited) and its length is at most $\tau(|W|, k)$.

Otherwise, the descriptor sequence of any track ρ can be split into 3 parts: $\rho_{ds} = \rho_{ds1} \cdot \rho_{ds2} \cdot \rho_{ds3}$ where ρ_{ds1} ends with a Type-1 descriptor element and it does not contain pairs of k-indistinguishable occurrences of a descriptor element, ρ_{ds2} is a subsequence of ρ_{ds} associated with a cluster $\mathcal{C}$ of (Type-2) descriptor elements with at least a pair of k-indistinguishable occurrences of descriptor elements and ρ_{ds3} (if it is not the empty sequence) begins with a Type-1 descriptor element. Namely, ρ_{ds2} is the "leftmost" subsequence of ρ_{ds} consisting of elements of a cluster $\mathcal{C}$, with at least a pair of k-indistinguishable occurrences of some descriptor element.

There exist two indexes i, j with $j < i$ such that $\rho_{ds2}(j)$ and $\rho_{ds2}(i)$ are two k-indistinguishable occurrences of some $d \in \mathcal{C}$. By proposition 7, there is a pair i', j' with $j' < i'$ such that $\rho_{ds2}(j')$ and $\rho_{ds2}(i')$ are *consecutive* k-indistinguishable occurrences of d. If there are many such pairs (even for different elements in $\mathcal{C}$), let's consider the one with the lower index i' (namely, precisely the pair which is found earlier by the unravelling algorithm). By theorem 2, the two tracks associated with $\rho_{ds1} \cdot \rho_{ds2}(0, j')$ and $\rho_{ds1} \cdot \rho_{ds2}(0, i')$, say $\tilde{\rho}_1$ and $\tilde{\rho}_2$, have the same B_k-descriptor. Then, by

the right extension proposition 2, the tracks $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ (for some $\overline{\rho}$) and $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ have the same B_k-descriptor.

The algorithm in figure 6 does not return $\tilde{\rho}_2$ and, due to the backtrack step, neither $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ is returned. But since $\mathrm{lst}(\tilde{\rho}_1) = \mathrm{lst}(\tilde{\rho}_2)$ ($\rho_{ds2}(j')$ and $\rho_{ds2}(i')$ are occurrences of the same descriptor element), the unravelling of $\mathcal{K}$ features $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$, as well. Now, by induction hypothesis, a track ρ'' of $\mathcal{K}$ is returned such that ρ' and ρ'' have the same B_k-descriptor, and $|\rho''| \leq \tau(|W|, k)$. ρ has in turn the same B_k-descriptor as ρ''. ■

I. Proof of Lemma 3

Proof: The proof is by induction on the structure of ψ. (Base cases). The cases in which $\psi = \top, \psi = \bot, \psi = p \in \mathcal{AP}$ are trivial. (Inductive cases). The cases in which $\psi = \neg\varphi$, $\psi = \varphi_1 \wedge \varphi_2$ are also trivial and omitted. We focus on the remaining cases.

- $\psi = \langle \mathrm{A} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, then there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\tilde{\rho}) = \mathrm{fst}(\rho)$ and $\mathcal{K}, \rho \models \varphi$. By theorem 5 the unravelling procedure returns $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\overline{\rho}) = \mathrm{fst}(\rho)$ and $\overline{\rho}$ and ρ have the same B_k-descriptor, thus $\mathcal{K}, \overline{\rho} \models \varphi$. By inductive hypothesis, $\texttt{Check}(\mathcal{K}, k, \varphi, \overline{\rho}) = 1$, hence $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$.
  Vice versa, if $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\tilde{\rho}) = \mathrm{fst}(\rho)$ and $\texttt{Check}(\mathcal{K}, k, \varphi, \rho) = 1$. By inductive hypothesis, $\mathcal{K}, \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.
- $\psi = \langle \overline{\mathrm{A}} \rangle \varphi$. The proof is symmetric to case $\psi = \langle \mathrm{A} \rangle \varphi$.
- $\psi = \langle \mathrm{B} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, there exists $\rho \in \mathrm{Pref}(\tilde{\rho})$ such that $\mathcal{K}, \rho \models \varphi$. By inductive hypothesis, $\texttt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. Since all prefixes of $\tilde{\rho}$ are checked, $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. Note that, by definition of descriptors, if $\tilde{\rho}$ is a track representative of a B_k-descriptor $\mathcal{D}_{B_k}$, a prefix of $\tilde{\rho}$ is a representative of a B_{k-1}-descriptor, whose root is a child of the root of $\mathcal{D}_{B_k}$. Vice versa, if $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, then for some $\rho \in \mathrm{Pref}(\tilde{\rho})$, $\texttt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. By inductive hypothesis $\mathcal{K}, \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.
- $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, then there exists ρ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ and $\mathcal{K}, \tilde{\rho} \cdot \rho \models \varphi$. If $|\rho| = 1$, since by induction hypothesis $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$, then $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. Otherwise, the unravelling algorithm returns a track $\overline{\rho}$ with the same B_k-descriptor as ρ. Thus, by the left extension proposition 3, $\tilde{\rho} \cdot \rho$ and $\tilde{\rho} \cdot \overline{\rho}$ have the same B_k-descriptor. Thus $\mathcal{K}, \tilde{\rho} \cdot \overline{\rho} \models \varphi$. So (by inductive hypothesis) $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \overline{\rho}) = 1$ implying that $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. Notice that, given two tracks ρ, ρ' of $\mathcal{K}$, if we are considering $\overline{\rho}$ as the track representative of the B_k-descriptor of ρ, and the unravelling algorithm returns $\overline{\rho}'$ as the representative of the B_k-descriptor of ρ', since by lemma 2 $\rho \cdot \rho'$ and $\overline{\rho} \cdot \overline{\rho}'$ have the same B_k-descriptor, we have that $\overline{\rho} \cdot \overline{\rho}'$ is the representative of the B_k-descriptor of $\rho \cdot \rho'$. Vice versa, if $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists ρ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ and $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$. By inductive hypothesis, $\mathcal{K}, \tilde{\rho} \cdot \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.
- $\psi = \langle \overline{\mathrm{E}} \rangle \varphi$. The proof is symmetric to case $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$.

■

J. Proof of Theorem 7

Proof: Let's consider a language L decided by a *non-deterministic one-tape* Turing machine M (w.l.o.g.) that halts after no more than $2^{n^k} - 3$ computation steps on an input of size n—we are assuming a sufficiently high constant $k \in \mathbb{N}$—hence L belongs to NEXP.

Let Σ and Q be respectively the alphabet and the set of states of M and let $\#$ be a special symbol not in Σ used as separator for configurations (in the following we let $\Sigma' = \Sigma \cup \{\#\}$). The alphabet Σ is assumed to contain the blank symbol $\sqcup$. As usual, a computation of M is a sequence of configurations of M, where each configuration fixes the content of the tape, the position of the head on the tape and the internal state of M.

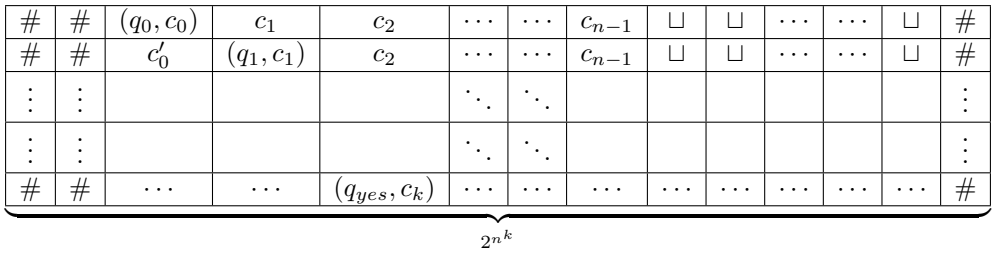


Fig. 9. An example of computation table (tableau).

We exploit a standard encoding for computations called *computation table* (or tableau) (see [23] for further details). Each configuration of M is a sequence over the alphabet $\Gamma = \Sigma' \cup (Q \times \Sigma)$; a symbol in $(q, c) \in Q \times \Sigma$ occurring in the i-th position encodes the fact that the machine has an internal state q and its head is currently on the i-th position of the tape (obviously exactly one occurrence of a symbol in $Q \times \Sigma$ occurs in each configuration). Since M halts after no more than $2^{n^k} - 3$ computation steps, M uses at most $2^{n^k} - 3$ cells on its tape, so the size of a configuration is 2^{n^k} (we need 3 occurrences of the auxiliary symbol #, two for delimiting the beginning of the configuration, and one for the end; additionally M never overwrites delimiters #). If a configuration is actually shorter than 2^{n^k}, it is padded with $\sqcup$ symbols in order to reach length 2^{n^k} (which is a fixed number, once the input length is known). Moreover, since M halts after no more than $2^{n^k} - 3$ computation steps, the number of configurations is $2^{n^k} - 3$. The computation table is basically a matrix of $2^{n^k} - 3$ rows and 2^{n^k} columns, where the i-th row records the configuration of M at the i-th computation step.

As an example, a possible table is depicted in figure 9. In the first configuration (row) the head is in the leftmost position (on the right of the delimiters #) and M is in state q_0. In addition, we have the string symbols $c_0 c_1 \cdots c_{n-1}$ padded with occurrences of $\sqcup$'s to reach length 2^{n^k}. In the second configuration, the head has moved one position to the right, c_0 has been overwritten with c'_0, and M is in state q_1. From the first two rows, we can deduce that the tuple $(q_0, c_0, q_1, c'_0, \rightarrow)$ belongs to the transition relation δ_M of M (we assume that $\delta_M \subseteq Q \times \Sigma \times Q \times \Sigma \times \{\rightarrow, \leftarrow, \bullet\}$ with the obvious standard meaning).

Following [23], we introduce the notion of (legal) window. A window is a 2×3 matrix, in which the first row represents three consecutive symbols of a possible configuration. The second row represents the three symbols which are placed exactly in the same position in the next configuration. A window is legal when the changes from the first to the second row are coherent with δ_M in the obvious sense. Actually, the set of legal windows, which we denote by $Wnd \subseteq \left(\Gamma^3\right)^2$, is a tabular representation of the transition relation δ_M.

For example, two legal windows associated with the table of the previous example are:

#	(q_0, c_0)	c_1
#	c'_0	(q_1, c_1)

(q_0, c_0)	c_1	c_2
c'_0	(q_1, c_1)	c_2

Formally, a $((x, y, z), (x', y', z')) \in Wnd$ can be represented as

x	y	z
x'	y'	z'

with $x, x', y, y', z, z' \in \Gamma$,

where the following constraints have to hold:

1) if all $x, y, z \in \Sigma'$ (x, y, z are not state-symbol pairs), then $y = y'$;
2) if one of x, y and z belongs to $Q \times \Sigma$, then x', y' and z' are coherent with δ_M, and

3) $(x = \# \Rightarrow x' = \#) \wedge (y = \# \Rightarrow y' = \#) \wedge (z = \# \Rightarrow z' = \#)$.

As we said, M never overwrites a #; moreover we can assume the head never visits a #, as well (see [23]). However, in some window, condition 2. would require to move the head right (or left) overwriting # (or just visiting it), while 3. does not allow to replace a # with another symbol (notice that $(q_i, \#)$ does not belong to Γ for any state q_i of M); in such a case the window is not valid and so it is discarded (it doesn't belong to Wnd).

In the following we define a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula ψ such that $\mathcal{K} \models \psi$ if and only if M accepts its input string $c_0 c_1 \cdots c_{n-1}$. The set of propositional letters is $\mathcal{AP} = \Gamma \cup \Gamma^3 \cup \{start\}$. The Kripke structure $\mathcal{K}$ is obtained by suitably composing a basic pattern called *gadget*. An instance of the gadget is associated with a triple of symbols $(a, b, c) \in \Gamma^3$ (i.e. a sequence of three adjacent symbols in a configuration) and consists of 3 states: $q^0_{(a,b,c)}$, $q^1_{(a,b,c)}$, $q^2_{(a,b,c)}$ such that

$$\mu\left(q^0_{(a,b,c)}\right) = \mu\left(q^1_{(a,b,c)}\right) = \{(a, b, c), c\}$$

and

$$\mu\left(q^2_{(a,b,c)}\right) = \emptyset.$$

Moreover,

$$\delta\left(q^0_{(a,b,c)}\right) = \left\{q^1_{(a,b,c)}\right\} \text{ and } \delta\left(q^1_{(a,b,c)}\right) = \left\{q^2_{(a,b,c)}\right\}.$$

The underlying idea is that a gadget associated with $(x, y, z) \in \Gamma^3$ "records" the current proposition letter z, as well as two more "past" letters (x and y).

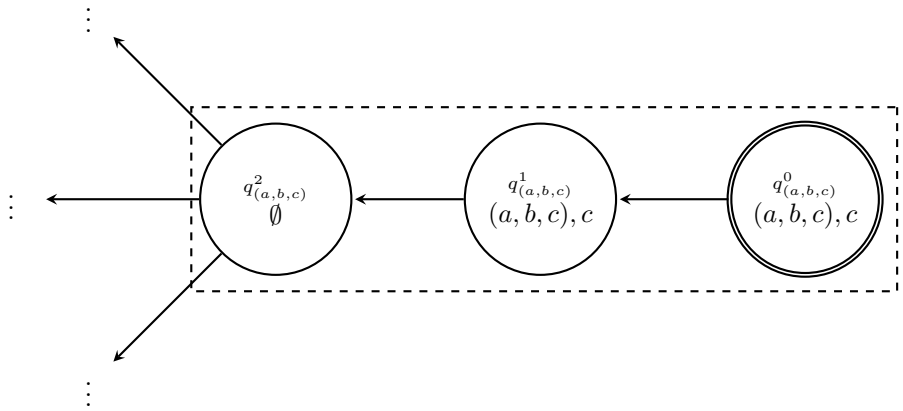


Fig. 10. An instance of the described gadget for $(a, b, c) \in \Gamma^3$.

The Kripke structure $\mathcal{K}$ has (an instance of) a gadget for every $(x, y, z) \in \Gamma^3$ and for all (x, y, z) and (x', y', z') in Γ^3, we have $q^0_{(x',y',z')} \in \delta\left(q^2_{(x,y,z)}\right)$ if and only if $x' = y$ and $y' = z$. Moreover $\mathcal{K}$ has some additional (auxiliary) states $w_0, \cdots, w_6$ described in figure 11 and $\delta(w_6) = \left\{q^0_{(\#,\#,x)} \mid x \in \Gamma\right\}$. Notice that the overall size of $\mathcal{K}$ only depends on $|\Gamma|$ and it is constant w.r.t. to the input string $c_0 c_1 \cdots c_{n-1}$ of M.

Now we want to decide whether an input string belongs to the language L by solving the model checking problem $\mathcal{K} \models start \rightarrow \langle \mathrm{A} \rangle \xi$ where ξ is satisfied only by tracks which represent a successful computation of M. Since the only (initial) track which satisfies *start* is $w_0 w_1$, we are actually verifying the existence of a track which begins with w_1 and satisfies ξ.

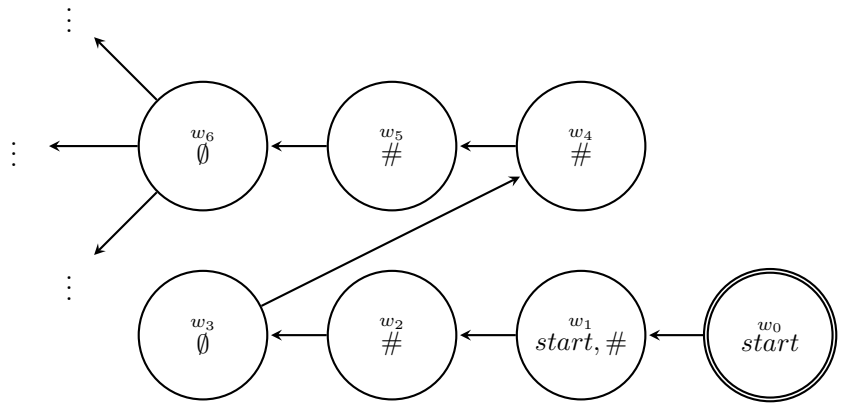


Fig. 11. Initial part of $\mathcal{K}$.

As for ξ, it demands that a track ρ, for which $\mathcal{K}, \rho \models \xi$ (with $\mathrm{fst}(\rho) = w_1$), mimics a successful computation of M in this way: every interval $\rho(i, i+1)$, for $i \mod 3 = 0$, satisfies the proposition letter $p \in \mathcal{AP}$ if and only if the $\frac{i}{3}$-th character of the computation represented by ρ is p (notice that as a consequence of the gadget structure, only ρ's subtracks $\overline{\rho} = \rho(i, i+1)$ for $i \mod 3 = 0$ can satisfy some proposition letters). A symbol of a configuration is mapped to an occurrence of an instance of a gadget in ρ; ρ, in turn, encodes a computation of M through the concatenation of the first, second, third... rows of the computation table (two consecutive configurations are separated by 3 occurrences of #, which require 9 states overall).

Let's now define the HS formula ξ:

$$\xi = \psi_{accept} \wedge \psi_{input} \wedge \psi_{window}$$

where

$$\psi_{accept} = \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle \bigvee_{a \in \Sigma} (q_{yes}, a)$$

and it requires a track to contain an occurrence of the accepting state of M, q_{yes}; ψ_{input} is a bit more involved and demands that the subtrack corresponding to the first configuration of M actually "spells" the input $c_0 c_1 \cdots c_{n-1}$, suitably padded with occurrences of $\sqcup$ and terminated by a # (in the following, $\ell(k)$, introduced in Section II, is satisfied only by those tracks whose length equals k ($k \geq 2$) and it has a binary encoding of $O(\log k)$ bits):

$$\psi_{input} = [B]\Big(\ell(7) \rightarrow \langle \mathrm{A} \rangle (q_0, c_0)\Big) \wedge [B]\Big(\ell(10) \rightarrow \langle \mathrm{A} \rangle\, c_1\Big) \wedge [B]\Big(\ell(13) \rightarrow \langle \mathrm{A} \rangle\, c_2\Big) \wedge$$

$$\vdots$$

$$[B]\Big(\ell(7 + 3(n-1)) \rightarrow \langle \mathrm{A} \rangle\, c_{n-1}\Big) \wedge$$

$$[B]\Bigg(\langle \mathrm{B} \rangle^{5+3n} \top \wedge [B]^{3 \cdot 2^{n^k} - 6} \bot \rightarrow \langle \mathrm{A} \rangle \left(\left(\ell(2) \wedge \bigwedge_{a \in \Gamma} \neg a \right) \vee \sqcup \right) \Bigg) \wedge [B]\Big(\ell\big(3 \cdot 2^{n^k} - 2\big) \rightarrow \langle \mathrm{A} \rangle\, \#\Big).$$

Finally ψ_{window} enforces the window constraint: if the proposition $(d, e, f) \in \Gamma^3$ is witnessed in a subinterval (of length 2) in the subtrack of ρ corresponding to the j-th configuration of

M, then in the same position of (the subtrack of ρ associated with) configuration $j-1$, some $(a,b,c) \in \Gamma^3$ must be there, such that $((a,b,c),(d,e,f)) \in Wnd$.

$$\psi_{window} = [B]\Bigg(\bigwedge_{i=2,\cdots,t} \bigwedge_{(d,e,f)\in\Gamma^3} \Big(\ell(3\cdot 2^{n^k} + 3i + 1) \wedge \langle \mathrm{A} \rangle (d,e,f) \\ \rightarrow [B]\big(\ell(3i+1) \rightarrow \bigvee_{((a,b,c),(d,e,f))\in Wnd} \langle \mathrm{A} \rangle (a,b,c)\big)\Big)\Bigg).$$

where $t = 2^{n^k} \cdot (2^{n^k} - 4) - 1$ is encoded in binary.

All the integers which need to be stored in the formula are less than $(2^{n^k})^2$, thus they need $O(n^k)$ bits; in this way the formula can be generated in polynomial time.

Finally, if the succinct encoding of formulas is not allowed, the proof is basically a simplification of the above one, but we have to restrict ourselves to computations of non-deterministic Turing machines using at most polynomial time. ■