UNIVERSITÀ DEGLI STUDI DI UDINE

DIPARTIMENTO DI MATEMATICA E INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

# MODEL CHECKING AND INTERVAL TEMPORAL LOGICS: CHECKING INTERVAL PROPERTIES OF COMPUTATIONS

Tesi di Laurea Magistrale

Relatore
prof. Angelo MONTANARI

Correlatore
prof. Adriano PERON

Laureando
Alberto MOLINARI

Anno Accademico 2013/2014

UNIVERSITÀ DEGLI STUDI DI UDINE

DIPARTIMENTO DI MATEMATICA E INFORMATICA
CORSO DI LAUREA MAGISTRALE IN INFORMATICA

# MODEL CHECKING AND INTERVAL TEMPORAL LOGICS: CHECKING INTERVAL PROPERTIES OF COMPUTATIONS

## MODEL CHECKING E LOGICHE TEMPORALI A INTERVALLI: VERIFICARE PROPRIETÀ INTERVALLARI DELLE COMPUTAZIONI

Tesi di Laurea Magistrale

Relatore
prof. Angelo MONTANARI

Correlatore
prof. Adriano PERON

Laureando
Alberto MOLINARI

Anno Accademico 2013/2014

**Abstract**

*Model checking* is an alternative to simulation and testing, two classic techniques of software engineering for software validation, which consists in expressing some properties of a finite-state transition system in formulas of a temporal logic and then verifying them over a model of the system itself (usually a finite Kripke structure) through *exhaustive enumeration of all the reachable states*. This technique is *fully automatic* and every time the design violates a desired property, a *counterexample* is produced, which illustrates a behavior that falsifies the property.

All model checking techniques, such as *partial order reduction*, *symbolic* and *bounded* model checking, were developed some years ago bearing in mind the well-known "point-based" temporal logics LTL and CTL. However, while the expressiveness of such logics is beyond doubt, there are some properties we may want to check that are inherently "interval-based" and thus can not be expressed by point-based temporal logics, e.g., "$p$ has to hold in at least an average number of system states in a given computation sector". Here interval temporal logics (ITLs) come into play, providing an alternative setting for reasoning about time. Such logics deal with intervals, instead of points, as their primitive entities; this choice gives them the ability to express temporal properties, such as actions with duration, accomplishments, and temporal aggregations, which cannot be dealt with in standard (point-based) logics.

A prominent position among ITLs is occupied by *Halpern and Shoham's modal logic of time intervals* (HS, for short). HS features one modality for each of the 13 possible ordering relations between pairs of intervals, apart from the equality relation.

Here, we focus our attention on the model checking problem for HS, for which a little work has been done, if compared to model checking for point-based temporal logics.

The idea is to evaluate HS formulas on finite Kripke structures, making it possible to check the correctness of the behavior of systems with respect to meaningful interval properties. To this end, we interpret each finite path (i.e., a *track*) in a Kripke structure as an interval, and we define its labeling on the basis of the labeling of the states that compose it.

Formally, we will show that finite Kripke structures can be suitably mapped into interval-based structures, called *abstract interval models*, over which HS formulas can be interpreted. Such models have in general an *infinite* domain, because finite Kripke structures may have loops and thus infinitely many tracks. In order to devise a model checking procedure for HS over finite Kripke structures, we prove a *small model theorem* showing that, given an HS formula $\psi$ and a finite Kripke structure $\mathcal{K}$, there exists a *finite* interval model which is equivalent to the one induced by $\mathcal{K}$ with respect to the satisfiability of $\psi$. In this way we can prove that the model checking problem for HS interpreted over finite Kripke structures is *decidable* (with a non-elementary upper bound); in addition we show it is EXPSPACE-hard if a succinct encoding of HS formulas is exploited.

Then we restrict our attention to the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$, consisting of HS formulas with modalities $A$, $\overline{A}$, $B$, $\overline{B}$ and $\overline{E}$ only, and we prove that its model checking is in EXPSPACE by exploiting *track representatives*, which are the only *bounded-length* tracks we need to take into consideration when checking an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula over a Kripke structure.

Finally, we consider the fragments $HS[A, \overline{A}, \overline{B}, \overline{E}]$ and $\forall HS[A, \overline{A}, B, E]$, whose model checking problems are PSPACE-complete and coNP-complete, respectively.

**Sommario**

Il *model checking* si propone come un'alternativa alla simulazione e al testing, due tecniche classiche dell'ingegneria del software per la validazione del software; esso consiste nell'esprimere alcune proprietà di un sistema di transizione a stati finiti mediante formule di una logica temporale e, successivamente, nel verificarle su un modello del sistema stesso (di solito una struttura di Kripke finita), tramite *l'enumerazione completa di tutti gli stati raggiungibili*. Questa tecnica è *totalmente automatica* ed ogni volta che viene violata una proprietà desiderata, viene fornito un *controesempio* che illustra un comportamento che falsifica tale proprietà.

Tutte le tecniche del model checking, come la *partial order reduction*, il *symbolic* ed il *bounded* model checking, sono state sviluppate prendendo in considerazione le logiche temporali LTL e CTL, che sono basate su punti. Tuttavia, nonostante l'indubbia espressività di tali logiche, esistono alcune proprietà che potremmo voler verificare, che hanno inerentemente una semantica intervallare e quindi non possono essere espresse da logiche puntuali, per esempio: "la proposizione $p$ deve valere in almeno un dato numero medio di stati del sistema, in un settore di computazione fissato". Le logiche temporali a intervalli entrano in gioco in questi casi: esse adottano gli intervalli, invece dei punti, come loro entità primitive. Tale caratteristica dà loro l'abilità di esprimere proprietà intervallari, come azioni con durata, conseguimenti di obiettivi e aggregazioni temporali, che non possono essere trattate nelle logiche (puntuali) standard.

Una posizione prominente fra le logiche a intervalli è occupata dalla *logica modale degli intervalli temporali di Halpern e Shoham* (HS in breve): essa possiede una modalità per ognuna delle 13 possibili relazioni di ordinamento fra coppie di intervalli, eccetto l'uguaglianza. In questa tesi viene considerato il problema del model checking per HS, il quale ha ricevuto ben poca attenzione in letteratura in confronto al model checking per logiche temporali puntuali.

L'idea è quella di valutare formule di HS su strutture di Kripke finite, per riuscire a verificare la correttezza del comportamento di un sistema rispetto a proprietà intervallari. A questo scopo interpretiamo ogni percorso finito di una struttura di Kripke, detto anche *traccia*, come un intervallo e definiamo l'etichettatura di quest'ultimo sulla base dell'etichettatura degli stati che lo costituiscono. Mostreremo infatti che le strutture di Kripke possono essere mappate in certe strutture intervallari, chiamate *abstract interval models*, sulle quali le formule di HS vengono interpretate; esse hanno in generale un dominio *infinito*, perché le strutture di Kripke possono avere cicli e quindi infinite tracce. Al fine di sviluppare una procedura di model checking per HS su strutture di Kripke finite, proviamo uno *small model theorem* che dimostra che data una formula di HS $\psi$ e una struttura di Kripke finita $\mathcal{K}$, esiste un interval model *finito* che è equivalente a quello indotto da $\mathcal{K}$ rispetto alla soddisfacibilità di $\psi$. In questo modo riusciamo a dimostrare che il problema del model checking per HS interpretato su strutture di Kripke finite è *decidibile* (con un upper bound non elementare); in aggiunta esso è EXPSPACE-hard, se si sfrutta una codifica succinta delle formule di HS.

Poi passiamo a considerare il frammento $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$, costituito dalle formule di HS con le sole modalità $A$, $\overline{A}$, $B$, $\overline{B}$, $\overline{E}$, e dimostriamo che per tale frammento il model checking è in EXPSPACE; sfruttiamo a questo scopo i *rappresentanti di tracce*: essi sono le sole tracce di *lunghezza limitata* che è necessario prendere in considerazione quando si deve verificare una formula di $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ su una struttura di Kripke.

Infine analizziamo i frammenti $HS[A, \overline{A}, \overline{B}, \overline{E}]$ e $\forall HS[A, \overline{A}, B, E]$: il problema del model checking per il primo è PSPACE-completo, coNP-completo per il secondo.

*Ringrazio i professori Angelo Montanari e Adriano Peron per il supporto che non mi hanno mai fatto mancare in questi sei mesi di lavoro.*

*Desidero ringraziare in particolare il prof. Montanari per avermi aiutato in un momento di difficoltà.*

# Contents

# 1 Introduction

## Contents



## 1.1 An overview of model checking and interval temporal logics

IT systems are becoming more and more pervasive in our lives; software is normally responsible for their operation, even in the case of hard-realtime and fault-intolerant systems, such as telephone networks, traffic control systems, medical instruments, e-commerce, ...where reliability is an essential requirement. However, the typical techniques of software engineering for software validation, that is, *simulation* and *testing*, have become outdated and are clearly not sufficient alone in several modern scenarios; their effectiveness decreases dramatically as the complexity of design grows, they do not always guarantee high quality results and often discover errors and unpredictable behaviors in software at late stages of development (or even when it has already been deployed). Moreover, such traditional methods are not effective at discovering the more subtle and hidden bugs.

*Formal verification* is an alternative to simulation and testing, which explores *all the possible states and scenarios of a system*, in order to prove that it features some desired properties such as correctness, deadlock freedom, data integrity, liveness, safety, fairness, responsiveness, interference freedom and so on. The two most famous approaches to formal verification are axiomatic reasoning and model checking.

*Axiomatic reasoning* involves specifying the desired properties of a system by means of *formulas*; then a *proof system,* consisting of axioms and rules, allows to formally prove that the system meets the expected behavior. An example is Hoare's tuple-based proof system. However, this method has several limitations, being the

most significant: the proof rules are designed only for *a posteriori* verification of existing software, not for its systematic development; moreover such a technique is very time consuming, cumbersome, and can be performed only by experts: as a consequence, it is mostly employed for (parts of) critical systems or security protocols.

In *model checking* [CE81, CGP02, QS81, VW86] some properties of a finite-state transition system are expressed in a *temporal logic* and then verified over a model of the system itself (usually a Kripke structure) through exhaustive (implicit or explicit) enumeration of all the reachable states. This technique is *fully automatic* and every time the design violates a desired property, a *counterexample* is produced, which illustrates a behavior that falsifies the property. Model checking techniques allow to analyze even partial specifications, in such a way that it is not necessary to completely specify the system before information can be obtained regarding its correctness.

The first attempt in this direction goes back to 1977, when the use of the linear temporal logic LTL in program verification was proposed by Pnueli [Pnu77]. LTL allows one to reason about changes in the truth value of formulas in a Kripke structure over a linearly-ordered temporal domain, where each moment in time has a unique possible future. More precisely, one has to consider all possible paths in a Kripke structure and to analyze, for each of them, how proposition letters, labeling the states, change from one state to the next one along the path.

The model checking problem for LTL turns out to be PSPACE-complete [CGP02, Pnu81]. This logic has been also investigated with respect to the *satisfiability* problem, useful for example in planning, which is, again, PSPACE-complete.

Four years later, in 1981, Clarke and Emerson invented the branching time logic CTL [CE81], whose model of time is a tree, i.e., the future is not determined, as there are different paths in the future, any one of which may be realized. The model checking problem for CTL is in P, while its satisfiability is EXP-complete. However, these two logics are somewhat complementary, as there are properties expressible in only one of CTL and LTL.

Software is extremely flexible and, as time goes by, it tends to get more and more complex: therefore model checking techniques have to cope with the problem of *state explosion*, which becomes particularly serious when the system being verified is very complex or many components are working in parallel. *Symbolic model checking* [BCM+90, CM90, McM93] allows an exhaustive implicit enumeration of a huge quantity of states (even more than $10^{120}$): ordered binary decision diagrams (OBDDs), particular data structures for representing boolean functions, are exploited in order to get concise representations of transition systems and to efficiently manipulate them. A very successful model checker was developed based on OBDDs: SMV.

*Partial order reduction* [Pel93] tries to reduce the size of the state space by making computations that differ only in the ordering of independently executed actions collapse, as they are indistinguishable by the specification (i.e., they can be considered equivalent) and only one for each group needs to be tested. The model checker SPIN makes use of the partial order reduction technique.

Another, by now traditional model checking method is *bounded model checking* [BCC+03]: proposed in [BCC+99], its basic idea is searching a counterexample in computations whose length is bounded by a fixed integer $k$. So, either a bug is found, or one can increase $k$ until the problem becomes intractable, or the so-called *completeness threshold* is reached (i.e., for high enough values of $k$, this technique is guaranteed to find any existing counterexample). In bounded model checking both the specifications of the system and properties to be checked have to be translated into a propositional formula. In this way, it is possible to employ SAT-solvers in model

checking, which are less sensitive to the state explosion problem than OBDD-based solvers. However this method is in general *incomplete* if the bound is not high enough, hence it used as a complementary technique to OBDD-based symbolic model checking: the former is usually exploited for falsification, i.e., finding counterexamples and bugs, while the latter for verification.

All of these techniques were developed some years ago, bearing in mind the "point-based" temporal logics LTL and CTL. However, while the expressiveness of such logics is beyond doubt, there are some properties we might want to check that are inherently "interval-based" and thus can not be expressed by point-based temporal logics, e.g., "the proposition $p$ has to hold in at least an average number of system states in a given computation sector". Here interval temporal logics (ITLs) come into play, providing an alternative setting for reasoning about time [HS91, Ven90, Ven91]. Such logics deal with intervals, instead of points, as their primitive entities; this choice gives them the ability to express temporal properties, such as actions with duration, accomplishments, and temporal aggregations, which can not be dealt with in standard (point-based) logics.

ITLs have been applied in a variety of computer science fields, including *artificial intelligence* (reasoning about action and change, qualitative reasoning, planning [BT03], configuration and multi-agent systems [GT99, LR06] and computational linguistics [Pra05]), *theoretical computer science* (formal verification [CH04, Mos83]), and *databases* (temporal and spatio-temporal databases [GMS04]).

A prominent position among ITLs is occupied by Halpern and Shoham's modal logic of time intervals (HS, for short) [HS91]. HS features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen's relations [All83]), apart from the equality relation. As an example, the condition: "the current interval meets an interval over which $p$ holds" can be expressed in HS by the formula $\langle A \rangle p$, where $\langle A \rangle$ is the (existential) HS modality for Allen's relation *meet*.

In [HS91], it was shown that the satisfiability problem for HS interpreted over all relevant classes of linear orders is *undecidable*. For example, it is undecidable over $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$, as well as over the classes of all dense, discrete and finite linear orders. Since then, a lot of work has been done on the satisfiability problem for HS fragments [DGMS11], which showed that undecidability rules over them [BDG+14, Lod00, MM14]. As an example, the fragment $HS[B,E]$ (i.e., formulas of HS with $B$ and $E$ modalities only) is undecidable, again, over the class of all linear orders, over all dense linear orders, discrete linear orders and finite linear orders.

However, meaningful exceptions exist, including the *interval logic of temporal neighbourhood* $HS[A,\overline{A}]$ and the *temporal logic of sub-intervals* $HS[D]$ [BGMS10, BGMS09, BMSS11b, MPS10, MMS12, MS12, BMSS11a]. In particular, the former is decidable over the class of all linear orders, and over all finite, discrete and dense linear orders (also over $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$); as for the latter, the situation is more involved: $HS[D]$ is *decidable* over all dense linear orders (e.g., $\mathbb{Q}$), *undecidable* over discrete linear orders and finite linear orders [MM14], and it is not known whether it is decidable or not in the case of all linear orders. Some other fragments, such as $HS[B,\overline{B}]$ and $HS[E,\overline{E}]$, have actually a point-based semantics: one of the endpoints of every interval related to the current one can "move", but the other remains fixed; as a consequence they can be polynomially translated to a basic point-based temporal logic; it follows that they are decidable over the class of all linear orders and, in particular, NP-complete. On the other hand, another fragment, $HS[A,\overline{A},B,\overline{B}]$, is decidable over finite linear orders, $\mathbb{Q}$, as well as over the class of all linear orders, but undecidable over Dedekind-complete linear orders (in particular, $\mathbb{N}$ and $\mathbb{R}$) [MPS14].

Here we focus our attention on the model checking problem for HS, for which a little work has been done, if compared to satisfiability of HS and especially to model checking of point-based temporal logics. The idea is to evaluate HS formulas on finite Kripke structures, making it possible to check the correctness of the behavior of systems with respect to meaningful interval properties. To this end, we interpret each finite path (i.e., a *track*) in a Kripke structure as an interval, and we define its labeling on the basis of the labeling of the states that compose it, according to the *homogeneity assumption* [Roe80] (i.e., a proposition letter $p$ holds on an interval $I$ if and only if $p$ holds on all the subintervals of $I$).

The next section provides an overview of how we tackle the problem of HS model checking by describing the contents of the following chapters.

## 1.2 Organization of the thesis

In Chapter 2, after giving syntax and semantics of HS (Section 2.1), we show that finite Kripke structures can be suitably mapped into interval-based structures, called *abstract interval models*, over which HS formulas can be interpreted (Section 2.2). Such models have, in general, an infinite domain, as finite Kripke structures may have loops and thus an infinite number of tracks. Moreover we introduce *track descriptors*, tree-like structures which give information about (possibly infinite) sets of tracks and allow us to define a *finite-index* equivalence relation over tracks (Section 2.3). The reference for this chapter is [MMPP14].

In Chapter 3, in order to show that the model checking problem for HS over finite Kripke structures is decidable, we prove a small model theorem (Section 3.1), which heavily rests on track descriptors, demonstrating that, given an HS formula $\psi$ and a finite Kripke structure $\mathcal{K}$, there exists a *finite* interval model which is equivalent to the one induced by $\mathcal{K}$ with respect to the satisfiability of $\psi$ (here we follow [MMPP14] again). Then, in Section 3.2, we introduce the novel notion of *correspondence* between descriptors, which allows us to precisely capture the relation of equivalence between tracks with respect to satisfiability of HS formulas. Finally, in Section 3.3, we show that the model checking problem for HS over finite Kripke structures is EXPSPACE-hard, if a succinct encoding of formulas is used.

In Chapter 4, in an attempt to lower the complexity of model checking, we analyze the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ (i.e., formulas of HS with $A$, $\overline{A}$, $B$, $\overline{B}$ and $\overline{E}$ modalities only). In Section 4.1 we present the notions of *cluster* and *descriptor element indistinguishability*, which allow to determine when two tracks are associated with the same descriptor, without directly building it—an operation which is very expensive in terms of complexity, due to the size of such structures. In Section 4.2 *track representatives* are introduced: they are tracks of *bounded length*, each of which is considered in place of all other tracks associated with its descriptor. Due to their finite length, their quantity is finite, and we can exploit them to provide an EXPSPACE model checking algorithm for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$. Finally, in Section 4.4, by rephrasing the proof of Section 3.3, we prove that the model checking problem for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ is NEXP-hard if a suitable succinct encoding of formulas is exploited.

In the last chapter (Chapter 5) we analyze some more fragments of HS of lower complexity, which can still express meaningful properties of transition systems. In Section 5.1 we provide a model checking algorithm for a fragment which turns out to be coNP-complete, $\forall HS[A, \overline{A}, B, E]$ (i.e., formulas of $HS[A, \overline{A}, B, E]$ where only universal modalities and conjunctions are allowed) and in Section 5.2 we study the

PSPACE-complete fragment $HS[A, \overline{A}, \overline{B}, \overline{E}]$.

A final overview of the results reached in this thesis is given in Chapter 6, together with the conclusions.

Finally, the Appendix A contains some proofs which are not directly presented with the corresponding theorems/lemmas, because particularly long and technical.

## 1.3 Related work

As already mentioned, the model checking problem for interval temporal logics has not been extensively studied in literature. Indeed, to our knowledge the only three papers that deal with HS model checking are [MMPP14, LM13, LM14].

In [MMPP14], Montanari et al. give a first characterization of the model checking problem for full HS, interpreted over finite Kripke structures (under the homogeneity assumption). In that paper, the authors provide the basic elements of the general 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 the small model theorem proving the decidability of the model checking problem for full HS against finite Kripke structures. We will present and extend these results in chapters 2 and 3.

In [LM13, LM14], Lomuscio and Michaliszyn address the model checking problem for some fragments of HS extended with epistemic modalities. Their semantic assumptions differ from those made in [MMPP14], making it difficult to compare the outcomes of the two research directions. In both cases, formulas of interval temporal logic are evaluated over tracks obtained from the unravelling of a finite Kripke structure. However, in [MMPP14] the authors state that a proposition letter holds over a track if and only if it holds over all its states (homogeneity principle), while in [LM13, LM14] truth of proposition letters is defined over pairs of states (the endpoints of tracks).

In [LM13], the authors focus their attention on the fragment $HS[B, E, D]$ (since modality $D$ is easily definable in terms of modalities $B$ and $E$, $HS[B, E, D]$ is actually as expressive as $HS[B, E]$), extended with epistemic modalities. They consider a restricted form of model checking, which verifies the given specification against a single (finite) initial computation interval. Their goal is indeed to reason about a given computation of a multi-agent system, rather than on all its admissible computations. The authors prove that the considered model checking problem is PSPACE-complete. Moreover, they show that the same problem restricted to the purely temporal fragment $HS[B, E, D]$, that is, the one obtained by removing epistemic modalities, is in P. These results do not come as a surprise as they trade expressiveness for efficiency: modalities $B$ 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 [LM14], they show that the picture drastically changes with other fragments of HS, that allow one to access infinitely many tracks/intervals. In particular, they prove that the model checking problem for the fragment $HS[A, \overline{B}, L]$ (since modality $L$ is easily definable in terms of modality $A$, $HS[A, \overline{B}, L]$ is actually as expressive as $HS[A, \overline{B}]$), extended with epistemic modalities, is decidable with a non-elementary upper bound. Notice that, thanks to modalities $A$ and $\overline{B}$, formulas of this logic can possibly refer to infinitely many (future) tracks/intervals.

# 2

# HS, Abstract Interval Models and Descriptors

## Contents



In this chapter, we first give syntax and semantics of Halpern and Shoham's interval temporal logic HS with respect to abstract interval models. Then, we provide a suitable mapping from Kripke structures to abstract interval models that allows us to interpret HS formulas over Kripke structures and then to define the notion of interval-based model checking.

Finally we introduce the notion of *track descriptor*, which allows to prove that, considered a Kripke structure $\mathcal{K}$ and an HS formula $\psi$, there exists a *finite* abstract interval model which is equivalent to the possibly *infinite* model corresponding to $\mathcal{K}$ with respect to satisfiability of $\psi$.

## 2.1 The interval temporal logic HS

An interval algebra to reason about intervals and their relative order was first proposed by Allen in [All83]; then, a systematic logical study of interval representation and reasoning was done by Halpern and Shoham, who introduced the interval temporal logic HS [HS91]; such a logic features one modality for each of the possible binary ordering relations between a pair of intervals (the so-called "Allen's relations"), except for the equality.

Table 2.1 depicts 6 of the 13 Allen's relations; the other 7 are the 6 inverse relations (given a generic binary relation $\mathcal{R}$, the inverse relation $\overline{\mathcal{R}}$ holds between two elements, $b\overline{\mathcal{R}}a$, if and only if $a\mathcal{R}b$), and the equality.

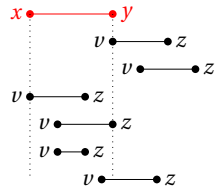
| Allen relation | HS modality | Definition w.r.t. interval structures | Example |
|---|---|---|---|
| *meets* | $\langle A \rangle$ | $[x,y]\mathcal{R}_A[v,z] \iff y = v$ |  |
| *before* | $\langle L \rangle$ | $[x,y]\mathcal{R}_L[v,z] \iff y < v$ | |
| *started-by* | $\langle B \rangle$ | $[x,y]\mathcal{R}_B[v,z] \iff x = v \wedge z < y$ | |
| *finished-by* | $\langle E \rangle$ | $[x,y]\mathcal{R}_E[v,z] \iff y = z \wedge x < v$ | |
| *contains* | $\langle D \rangle$ | $[x,y]\mathcal{R}_D[v,z] \iff x < v \wedge z < y$ | |
| *overlaps* | $\langle O \rangle$ | $[x,y]\mathcal{R}_O[v,z] \iff x < v < y < z$ | |



**Table 2.1:** Allen's interval relations and corresponding HS modalities.

In the table, each Allen's relation is shown together with the corresponding HS (existential) modality. In its original formulation, HS allowed point intervals as well, that is, intervals consisting of a single point, but that way HS modalities are neither *mutually exclusive* nor *jointly exhaustive*, i.e., more than one relation, or even none, may hold between any two intervals. In the following, we will consider only strict intervals, consisting of two or more points (*strict semantics*).

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, $\langle A \rangle$, $\langle L \rangle$, $\langle B \rangle$, $\langle E \rangle$, $\langle D \rangle$, $\langle O \rangle$, $\langle \overline{A} \rangle$, $\langle \overline{L} \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$, $\langle \overline{D} \rangle$, and $\langle \overline{O} \rangle$.

Formally, HS formulas are defined by the following grammar:

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

In the following, we will make use of the standard abbreviations of propositional logic, e.g., we will write $\psi \vee \phi$ for $\neg\psi \wedge \neg\phi$, $\psi \rightarrow \phi$ for $\neg\psi \vee \phi$, and $\psi \leftrightarrow \phi$ for $(\psi \rightarrow \phi) \wedge (\phi \rightarrow \psi)$. Moreover, for all $X$, dual universal modalities $[X]\psi$ and $[\overline{X}]\psi$ are respectively defined as $\neg\langle X \rangle \neg\psi$ and $\neg\langle \overline{X} \rangle \neg\psi$, as usual.

Finally, it can easily be shown that, when the strict semantics is assumed, all HS modalities can be expressed in terms of modalities $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$, and the transposed modalities $\langle \overline{A} \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$ as follows:

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

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. As an example, the fragment $HS[A, \overline{A}, B, \overline{B}, E, \overline{E}]$ features modalities $\langle A \rangle, \langle \overline{A} \rangle, \langle B \rangle, \langle \overline{B} \rangle, \langle E \rangle$, and $\langle \overline{E} \rangle$ only (observe that this fragment contains an equivalent formula for every HS formula).

HS can be viewed as a multi-modal logic with six primitive modalities, namely, $\langle A \rangle$, $\langle B \rangle$, $\langle E \rangle$, 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 2.1.** *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 $\sigma$ 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 induction on the structural complexity of 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 $\mathcal{A}, I \models \psi$;
- $\mathcal{A}, I \models \psi \wedge \phi$ iff $\mathcal{A}, I \models \psi$ and $\mathcal{A}, I \models \phi$;
- $\mathcal{A}, I \models \langle X \rangle \psi$, for $X \in \{A, B, E\}$, iff there exists $J \in \mathbb{I}$ such that $I \, X_{\mathbb{I}} \, J$ and $\mathcal{A}, J \models \psi$;
- $\mathcal{A}, I \models \langle \overline{X} \rangle \psi$, for $\overline{X} \in \{\overline{A}, \overline{B}, \overline{E}\}$, iff there exists $J \in \mathbb{I}$ such that $J \, X_{\mathbb{I}} \, I$ and $\mathcal{A}, J \models \psi$.

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

## 2.2 Finite Kripke structures and abstract interval models

Finite-state transition systems are usually modeled as finite Kripke structures. In the following, we first recall the definition of finite Kripke structure and then we define a suitable mapping from this class of structures to abstract interval models that makes it possible to specify properties of systems by means of HS formulas.

**Definition 2.2.** *A finite Kripke structure is a tuple* $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$*, where* $\mathcal{AP}$ *is a set of proposition letters,* $W$ *is a finite set of states (*worlds*),* $\delta \subseteq W \times W$ *is a left-total relation between pairs of states (*accessibility *relation),* $\mu : W \mapsto 2^{\mathcal{AP}}$ *a total labeling 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 $\delta$ is the transition relation that constrains the evolution of the system over time.

A simple Kripke structure, consisting of two states only, is reported in the following example. We will use it as a running example in this chapter.

**Example 2.3.** *Figure 2.1 depicts a two-state Kripke structure* $\mathcal{K}_{Equiv}$ *(the initial state is identified by a double circle). Despite its simplicity, it features an infinite number of different (finite) paths. 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* $\mu(v_1) = \{q\}$.

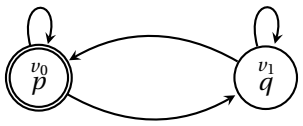



**Figure 2.1:** The Kripke structure $\mathcal{K}_{Equiv}$.

**Definition 2.4.** *A track $\rho$ 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 $\mathrm{lst}(\rho) = v_n$;
- $\rho(i, j) = v_i \cdots v_j$, $0 \leq i \leq j \leq |\rho| - 1$ is a subtrack of $\rho$;
- $\mathrm{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper prefixes of $\rho$;
- $\mathrm{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper suffixes of $\rho$.

If $\mathrm{fst}(\rho) = w_0$, where $w_0$ is the initial state of $\mathcal{K}$, $\rho$ is said to be an *initial track*. Notice that the length of tracks, prefixes, and suffixes is greater than 1, as they will be mapped into strict intervals.

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 2.5.** *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:*

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

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

In Definition 2.5, 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 $\sigma$, 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 $\rho$. This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval if and only if it holds over all of its subintervals.

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

**Definition 2.6.** *(Satisfiability of HS formulas over Kripke structures) Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ be a track in* $\mathrm{Trk}_{\mathcal{K}}$*, $\psi$ be an HS formula. We say that the pair $(\mathcal{K}, \rho)$ satisfies $\psi$, 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 2.7.** *(Model checking) Let $\mathcal{K}$ be a finite Kripke structure and $\psi$ be an HS formula. We say that $\mathcal{K}$ models $\psi$, denoted by $\mathcal{K} \models \psi$, if and only if*

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

We conclude the section by giving 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 $\rho$ with $|\rho| = 2$, independently of $\mathcal{K}$, it holds that $\mathcal{K}, \rho \models [B]\bot$, because $\rho$ 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 $\langle \mathrm{B} \rangle$ can indeed 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 $|\rho| = k$.

**Example 2.8.** *Let us consider the finite Kripke structure $\mathcal{K}_{Equiv}$ of Example 2.3, depicted in Figure 2.1. For the sake of brevity, for any track $\rho$, we denote by $\rho^n$ the track obtained by concatenating $n$ copies of $\rho$. The truth of the following statements can be easily checked:*

- $\mathcal{K}_{Equiv}, (v_0 v_1)^2 \models \langle \mathrm{A} \rangle\, q$;
- $\mathcal{K}_{Equiv}, v_0 v_1 v_0 \not\models \langle \mathrm{A} \rangle\, q$;
- $\mathcal{K}_{Equiv}, (v_0 v_1)^2 \models \langle \overline{\mathrm{A}} \rangle\, p$;
- $\mathcal{K}_{Equiv}, v_1 v_0 v_1 \not\models \langle \overline{\mathrm{A}} \rangle\, p$.

*The above statements show that 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.*

*Modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ can be exploited to distinguish between tracks encompassing a different number of iterations of a given loop. This is the case, for instance, with the following statements:*

- $\mathcal{K}_{Equiv}, (v_1 v_0)^3 v_1 \models \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle\, p \big) \big)$;
- $\mathcal{K}_{Equiv}, (v_1 v_0)^2 v_1 \not\models \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle\, p \big) \big)$.

*Finally, HS makes it possible to distinguish between* $\rho_1 = v_0^3 v_1 v_0$ *and* $\rho_2 = v_0 v_1 v_0^3$*, which feature the same number of iterations of the same loops, but differ in the order of loop occurrences:*

$$\mathcal{K}_{Equiv}, \rho_1 \models \langle \mathrm{B} \rangle \left( \langle \mathrm{A} \rangle q \wedge \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle p \right) \text{ but } \mathcal{K}_{Equiv}, \rho_2 \not\models \langle \mathrm{B} \rangle \left( \langle \mathrm{A} \rangle q \wedge \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle p \right).$$

**Example 2.9.** *In Figure 2.2, we provide an example of a finite Kripke structure* $\mathcal{K}_{Sched}$ *that models the behavior of a scheduler serving three processes which are continuously requesting the use of a common resource. The initial state is* $v_0$*: no process is served in that state. In any other state* $v_i$ *and* $\overline{v}_i$*, with* $i \in \{1,2,3\}$*, the* $i$*-th process is served (this is denoted by the fact that* $p_i$ *holds in those states). For the sake of readability, edges are marked either by* $r_i$*, for* $request(i)$*, or by* $u_i$*, for* $unlock(i)$*. Edge labels do not have a semantic value, that is, they are neither part of the structure definition, nor proposition letters; they are simply used to ease reference to edges. Process* $i$ *is served in state* $v_i$*, then, after "some time", a transition* $u_i$ *from* $v_i$ *to* $\overline{v}_i$ *is taken; subsequently, process* $i$ *cannot be served again immediately, as* $v_i$ *is not directly reachable from* $\overline{v}_i$ *(the scheduler cannot serve twice the same process in two successive rounds). A transition* $r_j$*, with* $j \neq i$*, from* $\overline{v}_i$ *to* $v_j$ *is then taken and process* $j$ *is served. This structure can easily be generalized to a higher number of processes.*

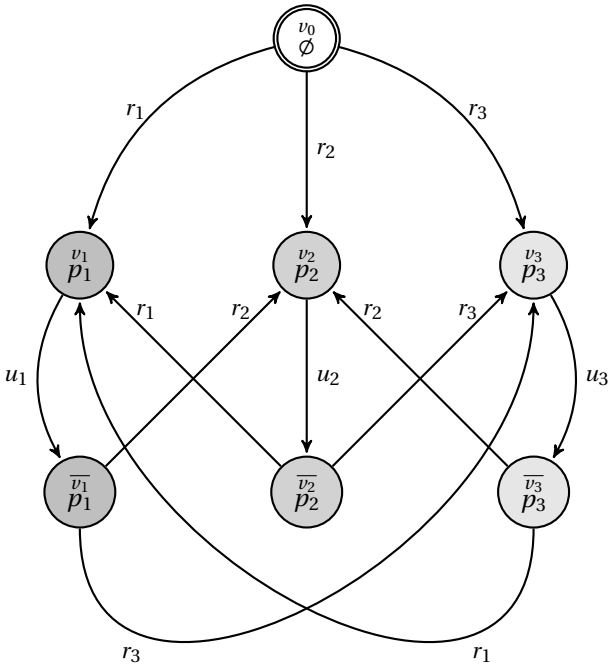



**Figure 2.2:** The Kripke structure $\mathcal{K}_{Sched}$.

*We show how some meaningful properties to be checked over* $\mathcal{K}_{Sched}$ *can be expressed in HS. In all formulas, we force the validity of the considered property over all legal computation sub-intervals by using modality* $[E]$ *(all computation sub-intervals are suffixes of at least one initial track). Moreover, we will make use of the shorthand* $wit_{\geq 2}(\{p_1, p_2, p_3\})$ *for the formula:*

$$(\langle \mathrm{D} \rangle p_1 \wedge \langle \mathrm{D} \rangle p_2) \vee (\langle \mathrm{D} \rangle p_1 \wedge \langle \mathrm{D} \rangle p_3) \vee (\langle \mathrm{D} \rangle p_2 \wedge \langle \mathrm{D} \rangle p_3),$$

*which states that there exist at least two sub-intervals such that $p_i$ holds over the former and $p_j$ over the latter, with $i, j \in \{1,2,3\}$ and $j \neq i$ (such a formula can be easily generalized to an arbitrary set of proposition letters and to any natural number $k$). The truth of the following statements can be easily checked:*

- $\mathcal{K}_{Sched} \models [E]\left(\langle \mathrm{B}\rangle^5 \top \rightarrow wit_{\geq 2}(\{p_1, p_2, p_3\})\right)$;
- $\mathcal{K}_{Sched} \not\models [E]\left(\langle \mathrm{B}\rangle^{10} \top \rightarrow \langle \mathrm{D}\rangle\, p_3\right)$;
- $\mathcal{K}_{Sched} \not\models [E]\left(\langle \mathrm{B}\rangle^7 \top \rightarrow \langle \mathrm{D}\rangle\, p_1 \wedge \langle \mathrm{D}\rangle\, p_2 \wedge \langle \mathrm{D}\rangle\, p_3\right)$.

*The first formula states that in any suffix of an initial track of length greater than or equal to 7 at least 2 proposition letters are witnessed. $\mathcal{K}_{Sched}$ satisfies the formula since a process cannot be executed twice in a row. The second formula states that in any suffix of an initial track of length at least 12 process 3 is executed at least once in some internal states. $\mathcal{K}_{Sched}$ does not satisfy the formula since the scheduler can avoid executing a process ad libitum. The third formula states that in any suffix of an initial track of length greater than or equal to 9, $p_1, p_2, p_3$ are all witnessed. The only way to satisfy this property is to constrain the scheduler to execute the three processes in a strictly periodic manner, but this is not the case.*

## 2.3 The fundamental notion of $BE_k$-descriptor

In the previous section, we have shown that, 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 $\mathcal{K}$ has loops (each state must have at least one successor), the number of its tracks, and thus the number of intervals of $\mathcal{A}_{\mathcal{K}}$, is infinite. In this section, we show that, given a finite Kripke structure $\mathcal{K}$ and an HS formula $\varphi$, we can build a *finite* abstract interval model, which—as we will prove in Chapter 3—is equivalent to $\mathcal{A}_{\mathcal{K}}$ with respect to the satisfiability of $\varphi$ (in fact, of a class of HS formulas including $\varphi$).

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

**Definition 2.10.** *Let $\psi$ be an HS formula. The* BE-nesting depth of $\psi$*, denoted by* $\mathrm{Nest}_{\mathrm{BE}}(\psi)$*, is defined by induction on the structural complexity of the formula as follows:*

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

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

**Definition 2.11.** *Let $\mathcal{K}$ be a finite Kripke structure and $\rho$ and $\rho'$ be two tracks in* $\mathrm{Trk}_{\mathcal{K}}$*. We say that $\rho$ and $\rho'$ are $k$-equivalent if and only if, for every HS-formula $\psi$ with* $\mathrm{Nest}_{\mathrm{BE}}(\psi) = k$*, $\mathcal{K}, \rho \models \psi$ if and only if $\mathcal{K}, \rho' \models \psi$.*

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

**Proposition 2.12.** *Let $\mathcal{K}$ be a finite Kripke structure and $\rho$ and $\rho'$ be two tracks in* $\mathrm{Trk}_{\mathcal{K}}$. *If $\rho$ and $\rho'$ are $k$-equivalent, then they are $h$-equivalent, for all $0 \leq h \leq k$.*

*Proof.* Let us assume that $\mathcal{K}, \rho \models \psi$, with $0 \leq \mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$. Consider the formula $\langle \mathrm{B} \rangle^k \top$, whose BE-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 $\mathrm{Nest}_{\mathrm{BE}}\left(\langle \mathrm{B} \rangle^k \top \wedge \psi\right) = k$, from the hypothesis it 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. □

We are now ready to introduce the notion of *descriptor*, which will play a fundamental role in the definition of finite abstract interval models.

**Definition 2.13.** *A $B$-descriptor (resp., $E$-descriptor) is a labeled tree $\mathcal{D} = (V, E, \lambda)$, 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 labeling function, that satisfies the following conditions:*

1. *for all $(v, v') \in E$, with $\lambda(v) = (v_{in}, S, v_{fin})$ and $\lambda(v') = (v'_{in}, S', v'_{fin})$, it holds that $S' \subseteq S$, $v_{in} = v'_{in}$, and $v'_{fin} \in S$ (resp., $S' \subseteq S$, $v_{fin} = v'_{fin}$, and $v'_{in} \in S$);*
2. *for all pairs of edges $(v, v'), (v, v'') \in E$, 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 (2) of Definition 2.13 simply states that no two subtrees, whose roots are siblings, can be isomorphic.

For $X \in \{B, E\}$, the *depth* of an $X$-descriptor $(V, E, \lambda)$ is the depth of the tree $(V, E)$. We call an $X$-descriptor of depth $k \in \mathbb{N}$ an $X_k$-descriptor. An $X_0$-descriptor $\mathcal{D}$ 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*. Hereafter, two descriptors will be considered *equal up to isomorphism*. The following lemma holds.

**Lemma 2.14.** *For all $k \in \mathbb{N}$, there exists a finite number of possible $B_k$-descriptors (resp., $E_k$-descriptors).*

*Proof.* Let us consider the case of $B_k$-descriptors (the case of $E_k$-descriptors is analogous). 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 2.14 provides an upper bound to the number of distinct $B_k$-descriptors (resp., $E_k$-descr.), and thus to the number of nodes of each $B_{k+1}$-descriptor (resp., $E_{k+1}$-descriptors), for $k \in \mathbb{N}$, which is *not* elementary with respect to $|W|$ and $k$, $|W|$ being the exponent and $k$ the height of the exponential tower. 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.

We show now how $B$-descriptors and $E$-descriptors can be exploited to extract relevant information from the tracks of a finite Kripke structure to be used in model

checking. Let $\mathcal{K}$ be a finite Kripke structure and $\rho$ be a track in $\mathrm{Trk}_{\mathcal{K}}$. We now describe how to build such descriptors for $\rho$.

For any $k \geq 0$, the label of the root of both the $B_k$-descriptor and $E_k$-descriptor for $\rho$ is the triple $(\mathrm{fst}(\rho), \mathrm{intstates}(\rho), \mathrm{lst}(\rho))$. The root of the $B_k$-descriptor has a child for each prefix $\rho'$ of $\rho$, labeled with $(\mathrm{fst}(\rho'), \mathrm{intstates}(\rho'), \mathrm{lst}(\rho'))$. 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. The $E_k$-descriptor is built in a similar way by considering the suffixes of $\rho$.

In general, $B$- and $E$-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 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;
- to deal with $\langle \mathrm{E} \rangle \psi$ formulas, the $E$-descriptor keeps information about all the suffixes of the track;
- no additional information is needed for $\langle \overline{\mathrm{B}} \rangle \psi$ and $\langle \overline{\mathrm{E}} \rangle \psi$ formulas.

Let $\mathcal{K}$ be a finite Kripke structure. The $B_k$-descriptor (resp., $E_k$-descriptor) for a track $\rho$ in $\mathrm{Trk}_{\mathcal{K}}$ is formally defined as follows.

**Definition 2.15.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ be a track in $\mathrm{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. The $B_k$-descriptor (respectively, $E_k$-descriptor) for $\rho$ is inductively defined as follows:*

- *for $k = 0$, the $B_k$-descriptor (respectively, $E_k$-descriptor) for $\rho$ 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 (respectively, $E_k$-descriptor) for $\rho$ 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 (respectively, suffix) $\rho'$ of $\rho$, 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 (respectively, $E_{k-1}$-descriptor) for $\rho'$.*
2. *for each vertex $v \in V$ such that $(\mathrm{root}(\mathcal{D}), v) \in E$, there exists a prefix (respectively, suffix) $\rho'$ of $\rho$ such that the subtree rooted in $v$ is the $B_{k-1}$-descriptor (respectively, $E_{k-1}$-descriptor) for $\rho'$;*

3. *for all pairs of edges* $(\text{root}(\mathcal{D}), v'), (\text{root}(\mathcal{D}), v'') \in E$*, if the subtree rooted in* $v'$ *is isomorphic to the subtree rooted in* $v''$*, then* $v' = v''$*.*

It can be easily checked that any $B_k$-descriptor (resp., $E_k$-descriptor) for some track of some finite Kripke structure satisfies the conditions of Definition 2.13 (in particular, condition (1)), but not vice versa.

Consider, for instance, the $B_1$-descriptor reported in Figure 2.3. It is built on a set of states $W$ including at least states $v_0, v_1, v_2$, and $v_3$, and it satisfies both conditions of Definition 2.13. However, no track of a finite Kripke structure can be described by it, as no track may feature two prefixes to associate with the first two children of the root.

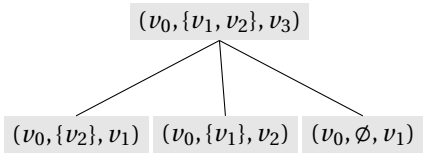
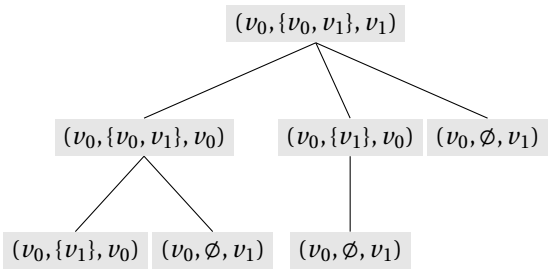
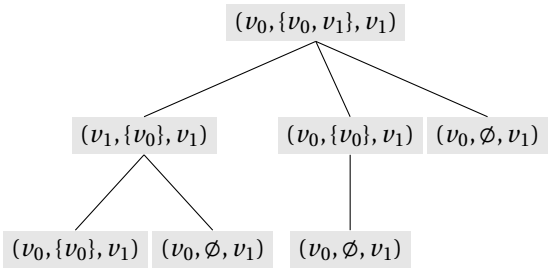



**Figure 2.3:** An example of a $B_1$-descriptor devoid of a corresponding track (in any Kripke structure).

**Example 2.16.** *Figure 2.4 depicts the* $B_2$*- and* $E_2$*-descriptors for the track* $v_0 v_1 v_0 v_0 v_1$ *of the Kripke structure* $\mathcal{K}_{Equiv}$ *of Figure 2.1.*




**(a)** $B_2$-descriptor for the track $v_0 v_1 v_0 v_0 v_1$ of $\mathcal{K}_{Equiv}$.




**(b)** $E_2$-descriptor for the track $v_0 v_1 v_0 v_0 v_1$ of $\mathcal{K}_{Equiv}$.

**Figure 2.4:** $B_2$- and $E_2$-descriptors for the track $v_0 v_1 v_0 v_0 v_1$ of $\mathcal{K}_{Equiv}$.

**Example 2.17.** *In Figure 2.5, we show the* $B_2$*-descriptor for* $\rho = v_0 v_1 v_0 v_0 v_0 v_0 v_1$ *of* $\mathcal{K}_{Equiv}$*. It is worth noticing that there exist two distinct prefixes of* $\rho$*, that is, the tracks*

*$\rho' = v_0v_1v_0v_0v_0v_0$ and $\rho'' = v_0v_1v_0v_0v_0$, which have the same $B_1$-descriptor. Since, according to Definition 2.13, 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 $\rho$ prefixes $\rho'$ and $\rho''$ 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.*

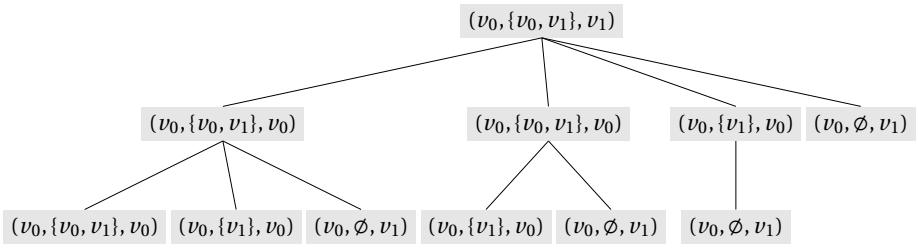
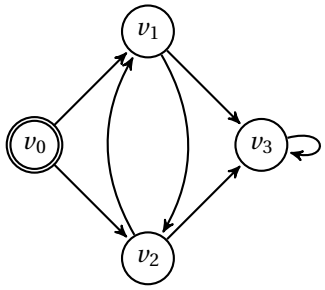
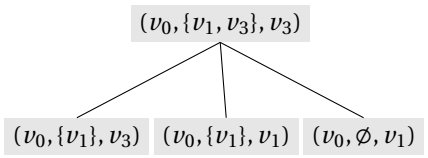



**Figure 2.5:** The $B_2$-descriptor for the track $v_0v_1v_0v_0v_0v_0v_1$ of $\mathcal{K}_{Equiv}$.

**Example 2.18.** *This example shows that not all of the $B_k$-descriptors that can be generated from the set of states of a given finite Kripke structure are $B_k$-descriptors for some track of that structure. (The same holds for $E_k$-descriptors.) Let us consider the finite Kripke structure $\mathcal{K}$ and the $B_1$-descriptor $\mathcal{D}_{B_1}$ respectively depicted on the left and the right of Figure 2.6. By inspecting $\mathcal{D}_{B_1}$, it can be easily checked that it can be the $B_1$-descriptor for tracks of the form $v_0v_1^hv_3^2$, with $h \geq 2$, only. However, no track of this form can be obtained from the unravelling of $\mathcal{K}$.*




**(a)** A Kripke structure $\mathcal{K}$.




**(b)** $\mathcal{D}_{B_1}$: a $B_1$-descriptor not corresponding to any of the tracks of $\mathcal{K}$ in figure 2.6(a).

**Figure 2.6:** Not all of the $B_k$-descriptors over $W$ are descriptors for some track of the Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$.

To check an HS formula against a given finite Kripke structure we actually need to account for both the *started-by* and *finished-by* relations at the same time. To this end, we introduce $BE_k$-descriptors for tracks. Given a finite Kripke structure $\mathcal{K}$ and a track $\rho$ in $\mathrm{Trk}_{\mathcal{K}}$, the $BE_k$-descriptor for $\rho$ can be obtained from a suitable merging of its $B_k$-descriptor and $E_k$-descriptor. It can be viewed as a sort of "product" of the $B_k$-descriptor and the $E_k$-descriptor for $\rho$, and it is formally defined as follows:

**Definition 2.19.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ be a track in $\mathrm{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. The $BE_k$-descriptor for $\rho$ is a labeled tree $\mathcal{D} = (V, E, \lambda)$, where $V$ is a finite set of vertices,*

*$E = E_B \cup E_E$, with $E_B \subseteq V \times V$ the set of "B-edges", $E_E \subseteq V \times V$ the set of "E-edges", and $E_B \cap E_E = \emptyset$, and $\lambda : V \mapsto W \times 2^W \times W$, which is inductively defined on $k \in \mathbb{N}$ as follows:*

- *for $k = 0$, the $BE_k$-descriptor for $\rho$ is $\mathcal{D} = (\text{root}(\mathcal{D}), \emptyset, \lambda)$, where*

$$\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho)).$$

- *for $k > 0$, the $BE_k$-descriptor for $\rho$ is $\mathcal{D} = (V, E, \lambda)$ with*

$$\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$$

  *which satisfies the following conditions:*

  *1a. for each prefix $\rho'$ of $\rho$, there exists $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E_B$ and the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\rho'$;*

  *1b. for each vertex $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E_B$, there exists a prefix $\rho'$ of $\rho$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\rho'$;*

  *1c. for all pairs of edges $(\text{root}(\mathcal{D}), v'), (\text{root}(\mathcal{D}), v'') \in E_B$, if the subtree rooted in $v'$ is isomorphic to the subtree rooted in $v''$, then $v' = v''$;*

  *2a. for each suffix $\rho''$ of $\rho$, there exists $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E_E$ and the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\rho''$;*

  *2b. for each vertex $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E_E$, there exists a suffix $\rho''$ of $\rho$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\rho''$;*

  *2c. for all pairs of edges $(\text{root}(\mathcal{D}), v'), (\text{root}(\mathcal{D}), v'') \in E_E$, if the subtree rooted in $v'$ is isomorphic to the subtree rooted in $v''$, then $v' = v''$.*

From Definition 2.19, it follows that for all $(v, v') \in E_B$, with $\lambda(v) = (v_{in}, S, v_{fin})$ and $\lambda(v') = (v'_{in}, S', v'_{fin})$, $S' \subseteq S$, $v_{in} = v'_{in}$, and $v'_{fin} \in S$, and for all $(v, v') \in E_E$, with $\lambda(v) = (v_{in}, S, v_{fin})$ and $\lambda(v') = (v'_{in}, S', v'_{fin})$, $S' \subseteq S$, $v_{fin} = v'_{fin}$ and $v'_{in} \in S$.

**Example 2.20.** *In Figure 2.7, with reference to the finite Krikpe structure $\mathcal{K}_{Equiv}$ of Figure 2.1, we give an example of a $BE_2$-descriptor. B-edges are represented by solid lines, while E-edges are represented by dashed lines. It is worth pointing out that the $BE_2$-descriptor of Figure 2.7 turns out to be the $BE_2$-descriptor for both the track $\rho = v_0 v_1 v_0^3 v_1$ and the track $\rho' = v_0 v_1 v_0^4 v_1$ (and many others). As we will see very soon, this is not an exception, but the rule: different tracks of a finite Kripke structure are described by the same BE-descriptor. Notice also that it features two isomorphic subtrees for the same node (the root). They both consist of a single node, labeled with $(v_0, \emptyset, v_1)$. However, this does not violate Definition 2.19 since one of them is connected to the parent via a B-edge and the other via an E-edge.*

$B_k$ and $E_k$-descriptors can be easily recovered from $BE_k$ ones. The $B_k$-descriptor $\mathcal{D}_{B_k}$ for a track $\rho$ can be obtained from the $BE_k$-descriptor $\mathcal{D}_{BE_k}$ for $\rho$ by pruning it in such a way that only those vertices of $\mathcal{D}_{BE_k}$ which are connected to the root via paths consisting of $B$-edges only are maintained (the set of edges of $\mathcal{D}_{B_k}$ and its labeling function can be obtained by restricting those of $\mathcal{D}_{BE_k}$ to the nodes of $\mathcal{D}_{B_k}$). The $E_k$-descriptor $\mathcal{D}_{E_k}$ of $\rho$ can be obtained in a similar way.

We focus now our attention on the relationships between the tracks obtained from the unravelling of a finite Kripke structure and their $BE_k$-descriptors. A key observation is that, even though the number of tracks of a finite Kripke structure $\mathcal{K}$

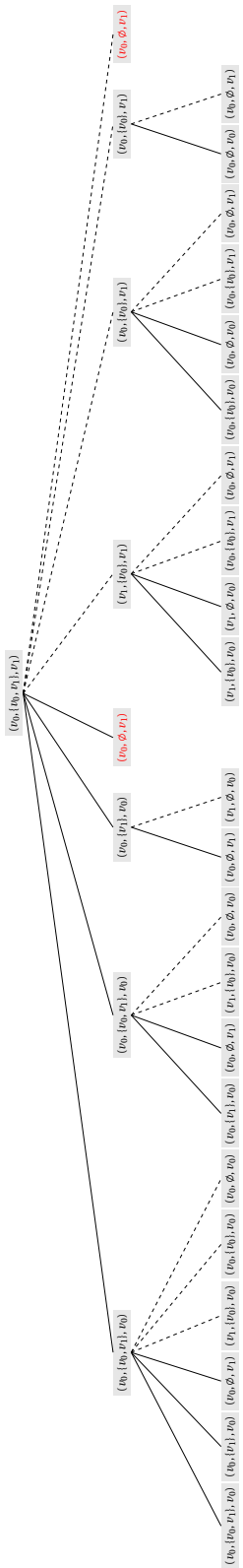


**Figure 2.7:** An example of $BE_2$-descriptor.

is infinite, for any $k \in \mathbb{N}$ the set of $BE_k$-descriptors for its tracks is finite. This is an immediate consequence of Definition 2.19 and Lemma 2.14. Thus, at least one $BE_k$-descriptor must be the $BE_k$-descriptor for infinitely many tracks. $BE_k$-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 2.21.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho, \rho'$ be two tracks in* $\mathrm{Trk}_{\mathcal{K}}$*, and $k \in \mathbb{N}$. We say that $\rho$ and $\rho'$ are* $k$-descriptor equivalent*, denoted by $\rho \sim_k \rho'$, if and only if the $BE_k$-descriptors for $\rho$ and $\rho'$ coincide.*

The equivalence class of a track $\rho$ will be denoted by $[\rho]_{\sim_k}$. In the next chapter (in Theorem 3.2), we will prove that, for any given pair of tracks $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$, if $\rho \sim_k \rho'$, then $\rho$ and $\rho'$ are $k$-equivalent (see Definition 2.11).

For all $k \in \mathbb{N}$, by exploiting the fact that the set of $BE_k$-descriptors for the tracks of a finite Kripke structure $\mathcal{K}$ is finite (or, equivalently, $\sim_k$ has a finite index), we can associate a *finite* abstract interval model with $\mathcal{K}$, called the *quotient induced abstract interval model of depth $k$*, as follows.

Let $\mathcal{K}$ be a finite Kripke structure, $\mathrm{Trk}_{\mathcal{K}}$ be the set of all its tracks, and $k \in \mathbb{N}$. Each class of $\sim_k$ is identified by a $BE_k$-descriptor $\mathcal{D}_{BE_k}$, and it consists of all and only those tracks in $\mathrm{Trk}_{\mathcal{K}}$ which have $\mathcal{D}_{BE_k}$ as their $BE_k$-descriptor. We denote by $k$-Desc the set of all $BE_k$-descriptors $\mathcal{D}_{BE_k}$ such that there exists at least one track $\rho$ in $\mathrm{Trk}_{\mathcal{K}}$ which is described by $\mathcal{D}_{BE_k}$ (we say that $\mathcal{D}_{BE_k}$ is witnessed by a track in $\mathrm{Trk}_{\mathcal{K}}$).

Allen's relations $A$ (*meets*), $B$ (*started-by*), and $E$ (*finished-by*) over $k$-Desc can be defined as follows.

**Definition 2.22.** *Let $\mathcal{D}_{BE_k}, \mathcal{D}'_{BE_k}$ be two $BE_k$-descriptors in $k$-*Desc*, with*

$$\mathcal{D}_{BE_k} = (V, E_B \cup E_E, \lambda) \text{ and } \mathcal{D}'_{BE_k} = (V', E'_B \cup E'_E, \lambda').$$

*We say that:*

1. $\left(\mathcal{D}_{BE_k}, \mathcal{D}'_{BE_k}\right) \in A_{\mathrm{Desc}}$ *iff*

$$\lambda\left(\mathrm{root}(\mathcal{D}_{BE_k})\right) = (v_{in}, S, v_{fin}), \lambda'\left(\mathrm{root}(\mathcal{D}'_{BE_k})\right) = (v'_{in}, S', v'_{fin}), \text{ and } v_{fin} = v'_{in};$$

2. $\left(\mathcal{D}_{BE_k}, \mathcal{D}'_{BE_k}\right) \in B_{\mathrm{Desc}}$ *iff there exists $v \in V$ such that $\left(\mathrm{root}(\mathcal{D}_{BE_k}), v\right) \in E_B$ and the subtree of $\mathcal{D}_{BE_k}$ rooted in $v$ is isomorphic to the tree obtained from $\mathcal{D}'_{BE_k}$ by removing the nodes at depth $k$ (if any) and the isomorphic subtrees possibly resulting from such a removal (see condition (1c) of Definition 2.19);*

3. $\left(\mathcal{D}_{BE_k}, \mathcal{D}'_{BE_k}\right) \in E_{\mathrm{Desc}}$ *iff there exists $v \in V$ such that $\left(\mathrm{root}(\mathcal{D}_{BE_k}), v\right) \in E_E$ and the subtree of $\mathcal{D}_{BE_k}$ rooted in $v$ is isomorphic to the tree obtained from $\mathcal{D}'_{BE_k}$ by removing the nodes at depth $k$ (if any) and the isomorphic subtrees possibly resulting from such a removal (see condition (2c) of Definition 2.19).*

Definition 2.22 can be read as follows. Item 1 states that, whenever the third component (final state) of the label of the root of a $BE_k$-descriptor is equal to the first component (initial state) of the label of the root of another $BE_k$-descriptor, the two $BE_k$-descriptor are related by $A_{\mathrm{Desc}}$. This amounts to say that any pair of tracks $\rho, \rho'$, which are described respectively by the former and latter $BE_k$-descriptor, are such that $\mathrm{lst}(\rho) = \mathrm{fst}(\rho')$, and thus Allen relation $A$ holds between $\rho$ and $\rho'$. Item 2 states

that, whenever there exists a subtree of $\mathcal{D}_{BE_k}$, linked to the root via a $B$-edge, which is isomorphic to the tree obtained from $\mathcal{D}'_{BE_k}$ by removing the nodes at depth $k$ (if any) and the isomorphic subtrees possibly resulting from such a removal (this is the case, for instance, with subtrees of $\mathcal{D}'_{BE_k}$ that differ on the labels of nodes at depth $k$ only), $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are related by $B_{\text{Desc}}$. As a matter of fact, several tracks may be described by the same $BE_k$-descriptor $\mathcal{D}_{BE_k}$. However, whenever a track is described by (the tree obtained from the pruning of) $\mathcal{D}'_{BE_k}$, it is a prefix of at least one of the tracks described by $\mathcal{D}_{BE_k}$. Item 3 is analogous to item 2.

The generalization of Definition 2.22 to pairs of descriptors belonging to $k$-Desc and $k'$-Desc, with $k \neq k'$, is straightforward.

We are now ready to formally define the notion of quotient induced abstract interval model of depth $k$.

**Definition 2.23.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\varphi$ be an HS formula with BE-nesting depth $k \in \mathbb{N}$, and*

$$\Omega = \bigcup_{h \leq k} h\text{-Desc}.$$

*The* quotient induced abstract interval model of depth $k$ *is the* finite *abstract interval model $\mathcal{A}/{\sim_k} = (\mathcal{AP}, \Omega, A_{\text{Desc}}, B_{\text{Desc}}, E_{\text{Desc}}, \sigma)$, where the valuation function $\sigma : \Omega \mapsto 2^{\mathcal{AP}}$ is such that for all $\mathcal{D}_{BE} \in \Omega$, with $\lambda(\text{root}(\mathcal{D}_{BE})) = (v_{in}, S, v_{fin})$,*

$$\sigma(\mathcal{D}_{BE}) = \mu(v_{in}) \cap \bigcap_{v \in S} \mu(v) \cap \mu(v_{fin}).$$

This notion is fundamental for the decidability of the model checking problem for HS, which is proved in the next chapter.

# 3 Decidability of model checking for HS and EXPSPACE-hardness

## Contents



In this chapter we prove the decidability of the model checking problem for HS over finite Kripke structures (under the homogeneity assumption). The proof makes an essential use of quotient induced abstract interval models. Formally, we show that, for any given finite Kripke structure $\mathcal{K}$, the (finite) quotient induced abstract interval model $\mathcal{A}/{\sim_k}$ and the (possibly infinite) abstract interval model $\mathcal{A}_\mathcal{K}$, induced by $\mathcal{K}$, are equivalent with respect to the satisfiability of HS formulas with nesting depth at most $k$. In addition, we show that the notions of $k$-equivalence and $k$-descriptor equivalence are not equivalent (if two tracks are $k$-descriptor equivalent, they are also $k$-equivalent, but not vice versa), and we show how to weaken the notion of $k$-descriptor equivalence to perfectly match $k$-equivalence.

Finally, we prove that the model checking problem for HS against finite Kripke structures is EXPSPACE-hard, if a suitable succinct encoding of formulas is exploited, otherwise it is PSPACE-hard.

## 3.1 The decidability proof

As a preliminary step, we prove a right extension lemma. Let $\mathcal{K}$ be a finite Kripke structure, $k \in \mathbb{N}$, and $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_\mathcal{K}$ with the same $BE_k$-descriptor (and thus, in particular, $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$). The lemma states that if we extend $\rho$ and $\rho'$

"to the right" with the same track $\overline{\rho}$ in $\mathrm{Trk}_{\mathcal{K}}$, with $\big(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})\big) \in \delta$, then the resulting tracks $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$ (both belonging to $\mathrm{Trk}_{\mathcal{K}}$) have the same $BE_k$-descriptor as well. An analogous lemma holds for the extension of the two tracks $\rho$ and $\rho'$ "to the left", which guarantees that $\overline{\rho} \cdot \rho$ and $\overline{\rho} \cdot \rho'$ have the same $BE_k$-descriptor (left extension lemma). In the proof, we will exploit the fact that if two tracks in $\mathrm{Trk}_{\mathcal{K}}$ have the same $BE_{k+1}$-descriptor, then they also have the same $BE_k$-descriptor. The latter can indeed be obtained from the former by removing the nodes of the $BE_{k+1}$-descriptor at depth $k+1$ (leaves) and then deleting isomorphic subtrees possibly originated by the removal (as a matter of fact, we have already specified how to extract a $BE_k$-descriptor from a $BE_{k+1}$-descriptor in Definition 2.22).

**Lemma 3.1.** *(Right extension lemma) Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure and let $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ with the same $BE_k$-descriptor. For any track $\overline{\rho}$ in $\mathrm{Trk}_{\mathcal{K}}$, with $\big(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})\big) \in \delta$, the two tracks $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$ belong to $\mathrm{Trk}_{\mathcal{K}}$ and have the same $BE_k$-descriptor.*

*Proof.* The proof is by induction on $k \in \mathbb{N}$.

- Base case ($k = 0$). Since $\rho$ and $\rho'$ have the same $BE_0$-descriptor, it holds that $\mathrm{fst}(\rho) = \mathrm{fst}(\rho')$, $\mathrm{intstates}(\rho) = \mathrm{intstates}(\rho')$, and $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$ and thus
  - $\mathrm{fst}(\rho \cdot \overline{\rho}) = \mathrm{fst}(\rho) = \mathrm{fst}(\rho') = \mathrm{fst}(\rho' \cdot \overline{\rho})$;
  - $\mathrm{lst}(\rho \cdot \overline{\rho}) = \mathrm{lst}(\rho' \cdot \overline{\rho}) = \mathrm{lst}(\overline{\rho})$;
  - $\mathrm{intstates}(\rho \cdot \overline{\rho}) = \mathrm{intstates}(\rho) \cup \{\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})\} \cup \mathrm{intstates}(\overline{\rho}) = \mathrm{intstates}(\rho') \cup \{\mathrm{lst}(\rho'), \mathrm{fst}(\overline{\rho})\} \cup \mathrm{intstates}(\overline{\rho}) = \mathrm{intstates}(\rho' \cdot \overline{\rho})$

  This allows us to conclude that $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$ have the same $BE_0$-descriptor.
- Inductive step ($k > 0$). Let $\overline{\mathcal{D}_{BE_k}} = (\overline{V}, \overline{E_B} \cup \overline{E_E}, \overline{\lambda})$ and $\overline{\mathcal{D}_{BE_k}}' = (\overline{V}', \overline{E_B}' \cup \overline{E_E}', \overline{\lambda}')$ be respectively the $BE_k$-descriptors of $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$. We prove that $\overline{\mathcal{D}_{BE_k}}$ and $\overline{\mathcal{D}_{BE_k}}'$ are equal (up to isomorphism).

  As for the roots, it is immediate to show that $\overline{\lambda}(\mathrm{root}(\overline{\mathcal{D}_{BE_k}})) = \overline{\lambda}'(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}'))$ (they have the same labeling).

  Let us consider now a node $v \in \overline{V}$ such that $(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}), v) \in \overline{E_B} \cup \overline{E_E}$. We show that there exists a $v' \in \overline{V}'$ such that $(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}'), v') \in \overline{E_B}' \cup \overline{E_E}'$ and the subtrees rooted in $v$ and in $v'$ are isomorphic. We distinguish two cases.
  - Let $\big(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}), v\big) \in \overline{E_B}$. By definition of $BE_k$-descriptor, there exists a prefix $\rho''$ of $\rho \cdot \overline{\rho}$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor of $\rho''$. Three cases are possible.
    - Case 1: $\rho''$ is a (proper) prefix of $\rho$. Since $\rho$ and $\rho'$ have the same $BE_k$-descriptor, there exists a prefix $\rho'''$ of $\rho'$ having the same $BE_{k-1}$-descriptor as $\rho''$.
    - Case 2: $\rho'' = \rho$. Since $\rho$ and $\rho'$ have the same $BE_k$-descriptor, they have also the same $BE_{k-1}$-descriptor.
    - Case 3: $\rho'' = \rho \cdot \tilde{\rho}$, where $\tilde{\rho}$ is a prefix of $\overline{\rho}$. By the inductive hypothesis, $\rho \cdot \tilde{\rho}$ and $\rho' \cdot \tilde{\rho}$ have the same $BE_{k-1}$-descriptor.
  - Let $(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}), v) \in \overline{E_E}$. By definition of $BE_k$-descriptor, there exists a suffix $\rho''$ of $\rho \cdot \overline{\rho}$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor of $\rho''$. We distinguish two cases.

* Case 1: let $\rho''$ be a proper suffix of $\overline{\rho}$ or $\rho'' = \overline{\rho}$. Then, $\rho''$ is a suffix of both $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$. Hence, the same $BE_{k-1}$-descriptor is rooted both in $v$ and in $v'$, for some $v' \in \overline{V}'$ such that $(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}'), v') \in \overline{E_E}'$.
* Case 2: let $\rho'' = \tilde{\rho} \cdot \overline{\rho}$, where $\tilde{\rho}$ is a suffix of $\rho$. If $|\tilde{\rho}| = 1$, $\rho''$ is a suffix of both $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}$, as $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$. Let $|\tilde{\rho}| \geq 2$. Since by hypothesis $\rho$ and $\rho'$ have the same $BE_k$-descriptor, there is a subtree of depth $k-1$ in this descriptor which is associated both with $\tilde{\rho}$ and with a suffix of $\rho'$, say, $\tilde{\rho}'$. By inductive hypothesis, $\rho'' = \tilde{\rho} \cdot \overline{\rho}$ and $\tilde{\rho}' \cdot \overline{\rho}$ have the same $BE_{k-1}$-descriptor.

To sum up, we have shown that (i) $\overline{\lambda}(\mathrm{root}(\overline{\mathcal{D}_{BE_k}})) = \overline{\lambda'}(\mathrm{root}(\overline{\mathcal{D}_{BE_k}}'))$, (ii) for each prefix of $\rho \cdot \overline{\rho}$ there exists a prefix of $\rho' \cdot \overline{\rho}$ with the same $BE_{k-1}$-descriptor, and (iii) for each suffix of $\rho \cdot \overline{\rho}$ there exists a suffix of $\rho' \cdot \overline{\rho}$ with the same $BE_{k-1}$-descriptor. The converse of conditions (ii) and (iii) holds by symmetry. This allows us to conclude that $\overline{\mathcal{D}_{BE_k}}$ and $\overline{\mathcal{D}_{BE_k}}'$ are isomorphic.

□

The next theorem proves that $k$-descriptor equivalent tracks are $k$-equivalent.

**Theorem 3.2.** *($k$-descriptor equivalence implies $k$-equivalence) Let $\psi$ be an HS formula, with $\mathrm{Nest_{BE}}(\psi) = k$, $\mathcal{K}$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$, and $\mathcal{A}_{\mathcal{K}}$ be the abstract interval model induced by $\mathcal{K}$. If $\rho$ and $\rho'$ have the same $BE_k$-descriptor, then*

$$\mathcal{A}_{\mathcal{K}}, \rho \models \psi \iff \mathcal{A}_{\mathcal{K}}, \rho' \models \psi$$

*Proof.* The proof is by induction on the structural complexity of $\psi$.

- $\psi = p$: $\mathcal{A}_{\mathcal{K}}, \rho \models p$ iff $p \in \bigcap_{w \in \mathrm{states}(\rho)} \mu(w)$. Since $\rho$ and $\rho'$ have the same $BE_k$-descriptor, they consist of occurrences of the same set of states of $\mathcal{K}$, that is, $\mathrm{states}(\rho) = \mathrm{states}(\rho')$, witnessed by the root of the $BE_k$-descriptor. Therefore, $\mathcal{A}_{\mathcal{K}}, \rho \models p$ iff $\mathcal{A}_{\mathcal{K}}, \rho' \models p$.

- $\psi = \neg\varphi$: $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$ iff $\mathcal{A}_{\mathcal{K}}, \rho \not\models \varphi$ iff (by inductive hypothesis) $\mathcal{A}_{\mathcal{K}}, \rho' \not\models \varphi$ iff $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$.

- $\psi = \varphi_1 \wedge \varphi_2$: let us assume that $\mathrm{Nest_{BE}}(\varphi_1) = \mathrm{Nest_{BE}}(\psi) = k$ and $\mathrm{Nest_{BE}}(\varphi_2) \leq k$. By the inductive hypothesis, $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi_1$ iff $\mathcal{A}_{\mathcal{K}}, \rho' \models \varphi_1$. Since any pair of tracks that have the same $BE_k$-descriptor have also the same $BE_{k'}$-descriptor, for all $k' \leq k$, by the inductive hypothesis, $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi_2$ iff $\mathcal{A}_{\mathcal{K}}, \rho' \models \varphi_2$. Hence, if $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi_1$ and $\mathcal{A}_{\mathcal{K}}, \rho \models \varphi_2$, and thus $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$. As for the converse, if $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$, then $\mathcal{A}_{\mathcal{K}}, \rho' \models \varphi_1$ and $\mathcal{A}_{\mathcal{K}}, \rho' \models \varphi_2$, and thus $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

- $\psi = \langle \mathrm{A} \rangle \varphi$: $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$ iff there exists $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\rho) = \mathrm{fst}(\overline{\rho})$ and $\mathcal{A}_{\mathcal{K}}, \overline{\rho} \models \varphi$. Analogously, $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$ iff there exists $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\rho') = \mathrm{fst}(\overline{\rho}')$ and $\mathcal{A}_{\mathcal{K}}, \overline{\rho}' \models \varphi$. Since $\rho$ and $\rho'$ have the same $BE_k$-descriptor, it holds that $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$. Hence, we can choose $\overline{\rho} = \overline{\rho}'$, so that $\mathcal{A}_{\mathcal{K}}, \overline{\rho} \models \varphi$ if and only if $\mathcal{A}_{\mathcal{K}}, \overline{\rho}' \models \varphi$.

- $\psi = \langle \mathrm{B} \rangle \varphi$: $\mathrm{Nest_{BE}}(\psi) = 1 + \mathrm{Nest_{BE}}(\varphi) = k$. If $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$, then there is $\overline{\rho} \in \mathrm{Pref}(\rho)$ such that $\mathcal{A}_{\mathcal{K}}, \overline{\rho} \models \varphi$. Let $\mathcal{D}_{BE_k} = (V, E_B \cup E_E, \lambda)$ be the $BE_k$-descriptor for $\rho$. By definition of $BE_k$-descriptor, there exists an edge $(\mathrm{root}(\mathcal{D}_{BE_k}), v) \in E_B$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\overline{\rho}$. Since, by hypothesis, $\rho$

and $\rho'$ have the same $BE_k$-descriptor, there exists a prefix $\overline{\rho}'$ of $\rho'$ such that the subtree rooted in $v$ is the $BE_{k-1}$-descriptor for $\overline{\rho}'$. Now, by the inductive hypothesis, $\mathcal{A}_{\mathcal{K}}, \overline{\rho}' \models \varphi$, and thus $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$. Exactly the same argument allows us to conclude that if $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

- $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$: if $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$, then there exists $\overline{\rho}$ in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \in \mathrm{Pref}(\overline{\rho})$ and $\mathcal{A}_{\mathcal{K}}, \overline{\rho} \models \varphi$. We can express $\overline{\rho}$ as $\rho \cdot \tilde{\rho}$ for some $\tilde{\rho}$ in $\mathrm{Trk}_{\mathcal{K}}$ such that $(\mathrm{lst}(\rho), \mathrm{fst}(\tilde{\rho})) \in \delta$. Now, since $\rho$ and $\rho'$ have the same $BE_k$-descriptor, it holds that $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$. By Lemma 3.1, $\overline{\rho} = \rho \cdot \tilde{\rho}$ and $\rho' \cdot \tilde{\rho}$ have the same $BE_k$-descriptor. By the inductive hypothesis, $\mathcal{A}_{\mathcal{K}}, \rho' \cdot \tilde{\rho} \models \varphi$, and thus $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$. Exactly the same argument allows us to conclude that if $\mathcal{A}_{\mathcal{K}}, \rho' \models \psi$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

The remaining cases can be proven by symmetry. □

Since $k$-descriptor equivalence preserves satisfiability of HS formulas, testing whether $\mathcal{K}, \rho \models \psi$ can be reduced to checking whether $\mathcal{A}/{\sim_k}, [\rho]_{\sim_k} \models \psi$.

**Corollary 3.3.** *Let $\psi$ be an HS formula, with* $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$*, $\mathcal{K}$ be a finite Kripke structure, and $\rho$ be a track in* $\mathrm{Trk}_{\mathcal{K}}$*. It holds that*

$$\mathcal{K}, \rho \models \psi \iff \mathcal{A}/{\sim_k}, [\rho]_{\sim_k} \models \psi.$$

*Proof.* By Definition 2.6, $\mathcal{K}, \rho \models \psi$ if and only if $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$. The proof of the left-to-right implication (if $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$, then $\mathcal{A}/{\sim_k}, [\rho]_{\sim_k} \models \psi$) is by induction on the structural complexity of $\psi$, and it basically makes use of Definition 2.22 and Definition 2.23. The proof of the opposite implication is straightforward. □

By exploiting Corollary 3.3, we can reduce the model checking problem for HS against finite Kripke structures to the model checking problem for multi-modal, finite Kripke structures, whose nodes are all possible (witnessed) descriptors, with depth up to $k$, and there is a distinct accessibility relation for each one of the HS modalities $A$, $B$, $E$, $\overline{A}$, $\overline{B}$, and $\overline{E}$. Since the model checking problem for multi-modal, finite Kripke structures and formulas is decidable (in [Gab87, Lan06], it has been shown that the model checking problem for multi-modal Kripke structures and formulas is decidable in polynomial time with respect to both the size of the Kripke structure and the length of the formula), decidability of the model checking problem for HS against finite Kripke structures immediately follows.

**Theorem 3.4.** *The model checking problem for HS against finite Kripke structures is decidable (with a* non-elementary *algorithm).*

*Proof.* Lemma 2.14 provides a non-elementary upper bound to the number of $BE_h$-descriptors, with $0 \leq h \leq k$, as well as to the size of $BE_{k+1}$-descriptors, with respect to the size of the Kripke structure and the nesting depth $k$ of the input HS formula. Hence, the derived model checking problem for multi-modal, finite Kripke structures has to be solved over a model whose size has a non-elementary upper bound. □

## 3.2 The relation between $k$-equivalence and corresponding $BE_k$-descriptors

In the previous section (in Theorem 3.2), we prove that $k$-descriptor equivalence is a sufficient condition for $k$-equivalence, that is, if two tracks are $k$-descriptor

equivalent, then they are $k$-equivalent. However, it is not a necessary one. To show that the converse does not hold, consider once more the finite Kripke structure $\mathcal{K}_{Equiv}$ in Figure 2.1. The tracks $v_0^5$ and $v_0^6$ of $\mathcal{K}_{Equiv}$ have the same $BE_2$-descriptor, but not the same $BE_3$-descriptor, yet there exists no formula $\psi$, with $\text{Nest}_{\text{BE}}(\psi) \leq 3$, such that $\mathcal{K}, v_0^6 \models \psi$ and $\mathcal{K}, v_0^5 \not\models \psi$. Intuitively, since these two tracks are made of a different number of occurrences of the same state, the only way to distinguish them is by means of the formula $\langle \text{B} \rangle^4 \top$, or similar ones, for which $\mathcal{K}, v_0^6 \models \langle \text{B} \rangle^4 \top$ and $\mathcal{K}, v_0^5 \not\models \langle \text{B} \rangle^4 \top$, but these formulas have a BE-nesting depth *greater* than 3.

In the following, we introduce the notion of *corresponding $BE_k$-descriptors*, and we prove that it provides a necessary and sufficient condition for $k$-equivalence. Such a notion allows us to rephrase equivalence between tracks in terms of more abstract characteristics of their descriptors, in a stronger way than Theorem 3.2. As an example, by exploiting the correspondence among descriptors it defines and the statement of Theorem 3.11 below, it will be possible to prove that $v_0^5$ and $v_0^6$ are actually 3-equivalent.

**Definition 3.5.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, let $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ be two $BE_k$-descriptors associated with some of its tracks, and let $(v_{in}, S, v_{fin})$ and $(v'_{in}, S', v'_{fin})$ be the labels of the root of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively. We say that $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are* corresponding $BE_k$*-descriptors if and only if:*

- *the two roots are labelled by the same set of propositions, that is,*

$$\bigcap_{w \in \{v_{in}\} \cup S \cup \{v_{fin}\}} \mu(w) \;=\; \bigcap_{w' \in \{v'_{in}\} \cup S' \cup \{v'_{fin}\}} \mu(w');$$

- *for any track $\rho \in \text{Trk}_{\mathcal{K}}$, with $\text{fst}(\rho) = v_{fin}$, there is a track $\rho' \in \text{Trk}_{\mathcal{K}}$ with $\text{fst}(\rho') = v'_{fin}$, such that $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors, and vice versa (we say that the $BE_k$-descriptors for $\rho$ and $\rho'$ are $A$-successors of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively);*

- *for any track $\rho \in \text{Trk}_{\mathcal{K}}$, with $\text{lst}(\rho) = v_{in}$, there is a track $\rho' \in \text{Trk}_{\mathcal{K}}$, with $\text{lst}(\rho') = v'_{in}$, such that $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors, and vice versa (we say that the $BE_k$-descriptors for $\rho$ and $\rho'$ are $\overline{A}$-successors of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively);*

- *given a track $\tilde{\rho}$ associated with $\mathcal{D}_{BE_k}$ and a track $\tilde{\rho}'$ associated with $\mathcal{D}'_{BE_k}$, for any track $\rho$, with $(v_{fin}, \text{fst}(\rho)) \in \delta$, there is a track $\rho'$, with $(v'_{fin}, \text{fst}(\rho')) \in \delta$, such that both $\tilde{\rho} \cdot \rho$ and $\tilde{\rho}' \cdot \rho'$ belong to $\text{Trk}_{\mathcal{K}}$, and they are associated with corresponding $BE_k$-descriptors, and vice versa (we say that the $BE_k$-descriptors for $\tilde{\rho} \cdot \rho$ and $\tilde{\rho}' \cdot \rho'$ are $\overline{B}$-successors of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively)*[1]*;*

- *given a track $\tilde{\rho}$ associated with $\mathcal{D}_{BE_k}$ and a track $\tilde{\rho}'$ associated with $\mathcal{D}'_{BE_k}$, for any track $\rho$, with $(\text{lst}(\rho), v_{in}) \in \delta$, there is a track $\rho'$, with $(\text{lst}(\rho'), v'_{in}) \in \delta$, such that both $\rho \cdot \tilde{\rho}$ and $\rho' \cdot \tilde{\rho}'$ belong to $\text{Trk}_{\mathcal{K}}$, and they are associated with corresponding $BE_k$-descriptors, and vice versa (we say that the $BE_k$-descriptors for $\rho \cdot \tilde{\rho}$ and $\rho' \cdot \tilde{\rho}'$ are $\overline{E}$-successors of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively);*

[1] If a track $\overline{\rho}$ was considered in place of $\tilde{\rho}$ and they are both associated with $\mathcal{D}_{BE_k}$, by the right extension lemma $\tilde{\rho} \cdot \rho \in \text{Trk}_{\mathcal{K}}$ and $\overline{\rho} \cdot \rho \in \text{Trk}_{\mathcal{K}}$ are associated with the same descriptor as well.

- *whenever $k > 0$, for any subtree of depth $k-1$ in $\mathcal{D}_{BE_k}$, whose root is linked to the root of $\mathcal{D}_{BE_k}$ via a $B$-edge, there is a corresponding subtree of depth $k-1$ in $\mathcal{D}'_{BE_k}$, whose root is linked to the root of $\mathcal{D}'_{BE_k}$ via a $B$-edge, and vice versa;*
- *whenever $k > 0$, for any subtree of depth $k-1$ in $\mathcal{D}_{BE_k}$, whose root is linked to the root of $\mathcal{D}_{BE_k}$ via an $E$-edge, there exists a corresponding subtree of depth $k-1$ in $\mathcal{D}'_{BE_k}$, whose root is linked to the root of $\mathcal{D}'_{BE_k}$ via an $E$-edge, and vice versa.*

It can easily be checked that the correspondence between descriptors of Definition 3.5 is an equivalence relation. Definition 3.5 expresses a form of bisimulation among $BE_k$-descriptors with respect to the defined relations of $A$-successor, $\overline{A}$-successor, $B$-successor, $\overline{B}$-successor, $E$-successor, and $\overline{E}$-successor. Since, in a finite Kripke structure, every state has (at least) a successor with respect to $\delta$, $BE_k$-descriptors always have both $A$-successors and $\overline{B}$-successors. On the contrary, $BE_k$-descriptors may have no $\overline{A}$-successors or $\overline{E}$-successors, because a state does not necessarily have a predecessor with respect to $\delta$, Finally, a $BE_k$-descriptor has $B$- and $E$-successors if and only if $k \geq 1$ and the represented tracks are long enough. The number of proper prefixes (resp., suffixes) of short tracks can indeed be less than $k$. In such a case, the actual height of $BE_k$-descriptors is less than the nominal height $k$, and thus it may happen that the $BE_i$-descriptor and $BE_j$-descriptor, with $i \neq j$, for a track are isomorphic. When collecting all the $BE_i$-descriptor, for $0 \leq i \leq k$ (for instance, in Definition 3.6 below), isomorphic descriptors of different depths will be considered as distinct elements.

The set of descriptors and their successor relations, corresponding to the various HS modalities, allow us to define a graph structure.

**Definition 3.6.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure. The graph $\mathcal{G}$ of the $BE$-descriptors of depth at most $k$, with $k \geq 0$, witnessed by some tracks of $\mathcal{K}$, is a pair $(V_\mathcal{G}, E_\mathcal{G})$, where $E_\mathcal{G} \subseteq V_\mathcal{G} \times V_\mathcal{G}$ is a set of labeled edges, such that:*

- *$V_\mathcal{G}$ contains a node for each $BE_t$-descriptor, with $0 \leq t \leq k$, witnessed by some track of $\mathcal{K}$;*
- *edges in $E_\mathcal{G}$ are labeled with $X \in \{A, B, E, \overline{A}, \overline{B}, \overline{E}\}$ according to the following criteria:*
  - *$(v, v') \in E_\mathcal{G}$ is an $X$-edge, with $X \in \{A, \overline{A}, \overline{B}, \overline{E}\}$, whenever the descriptor of $v'$ is an $X$-successor of the descriptor of $v$;*
  - *$(v, v') \in E_\mathcal{G}$ is a $B$-edge whenever the descriptors associated with $v$ and $v'$ are $\mathcal{D}_{BE_t}$ and $\mathcal{D}'_{BE_{t-1}}$, respectively (for some $t \geq 1$), and $\mathcal{D}'_{BE_{t-1}}$ is isomorphic to a subtree of $\mathcal{D}_{BE_t}$ connected to the root of $\mathcal{D}_{BE_t}$ via a $B$-edge;*
  - *$(v, v') \in E_\mathcal{G}$ is an $E$-edge whenever the descriptors associated with $v$ and $v'$ are $\mathcal{D}_{BE_t}$ and $\mathcal{D}'_{BE_{t-1}}$, respectively (for some $t \geq 1$), and $\mathcal{D}'_{BE_{t-1}}$ is isomorphic to a subtree of $\mathcal{D}_{BE_t}$ connected to the root of $\mathcal{D}_{BE_t}$ via an $E$-edge.*

The set of nodes $V_\mathcal{G}$ is finite and the out-degree of every node is finite as well. Moreover, $V_\mathcal{G}$ can be partitioned into $k$ sets, according to the depth of the descriptors associated with its nodes. A node associated with a descriptor of depth $t$ can be connected to a node associated with a descriptor of depth $t-1$, with $0 < t \leq k$, only by $B$- or $E$-edges. It is possible to show that two descriptors are corresponding if and only if the associated nodes in the graph are bisimilar. Thus, the definition of

correspondence between descriptors could be equivalently expressed in terms of a standard notion of bisimilarity among nodes of $\mathcal{G}$.

For technical reasons, we need to introduce a variant of two of the previously-defined concepts: the nesting depth of formulas, to take into consideration the nesting of all HS modalities (not only $B$ and $E$ as in Definition 2.10), and descriptor correspondence *up to* a bounded number of "steps".

**Definition 3.7.** *The nesting depth of an HS formula $\psi$, denoted by* $\mathrm{Nest}(\psi)$*, is inductively defined on the structure of $\psi$ as follows:*

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

It trivially holds that $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq \mathrm{Nest}(\psi)$ for all formulas $\psi$.

The next definition differs from Definition 3.5 in the limitation to a bounded number of coinductive steps that it introduces in the descriptor correspondence.

**Definition 3.8.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, let $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ be two $BE_k$-descriptors associated with some of its tracks, and let $(v_{in}, S, v_{fin})$ and $(v'_{in}, S', v'_{fin})$ be the labels of the root of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively. We say that $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are* corresponding *$BE_k$-descriptors up to depth $n$ if and only if:*

- *the two roots are labeled by the same set of propositions, that is,*

$$\bigcap_{w \in \{v_{in}\} \cup S \cup \{v_{fin}\}} \mu(w) = \bigcap_{w' \in \{v'_{in}\} \cup S' \cup \{v'_{fin}\}} \mu(w');$$

- *if $n > 0$:*
  - *for any track $\rho \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\rho) = v_{fin}$, there is a track $\rho' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\rho') = v'_{fin}$, such that $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors up to depth $n-1$, and vice versa;*
  - *for any track $\rho \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{lst}(\rho) = v_{in}$, there is a track $\rho' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{lst}(\rho') = v'_{in}$, such that $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors up to depth $n-1$, and vice versa;*
  - *given two tracks $\tilde{\rho}$ and $\tilde{\rho}'$ associated with $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively, for any track $\rho$, with $(v_{fin}, \mathrm{fst}(\rho)) \in \delta$, there is a track $\rho'$, with $(v'_{fin}, \mathrm{fst}(\rho')) \in \delta$, such that both $\tilde{\rho} \cdot \rho$ and $\tilde{\rho}' \cdot \rho'$ belong to $\mathrm{Trk}_{\mathcal{K}}$, and they are associated with corresponding $BE_k$-descriptors up to depth $n-1$, and vice versa;*
  - *given two tracks $\tilde{\rho}$ and $\tilde{\rho}'$ associated with $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$, respectively, for any track $\rho$, with $(\mathrm{lst}(\rho), v_{in}) \in \delta$, there is a track $\rho'$, with $(\mathrm{lst}(\rho'), v'_{in}) \in \delta$, such that both $\rho \cdot \tilde{\rho}$ and $\rho' \cdot \tilde{\rho}'$ belong to $\mathrm{Trk}_{\mathcal{K}}$, and they are associated with corresponding $BE_k$-descriptors up to depth $n-1$, and vice versa;*
  - *whenever $k > 0$, for any subtree of depth $k-1$ in $\mathcal{D}_{BE_k}$, whose root is linked to the root of $\mathcal{D}_{BE_k}$ via a $B$-edge (resp. $E$-edge), there is a subtree of depth $k-1$ in $\mathcal{D}'_{BE_k}$, whose root is linked to the root of $\mathcal{D}'_{BE_k}$ via a $B$-edge (resp., $E$-edge), corresponding up to depth $n-1$, and vice versa.*

In order to prove that corresponding descriptors actually capture the notion of $k$-equivalence (Theorem 3.11 below), we need a couple of preparatory lemmas, whose proofs are given in the appendix (Section A.1).

**Lemma 3.9.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure and $\rho, \rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$. For all $n, k \in \mathbb{N}$, with $k \leq n$, if $\rho$ and $\rho'$ are $k$-equivalent with respect to all HS formulas $\psi$, with $\mathrm{Nest}(\psi) \leq n$, then the $BE_k$-descriptors of $\rho$ and $\rho'$ are corresponding up to depth $n$.*

**Lemma 3.10.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure. For all $n, k \in \mathbb{N}$, with $k \geq 1$, if two descriptors $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are corresponding up to depth $n$, then $\mathcal{D}_{BE_k}|_{k-1}$ and $\mathcal{D}'_{BE_k}|_{k-1}$ are corresponding up to depth $n$, where $\mathcal{D}_{BE_k}|_{k-1}$ denotes the descriptor obtained from $\mathcal{D}_{BE_k}$ by deleting its nodes at depth $k$ (and removing possible isomorphic subtrees).*

The following theorem formally states the relationship between corresponding descriptors and track equivalence.

**Theorem 3.11.** *Let $\mathcal{K}$ be a finite Kripke structure, $k \in \mathbb{N}$, and $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$. The tracks $\rho$ and $\rho'$ are $k$-equivalent if and only if $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors.*

*Proof.* ($\Rightarrow$) Let us first show that if $\rho$ and $\rho'$ are $k$-equivalent, then they are associated with corresponding $BE_k$-descriptors. The proof directly follows from Lemma 3.9. Since $\rho$ and $\rho'$ are $k$-equivalent, with no bound on the nesting depth of formulas, then their $BE_k$-descriptors are corresponding, with no bound on the depth.

($\Leftarrow$) We now prove that, for any HS formula $\psi$, with $\mathrm{Nest}_{\mathrm{BE}}(\psi) = k$, if $\rho$ and $\rho'$ are associated with corresponding $BE_k$-descriptors, then $\mathcal{K}, \rho \models \psi \iff \mathcal{K}, \rho' \models \psi$. The proof is by induction on the structure of the formula.

- Let $\mathcal{K}, \rho \models p$, for some $p \in \mathcal{AP}$. Since the roots for the $BE$-descriptors of $\rho$ and $\rho'$ are labeled with the same set of proposition letters, it immediately follows that $\mathcal{K}, \rho' \models p$.

- Let $\mathcal{K}, \rho \models \psi_1 \wedge \psi_2$. Then, $\mathcal{K}, \rho \models \psi_1$ and $\mathcal{K}, \rho \models \psi_2$. Let $\mathrm{Nest}_{\mathrm{BE}}(\psi_1) = k$ and assume w.l.o.g. that $\mathrm{Nest}_{\mathrm{BE}}(\psi_2) = t \leq k$. By Definition 3.8 and Lemma 3.10, it immediately follows that if $\rho$ and $\rho'$ have corresponding $BE_k$-descriptors, then they also have corresponding $BE_t$-descriptors, with $t \leq k$. Hence, by the inductive hypothesis, $\mathcal{K}, \rho' \models \psi_1$ and $\mathcal{K}, \rho' \models \psi_2$, and, as a consequence, $\mathcal{K}, \rho' \models \psi_1 \wedge \psi_2$.

- Let $\mathcal{K}, \rho \models \neg\psi$. Then, $\mathcal{K}, \rho \not\models \psi$. By the inductive hypothesis, $\mathcal{K}, \rho' \not\models \psi$, and thus $\mathcal{K}, \rho' \models \neg\psi$.

- Let $\mathcal{K}, \rho \models \langle \mathrm{A} \rangle \psi$. Then, there exists a track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$, such that $\mathcal{K}, \overline{\rho} \models \psi$. Since the $BE_k$-descriptors for $\rho$ and $\rho'$ are corresponding, there exists, in particular, a track $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, such that the $BE_k$-descriptors for $\overline{\rho}$ and $\overline{\rho}'$ are corresponding. By the inductive hypothesis, $\mathcal{K}, \overline{\rho}' \models \psi$, so $\mathcal{K}, \rho' \models \langle \mathrm{A} \rangle \psi$. The $\langle \overline{\mathrm{A}} \rangle$ case is symmetric (notice that, due to the correspondence of the $BE_k$-descriptors for $\rho$ and $\rho'$, there exists $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{lst}(\overline{\rho}) = \mathrm{fst}(\rho)$, if and only if there exists $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{lst}(\overline{\rho}') = \mathrm{fst}(\rho')$).

- Let $\mathcal{K}, \rho \models \langle \overline{\mathrm{B}} \rangle \psi$. Then, there is a track $\overline{\rho}$, with $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$ and $\rho \cdot \overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, such that $\mathcal{K}, \rho \cdot \overline{\rho} \models \psi$. Since the $BE_k$-descriptors for $\rho$ and $\rho'$ are corresponding, there exists, in particular, a track $\overline{\rho}'$, with $(\mathrm{lst}(\rho'), \mathrm{fst}(\overline{\rho}')) \in \delta$, such that $\rho \cdot \overline{\rho}$ and $\rho' \cdot \overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$ have corresponding $BE_k$-descriptors. By the inductive hypothesis $\mathcal{K}, \rho' \cdot \overline{\rho}' \models \psi$, and thus $\mathcal{K}, \rho' \models \langle \overline{\mathrm{B}} \rangle \psi$. The $\langle \overline{\mathrm{E}} \rangle$ case is symmetric (a remark similar to the one for the $\langle \overline{\mathrm{A}} \rangle$ case can be done).

- Let $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle \psi$. Then, there exists a track $\overline{\rho} \in \mathrm{Pref}(\rho)$ such that $\mathcal{K}, \overline{\rho} \models \psi$. Since the $BE_k$-descriptors for $\rho$ and $\rho'$ are corresponding, the subtree of depth $k-1$ for $\overline{\rho}$, in the $BE_k$-descriptor for $\rho$, corresponds to a subtree of depth $k-1$, in the $BE_k$-descriptor for $\rho'$. By definition of descriptor, there exists a track $\overline{\rho}' \in \mathrm{Pref}(\rho')$ associated with the latter subtree. By the inductive hypothesis, $\mathcal{K}, \overline{\rho}' \models \psi$, and thus $\mathcal{K}, \rho' \models \langle \mathrm{B} \rangle \psi$. The $\langle \mathrm{E} \rangle$ case is symmetric, and thus its analysis is omitted.

This concludes the proof. □

We started this section by illustrating the case of the two tracks $v_0^5$ and $v_0^6$ of $\mathcal{K}_{Equiv}$. They have the same $BE_2$-descriptor (it is shown in Figure 3.1(a)), but not the same $BE_3$-descriptor (the $BE_3$-descriptor for $v_0^5$ is shown in Figure 3.1(b)). The $BE_3$-descriptor for $v_0^6$, indeed, features one more subtree, that is, the $BE_2$-descriptor for $v_0^5$, which is not present in Figure 3.1(b). However, such a subtree corresponds to the $BE_2$-descriptor for $v_0^4$. Symmetrically, the same happens for the suffix $v_0^5$ of $v_0^6$. Thus, $v_0^5$ and $v_0^6$, which have corresponding $BE_3$-descriptors, are 3-equivalent by Theorem 3.11.

On the other hand, the $BE_4$-descriptors for $v_0^5$ and $v_0^6$ are not corresponding. In Figure 3.2, a part of the graph $\mathcal{G}$ of the $BE$-descriptors for the tracks of $\mathcal{K}_{Equiv}$ is shown. As it is evident from the figure, there exists a path consisting of 4 $B$-edges starting from the node of the $BE_4$-descriptor for $v_0^6$, whereas there is no a path of the same length starting from the node of the $BE_4$-descriptor for $v_0^5$. Hence, $v_0^5$ and $v_0^6$ are not 4-equivalent (as we already pointed out, $\mathcal{K}, v_0^6 \models \langle \mathrm{B} \rangle^4 \top$ but $\mathcal{K}, v_0^5 \not\models \langle \mathrm{B} \rangle^4 \top$).

## 3.3 EXPSPACE-hardness of HS model checking

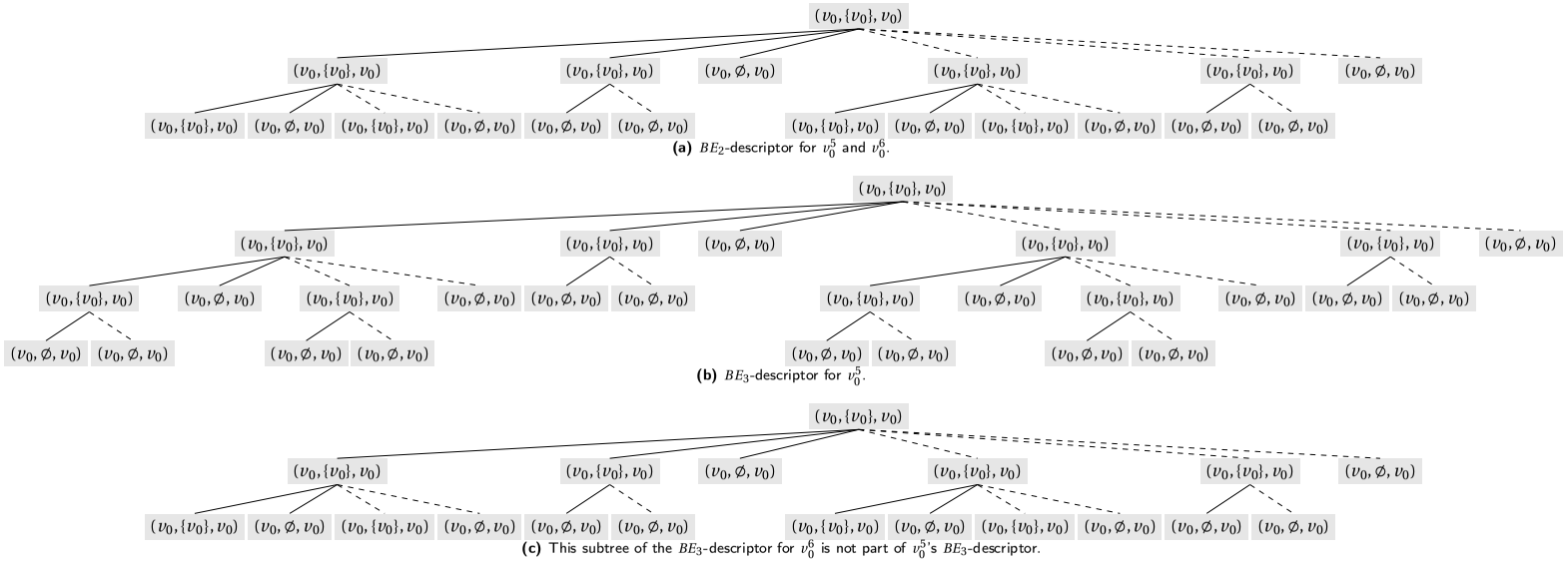
We conclude this chapter by proving that the model checking problem for HS over finite Kripke structures is EXPSPACE-hard. As a preparatory work, we introduce a succinct encoding of HS formulas, according to which we write $\langle \mathrm{B} \rangle^k \psi$ for

$$\underbrace{\langle \mathrm{B} \rangle \langle \mathrm{B} \rangle \cdots \langle \mathrm{B} \rangle}_{k \text{ times}} \psi,$$

and we represent $k$ in binary (the same for all the other HS modalities). As we will prove, if we exploit this encoding, the model checking problem for HS is EXPSPACE-hard, otherwise—using the standard unary notation—it is PSPACE-hard.

**Theorem 3.12.** *The model checking problem for HS against finite Kripke structures is EXPSPACE-hard (under a LOGSPACE reduction), if formulas are* succinctly encoded, *otherwise it is PSPACE-hard.*

*Proof.* Let us consider a language $L$ decided by a *deterministic one-tape* Turing machine $M$ (w.l.o.g.) that, on an input of size $n$, requires no more than $2^{n^k} - 3$ symbols




**(a)** $BE_2$-descriptor for $v_0^5$ and $v_0^6$.

**(b)** $BE_3$-descriptor for $v_0^5$.

**(c)** This subtree of the $BE_3$-descriptor for $v_0^6$ is not part of $v_0^5$'s $BE_3$-descriptor.

**Figure 3.1:** Tracks $v_0^5$ and $v_0^6$ have the same $BE_2$-descriptor, which is shown in Figure 3.1(a). Figure 3.1(b) depicts the $BE_3$-descriptor for $v_0^5$ (the $BE_3$-descriptor for $v_0^6$ is not shown due to lack of space). With respect to the one in Figure 3.1(b), it features two additional subtrees, one associated with a prefix, the other with a suffix. The former subtree is shown in 3.1(c); not surprisingly, it is isomorphic (equal) to 3.1(a).

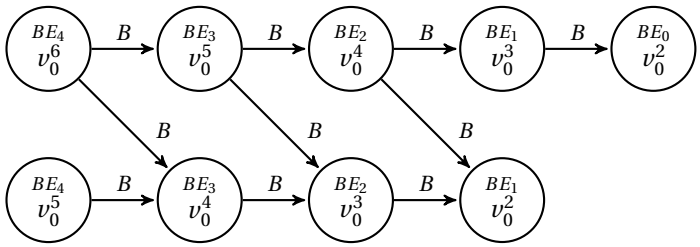



**Figure 3.2:** Part of the graph $\mathscr{G}$ of the $BE_t$-descriptors ($t \leq 4$) for the tracks of $\mathcal{K}_{Equiv}$. In all nodes, we report the depth of the descriptors they are associated with (top) and a witness track for the descriptor (bottom).

on its tape (we are assuming a high enough constant $k \in \mathbb{N}$). Hence, $L$ belongs to EXPSPACE. Let $\Sigma$ and $Q$ be respectively the alphabet and the set of states of $M$, and let # be a special symbol, which does not belong to $\Sigma$, used as separator for configurations (in the following, we let $\Sigma' = \Sigma \cup \{\#\}$). The alphabet $\Sigma$ 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 $M$.

We exploit a standard encoding for computations, called *computation table* (or tableau) (see [Pap94, Sip12] for further details). Each configuration of $M$ is a sequence over the alphabet $\Gamma = \Sigma' \cup (Q \times \Sigma)$. A symbol $(q, c) \in Q \times \Sigma$ occurring at 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, there is exactly one occurrence of a symbol in $Q \times \Sigma$ in each configuration).

Since $M$ uses no more than $2^{n^k} - 3$ symbols on its tape, the size of a configuration is $2^{n^k}$ (we need 3 occurrences of the special 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 to reach length $2^{n^k}$ (which is a fixed number, once the input length is known).

The computation table is a matrix of $2^{n^k}$ columns, where the $i$-th row records the configuration of $M$ at the $i$-th computation step.

An example of a table is given in Figure 3.3. In the first configuration (row), the head is in the leftmost position (to the right of the delimiters #) and $M$ is in state $q_0$. In addition, the string symbols $c_0 c_1 \cdots c_{n-1}$ are 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 $\delta_M \subseteq Q \times \Sigma \times Q \times \Sigma \times \{\rightarrow, \leftarrow, \bullet\}$ of $M$, with the standard meaning for the components (the first one gives the current state, the second the symbol on tape currently read, the third the next state, the fourth the symbol replaced in the current position, the fifth the move of the head to right, left, or stay). Being $M$ deterministic , $\delta_M$ is actually a function of $Q \times \Sigma$.

Following [Pap94, Sip12], we introduce the notion of (legal) window. A window is a $2 \times 3$ 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 $\delta_M$ in the obvious

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | # | $(q_0, c_0)$ | $c_1$ | $c_2$ | $\cdots$ | $\cdots$ | $c_{n-1}$ | ⊔ | ⊔ | $\cdots$ | $\cdots$ | ⊔ | # |
| # | # | $c'_0$ | $(q_1, c_1)$ | $c_2$ | $\cdots$ | $\cdots$ | $c_{n-1}$ | ⊔ | ⊔ | $\cdots$ | $\cdots$ | ⊔ | # |
| ⋮ | ⋮ | | | | $\ddots$ | $\ddots$ | | | | | | | ⋮ |
| ⋮ | ⋮ | | | | $\ddots$ | $\ddots$ | | | | | | | ⋮ |
| # | # | $\cdots$ | $\cdots$ | $(q_{yes}, c_k)$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | # |

$2^{n^k}$

**Figure 3.3:** An example of a computation table.

sense. Actually, the set of legal windows, which we denote by $Wnd \subseteq (\Gamma^3)^2$, is a suitable tabular representation of the transition relation $\delta_M$. For instance, 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 pair $((x, y, z), (x', y', z')) \in Wnd$ can be represented as follows:

| | | |
|---|---|---|
| $x$ | $y$ | $z$ |
| $x'$ | $y'$ | $z'$ |

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

where the following constraints must be satisfied:

1. if all $x, y, z \in \Sigma'$ ($x$, $y$, $z$ are not state-symbol pairs), then $y = y'$;
2. if one among $x$, $y$, and $z$ belongs to $Q \times \Sigma$, then $x'$, $y'$ and $z'$ are univocally determined by $\delta_M$;
3. $(x = \# \Rightarrow x' = \#) \wedge (y = \# \Rightarrow y' = \#) \wedge (z = \# \Rightarrow z' = \#)$.

As we already said, $M$ never overwrites an occurrence of #; we can assume that the head never visits a cell labelled with # as well (see [Pap94]). As a matter of fact, in some window, condition 2 would require to move the head right (or left) overwriting # (or just visiting it), while 3 does not allow one to replace an occurrence of # with another symbol (notice that $(q_i, \#)$ does not belong to $\Gamma$ for any state $q_i$ of $M$). In such a case, the window is not valid and thus it is discarded (it does not belong to $Wnd$).

In the following, we define a finite Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and a formula $\psi$ of HS 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 proposition letters is $\mathcal{AP} = \Gamma \cup \Gamma^3 \cup \{start\}$. The finite Kripke structure $\mathcal{K}$ is obtained by suitably composing a basic pattern, called *gadget* (see Figure 3.4). Any instance of the gadget is associated with a triple of symbols $(a, b, c) \in \Gamma^3$, that is, a sequence of three adjacent symbols in a configuration, and it consists of 3 states $q^0_{(a,b,c)}$, $q^1_{(a,b,c)}$, and $q^2_{(a,b,c)}$ such that

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

Moreover,

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

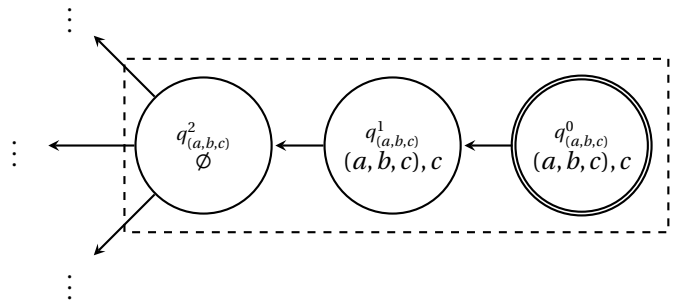



**Figure 3.4:** An instance of the gadget for $(a,b,c) \in \Gamma^3$.

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

The finite Kripke structure $\mathcal{K}$ has (an instance of) a gadget for every $(x,y,z) \in \Gamma^3$, and for all $(x,y,z),(x',y',z') \in \Gamma^3$, it holds that $q^0_{(x',y',z')} \in \delta(q^2_{(x,y,z)})$ if and only if $x' = y$ and $y' = z$. Moreover, $\mathcal{K}$ has some additional (auxiliary) states $w_0,\cdots,w_6$, whose relationships are described in Figure 3.5, and $\delta(w_6) = \{q^0_{(\#,\#,x)} \mid x \in \Gamma\}$. It is worth noticing that the overall size of $\mathcal{K}$ only depends on $|\Gamma|$ and it is constant with respect to the input string $c_0c_1\cdots c_{n-1}$ of $M$.

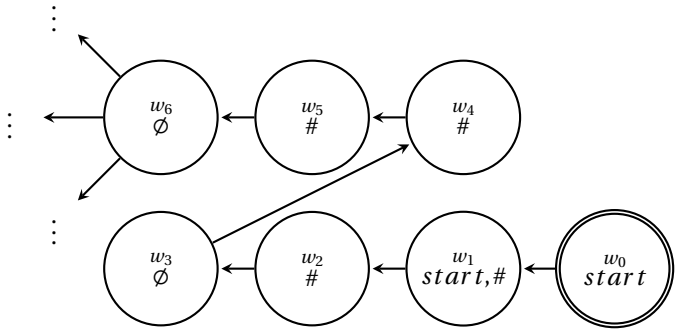



**Figure 3.5:** Initial part of $\mathcal{K}$.

Now, we want to decide whether or not an input string belongs to the language $L$ by solving the model checking problem $\mathcal{K} \models start \rightarrow \langle A \rangle \xi$, where $\xi$ is satisfied only by those tracks which represent a successful computation of $M$. Since the only (initial) track which satisfies $start$ is $w_0w_1$ (see Figure 3.5), we are actually verifying the existence of a track which begins with $w_1$ and satisfies $\xi$.

As for $\xi$, it basically requires that a track $\rho$, with $\mathrm{fst}(\rho) = w_1$, for which $\mathcal{K},\rho \models \xi$, mimics a successful computation of $M$. First, 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 $\rho$ is $p$ (notice that as a consequence of the gadget structure, only subtracks $\overline{\rho} = \rho(i,i+1)$, with $i \mod 3 = 0$, of $\rho$ can satisfy some proposition letters). A symbol of a configuration is mapped to an occurrence of an instance of a gadget in $\rho$; in turn, $\rho$ 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 us now formally define the HS formula $\xi$:

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

The first conjunct

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

requires a track to contain an occurrence of the accepting state of $M$, $q_{yes}$.

The second conjunct $\psi_{input}$ is a bit more involved. It requires 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 ended by a # (we recall that $\ell(k)$, which has been introduced in Section 2.2, is satisfied only by those tracks whose length equals $k$, with $k \geq 2$, and it has a binary encoding of $O(\log k)$ bits):

$$\begin{aligned}\psi_{input} = [B](\ell(7) \rightarrow \langle \mathrm{A} \rangle (q_0, c_0)) \wedge [B](\ell(10) \rightarrow \langle \mathrm{A} \rangle c_1) \wedge [B](\ell(13) \rightarrow \langle \mathrm{A} \rangle c_2) \wedge \\ \vdots \\ [B](\ell(7+3(n-1)) \rightarrow \langle \mathrm{A} \rangle c_{n-1}) \wedge \\ [B](\langle \mathrm{B} \rangle^{5+3n} \top \wedge [B]^{3 \cdot 2^{n^k} - 6} \bot \rightarrow \langle \mathrm{A} \rangle ((\ell(2) \wedge \bigwedge_{a \in \Gamma} \neg a) \vee \sqcup)) \wedge [B](\ell(3 \cdot 2^{n^k} - 2) \rightarrow \langle \mathrm{A} \rangle \#).\end{aligned}$$

Finally, the conjunct $\psi_{window}$ enforces the window constraint: if $(d, e, f) \in \Gamma^3$ is witnessed by a subinterval (of length 2) in the subtrack of $\rho$ corresponding to the $j$-th configuration of $M$, then, at the same position of (the subtrack of $\rho$ associated with) configuration $j-1$, there must be some $(a, b, c) \in \Gamma^3$ which is such that $((a, b, c), (d, e, f)) \in Wnd$.

$$\begin{aligned}\psi_{window} = [B](\langle \mathrm{B} \rangle^{3(2^{n^k}+2)+1} \top \rightarrow \\ \bigwedge_{(d,e,f) \in \Gamma^3} (\langle \mathrm{A} \rangle (d, e, f) \rightarrow [E](\ell(3 \cdot 2^{n^k}) \rightarrow \bigvee_{((a,b,c),(d,e,f)) \in Wnd} \langle \overline{\mathrm{A}} \rangle (a, b, c)))).\end{aligned}$$

The subformula $\langle \mathrm{B} \rangle^{3(2^{n^k}+2)+1} \top$ guarantees that we are not considering the (subtrack associated with the) first configuration. Moreover, if some prefix $\tilde{\rho}$ of $\rho$ satisfies ($\langle \mathrm{B} \rangle^{3(2^{n^k}+2)+1} \top$ and) $\langle \mathrm{A} \rangle (d, e, f)$, for some $(d, e, f) \in \Gamma^3$, then it holds that

$$\mathcal{K}, \tilde{\rho} \models [E](\ell(3 \cdot 2^{n^k}) \rightarrow \bigvee_{((a,b,c),(d,e,f)) \in Wnd} \langle \overline{\mathrm{A}} \rangle (a, b, c)).$$

This amounts to say that the suffix $\hat{\rho}$ of $\tilde{\rho}$ of length $3 \cdot 2^{n^k}$ is such that

$$\mathcal{K}, \hat{\rho} \models \bigvee_{((a,b,c),(d,e,f)) \in Wnd} \langle \overline{\mathrm{A}} \rangle (a, b, c),$$

that is, $\hat{\rho}$ is the subtrack between (the prefixes of $\rho$ corresponding to) the same position (same column) in two adjacent configurations (rows of the table), and it is forced to begin with an occurrence of $q^1_{(a,b,c)}$ and to end with $q^0_{(d,e,f)}$, for some $((a, b, c), (d, e, f)) \in Wnd$.

It is immediate to check that all the integers which need to be stored in the formula are less than or equal to $3 \cdot 2^{n^k} + 7$, and thus $O(n^k)$ bits suffice. This allows us to conclude that the formula can be generated in polynomial time (and logarithmic working space).

If we *do not* allow the binary encoding of the exponents, the model checking problem for HS formulas is PSPACE-hard (under a LOGSPACE reduction): the proof is the same as before, but in order for the formula $\xi$ to be generated in polynomial time, we must restrict ourselves to computations of Turing machines using at most polynomial space. □

In conclusion, in this chapter we have proved that the model checking problem for HS is decidable, and it is EXPSPACE-hard if a proper succinct encoding of formulas is exploited, otherwise it is PSPACE-hard. Actually, we have also proved that the fragment $HS[A, \overline{A}, B, E]$ is, again, EXPSPACE-hard if the succinct encoding of formulas is allowed, and PSPACE-hard otherwise, because the formula $\xi$ of the above proof does not contain occurrences of $\overline{B}$ and $\overline{E}$ modalities.

In the next chapter we shall analyze a pair of fragments whose model checking is featured by a lower complexity if compared to that of (full) HS, namely $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ and $HS[A, \overline{A}, E, \overline{B}, \overline{E}]$.

# 4 A model checking algorithm based on track representatives

## Contents



In this chapter, we restrict our attention to the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ and we show that the model checking problem for it has a lower complexity (compared to that of full HS). By symmetry, all the results presented in the next sections are applicable to $HS[A, \overline{A}, E, \overline{B}, \overline{E}]$, as well.

Since $E$ modality does not occur in formulas of $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$, we shall exploit $B_k$-descriptors instead of $BE_k$-descriptors in the following algorithms; moreover we preliminary have to "adapt" some of the notions we have already defined in the previous chapter to such a fragment:

- the B-nesting depth of an $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula $\psi$, $\mathrm{Nest_B}(\psi)$, is defined like $\mathrm{Nest_{BE}}$, but it does not account for the $E$ modality.
- Two tracks $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$ are said $k$-equivalent if and only if, for every formula $\psi$ of $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ with $\mathrm{Nest_B}(\psi) = k$, it holds that $\mathcal{K}, \rho \models \psi$ iff $\mathcal{K}, \rho' \models \psi$.
- Two tracks $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$ are $k$-descriptor equivalent, denoted by $\rho \sim_k \rho'$, if and only if the $B_k$-descriptors for $\rho$ and $\rho'$ coincide.

Next we state a lemma and two propositions, which are preparatory to Theorem 4.4.

**Lemma 4.1.** *Let $k \in \mathbb{N}$, $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure and $\rho_1, \rho_1', \rho_2, \rho_2'$ be tracks in $\mathrm{Trk}_{\mathcal{K}}$ 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 proof can be found in A.2.1. The next propositions immediately follow:

**Proposition 4.2.** *(Right extension) Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$. For any $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$, it holds that $\rho \cdot \overline{\rho} \sim_k \rho' \cdot \overline{\rho}$.*

**Proposition 4.3.** *(Left extension) Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$. For any $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $(\mathrm{lst}(\overline{\rho}), \mathrm{fst}(\rho)) \in \delta$, it holds that $\overline{\rho} \cdot \rho \sim_k \overline{\rho} \cdot \rho'$.*

The former proposition states that if we extend the two tracks $\rho$ and $\rho'$ 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 $\rho$ and $\rho'$ "to the left". In these Propositions 4.2 and 4.3, $|\overline{\rho}| \geq 2$; however both continue to hold if $|\overline{\rho}| = 1$.

The following theorem is the analogous to Theorem 3.2 for $B_k$-descriptors; its proof is omitted, as it is basically a simplification of that of Theorem 3.2.

**Theorem 4.4.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ and $\rho'$ two tracks in $\mathrm{Trk}_{\mathcal{K}}$, $\mathcal{A}_{\mathcal{K}}$ the abstract interval model induced by $\mathcal{K}$ and $\psi$ a formula of $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ with $\mathrm{Nest_B}(\psi) = k$. If $\rho \sim_k \rho'$, then $\mathcal{A}_{\mathcal{K}}, \rho \models \psi \iff \mathcal{A}_{\mathcal{K}}, \rho' \models \psi$.*

Thanks to this theorem and to the notions of *descriptor element indistinguishability* and *track representatives* presented in the next sections, we will provide an EXPSPACE model checking algorithm for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$. Finally, we will prove that the model checking problem for the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ is NEXP-hard if a suitable succinct encoding of formulas is exploited.

## 4.1 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 cardinality 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

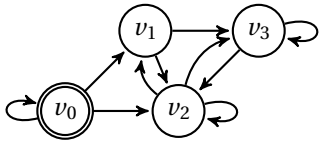



**Figure 4.1:** An example of finite Kripke structure.

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 formula of $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ with $B$-nesting depth up to $k$.

**Definition 4.5.** *Let $\rho = v_0 v_1 \cdots v_n$ be a track of a finite Kripke structure. The* descriptor sequence *$\rho_{ds}$ for $\rho$ is $d_0 \cdots d_{n-1}$, where $d_i = \rho_{ds}(i) = (v_0, \mathrm{intstates}(v_0 \cdots v_{i+1}), v_{i+1})$, for $0 \leq i \leq n-1$. We denote the set of descriptor elements occurring in $\rho_{ds}$ by $DElm(\rho_{ds})$.*

As an example, let us consider the finite Kripke structure of Figure 4.1 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 $\rho$ is:

$$\begin{aligned}\rho_{ds} = {} & (v_0, \emptyset, v_0)(v_0, \{v_0\}, v_0)(v_0, \{v_0\}, v_1) \\ & (v_0, \{v_0, v_1\}, v_2)(v_0, \{v_0, v_1, v_2\}, v_1)(v_0, \{v_0, v_1, v_2\}, v_2) \\ & (v_0, \{v_0, v_1, v_2\}, v_3)(v_0, \Delta, v_3)(v_0, \Delta, v_2)(v_0, \Delta, v_2)\end{aligned}$$

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

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

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 $\rho'$ and $\rho''$, respectively, and $\rho'$ is a prefix of $\rho''$.

**Definition 4.6.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ and $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v_{in}, S'', v''_{fin})$ be two descriptor elements in $\rho_{ds}$. 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 4.7.** *Let $\rho_{ds}$ 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 $0 \le i < j < n$.*

*Proof.* $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are associated with the tracks $\rho_1 = v_0 \cdots v_{i+1}$ 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 4.8.** *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 easily be 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 4.9.** *If it holds that both $d'\,\mathrm{R_t}\,d''$ and $d''\,\mathrm{R_t}\,d'$, where $d' = (v_{in}, S', v'_{fin})$ 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.* $S' \cup \{v'_{fin}\} \subseteq S'' \subseteq S'' \cup \{v''_{fin}\} \subseteq S'$ 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 $\rho_{ds}$ for a track $\rho$: $\rho_{ds}$ 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 4.10.** *A* cluster *$\mathcal{C}$ of (Type-2) descriptor elements is a maximal set of descriptor elements $\{d_1, \cdots, d_s\} \subseteq DElm(\rho_{ds})$ such that $d_i\,\mathrm{R_t}\,d_j$ and $d_j\,\mathrm{R_t}\,d_i$ for all $i, j \in \{1, \cdots, s\}$.*

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 4.11.** *Let $\rho_{ds}$ be a descriptor sequence and $\mathcal{C}$ be one of its clusters. The* subsequence of $\rho_{ds}$ associated with *$\mathcal{C}$ is a subsequence $\rho_{ds}(i, j)$ such that $\rho_{ds}(i') \in \mathcal{C}$ if and only if $i \le i' \le j < |\rho_{ds}|$.*

For example, the descriptor sequence $\rho_{ds}$ for $\rho = v_0 v_0 v_0 v_1 v_2 v_1 v_2 v_3 v_3 v_2 v_2$, a track of the finite Kripke structure in Figure 4.1, is shown again here; subsequences associated with clusters are surrounded by boxes ($\Delta = \{v_0, v_1, v_2, v_3\}$):

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

It is worth observing that:

- the descriptor elements of a cluster $\mathcal{C}$ are contiguous (in other words, they form a subsequence), i.e., 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 $R_t$ 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 descriptor elements may have multiple occurrences in a descriptor sequence, implies that we need to somehow limit the number of occurrences of Type-2 descriptor 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 4.12.** *Let $\rho_{ds}$ 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 4.12, 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 $\rho_{ds}$.

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.

Propositions 4.13, 4.14 state some basic properties of the $k$-indistinguishability relation.

**Proposition 4.13.** *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 $\rho_{ds}$. Then, $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are $(k-1)$-indistinguishable.*

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

*Base case* ($k = 2$). 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. □

**Proposition 4.14.** *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 $\rho_{ds}$. If $\rho_{ds}(j) = \rho_{ds}(m)$, for some $i < j < m$, then $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are $k$-indistinguishable.*

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

*Base case* ($k = 1$). 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

$$DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, m-1)) = DElm(\rho_{ds}(0, j-1)),$$

proving the property.

*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. □

In Figure 4.2, we provide some examples of $k$-indistinguishability relations, for $k \in \{1, 2, 3\}$, for a track of the finite Kripke structure depicted in Figure 4.1.

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

**Theorem 4.15.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$. Two occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$, $0 \leq i < j < |\rho_{ds}|$, of the same descriptor element are $k$-indistinguishable if and only if $\rho(0, i+1) \sim_k \rho(0, j+1)$.*

*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 labeled 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 labeled by different descriptor elements).

*Base case* ($k = 1$). If $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* 1-indistinguishable, then $DElm(\rho_{ds}(0, i-1)) \subset DElm(\rho_{ds}(0, j-1))$. So there exists $d \in DElm(\rho_{ds}(0, j-1))$

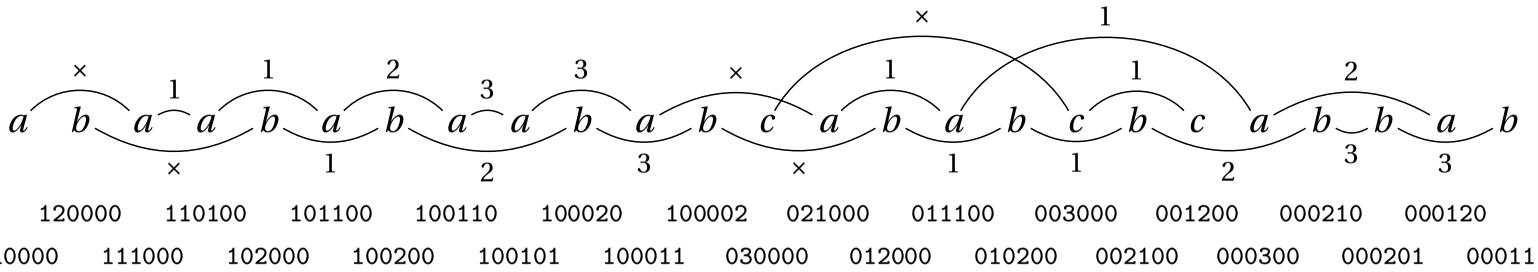



**Figure 4.2:** The track $\rho = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2v_2v_3v_2$ of the finite Kripke structure depicted in Figure 4.1 generates the descriptor sequence $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, \{v_1, v_2\}, v_3)abaababaababcababcbcabbab$, 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 4.14 and Proposition 4.16. 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.

such that $d \notin DElm(\rho_{ds}(0, i-1))$, and the $B_1$-descriptor for $\rho(0, j+1)$ has a leaf labeled 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 $\rho(0, i+1)$. □

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 any pair of tracks of a Kripke structure.

We conclude the section with the following proposition, which easily follows from Theorem 4.15.

**Proposition 4.16.** *Let $\rho_{ds}(i)$, $\rho_{ds}(j)$, and $\rho_{ds}(m)$, with $0 \leq i < j < m < |\rho_{ds}|$, be three occurrences of the same descriptor element. If both the pair $\rho_{ds}(i)$ and $\rho_{ds}(j)$ and the pair $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are $k$-indistinguishable, for some $k \geq 1$, then $\rho_{ds}(i)$ and $\rho_{ds}(m)$ are $k$-indistinguishable as well.*

## 4.2 Track representatives

In this section, we will exploit the $k$-indistinguishability relation between descriptor elements in a descriptor sequence $\rho_{ds}$ for a track $\rho$ to possibly replace $\rho$ by a $k$-descriptor equivalent, *shorter* track $\rho'$ 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 $\rho_{ds}$ be the descriptor sequence for a track $\rho$. 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 $\rho$ by the $k$-descriptor equivalent, shorter track $\rho(0, i+1) \cdot \overline{\rho}$. Indeed, by Theorem 4.15, $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same $B_k$-descriptor and thus, by Proposition 4.2, $\rho = \rho(0, j+1) \cdot \overline{\rho}$ 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 $\rho'$, which is $k$-descriptor equivalent to $\rho$, 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 $\rho$, $|DElm(\rho_{ds})| \leq |W|^2 + 1$, 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 $\rho$ 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 4.17.** *For each track $\rho$ of $\mathcal{K}$, with descriptor element $d$, there exists a track $\rho'$ of $\mathcal{K}$, with the same descriptor element, such that $|\rho'| \leq 2 + |W|^2$.*

*Proof.* By induction on the length $\ell (\geq 2)$ of $\rho$.

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

*Inductive step* ($\ell > 2$). We distinguish two cases. If $\rho_{ds}$ 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 $\rho$ satisfies the condition $\ell \leq 2 + |W|^2$. If $\rho_{ds}(i) = \rho_{ds}(j)$, for $0 \leq i < j < |\rho_{ds}|$, $\rho(0, i+1)$ 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 4.2, $\rho = \rho(0, j+1) \cdot \rho(j+2, |\rho| - 1)$ and $\rho'$ have the same descriptor element. By the inductive hypothesis, there is a track $\rho''$ of $\mathcal{K}$ associated with the same descriptor element of $\rho'$ (and $\rho$) with $|\rho''| \leq 2 + |W|^2$. □

Proposition 4.17 will be used in the unravelling Algorithm 1 as a termination criterion (referred to as *0-termination criterion*) for unravelling a finite Kripke structure when it is not necessary to observe multiple occurrences of the same descriptor element.

**Definition 4.18** (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 considering tracks longer than $2 + |W|^2$, while exploring the unravelling of the Kripke structure from $v$.*

Let us now consider the problem of establishing a bound for tracks devoid of pairs of $k$-indistinguishable occurrences of descriptor elements. We first notice that in a descriptor sequence $\rho_{ds}$ for a track $\rho$, 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 $\rho_{ds}$, one has to bound the length of subsequences of $\rho_{ds}$ 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 $\rho_{ds}$ 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 $i > u$. 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 $\rho_{ds}(u, i)$;
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 $\rho_{ds}(u, i-1)$.

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 $Q_0(u)$, $Q_1(u)$,... 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 4.3, which maps the descriptor sequence $\rho_{ds}$ 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 $\rho_{ds}$ 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 indexes $h < h' < i < i'$ 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 characterization 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 4.19.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ and $i$ be a position in the subsequence of $\rho_{ds}$ 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 $f(\rho_{ds}, i) = (Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \cdots, Q_s(i))$.*

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

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

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

**Figure 4.3:** Definition of the scan function $f$.

An example of a configuration sequence is given in Figure 4.2.

**Theorem 4.20.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ 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 $d \in Q_s(i-1)$, $d \in Q_{s+1}(i)$, for some $s \in \{-2,-1\} \cup \mathbb{N}$, and $c(i-1) >_{lex} c(i)$.*

The proof of Theorem 4.20 is given in A.2.2. It is worth pointing out that, by this theorem, it follows that the definition of $f$ is in fact redundant: cases (c) and (e) never happen.

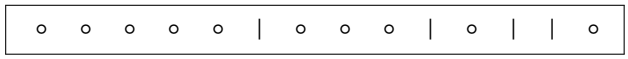
We show now how to select all and only those tracks which do not feature any pair of $k$-indistinguishable occurrences of descriptor elements. To this end, we make use of the scan function $f$ with $k+3$ arrays (the value $k+3$ derives from the $k$ of descriptor element indistinguishability, plus the three arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$). Theorem 4.20 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 4.21.** *For all $n, t \in \mathbb{N}^+$, the number of distinct $t$-tuples of natural numbers whose sum equals $n$ is*

$$\varepsilon(n,t) = \binom{n+t-1}{n} = \binom{n+t-1}{t-1}.$$

*Proof.* The following figure suggests an alternative representation of a tuple, in the form of a configuration of separators/bullets; such a representation is *unambiguous* (i.e., there exists a bijection between configurations of separators/bullets and tuples):




The sum of the integers of the tuple equals the number of bullets and the size of the tuple is the number of separators plus 1. Since there exist $\varepsilon(n,t) = \binom{n+t-1}{t-1}$ distinct ways of choosing $t-1$ separators among $n+t-1$ different places (and places which are not chosen must contain bullets), there are exactly $\varepsilon(n,t)$ distinct $t$-tuples of naturals whose sum equals $n$. □

Two upper bounds for $\varepsilon(n,t)$ can be derived: $\varepsilon(n,t) \leq (n+1)^{t-1}$ 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 4.21 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 $\rho$, $\rho_{ds}$ 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 $\rho$ is exceeded, then there is at least one pair of $k$-indistinguishable occurrences of a descriptor element in $\rho_{ds}$: $|\rho| \leq 1 + (|\mathcal{C}_1|+1)^{k+2} + (|\mathcal{C}_2|+1)^{k+2} + \cdots + (|\mathcal{C}_s|+1)^{k+2} + |W|$, with $s \leq |W|$ and the last addend is to count the occurrences of Type-1 descriptor elements.

Since clusters are disjoint, their union is a subset of $DElm(\rho_{ds})$, and $|DElm(\rho_{ds})| \leq 1+|W|^2$, we have:

$$|\rho| \leq 1+(|C_1|+|C_2|+\cdots+|C_s|+|W|)^{k+2}+|W| \leq$$
$$1+(|DElm(\rho_{ds})|+|W|)^{k+2}+|W| \leq$$
$$1+(1+|W|^2+|W|)^{k+2}+|W| \leq 1+(1+|W|)^{2k+4}+|W|.$$

Analogously:

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

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 4.22.** *Let $\mathcal{K}$ be a finite Kripke structure and $\rho$ be a track in $Trk_{\mathcal{K}}$. If it holds that $|\rho| > \tau(|W|,k)$, 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 $\rho$.*

*Proof.* (Sketch) If $|\rho| > \tau(|W|,k)$, then there exists at least one subsequence of $\rho_{ds}$, associated with some cluster $\mathcal{C}$, which contains at least a 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 4.15, 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 $\rho$ as the concatenation $\tilde{\rho}_2 \cdot \overline{\rho}$ for some $\overline{\rho}$. By Proposition 4.2, 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 $\mathcal{K}$, and it is shorter than $\rho$. If $|\rho'| \leq \tau(|W|,k)$, we have proved the thesis; otherwise, we can iterate the process by applying the above contraction to $\rho'$. □

Theorem 4.22 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 4.23** ($(k \geq 1)$-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 taking into consideration tracks longer than $\tau(|W|,k)$ while exploring the unravelling of the structure from $v$.*

Now we outline the *unravelling Algorithm* 1: it scans 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 $\rho$ whenever there exist $k$-indistinguishable occurrences of a descriptor element $d$ in $\rho_{ds}$.

The following theorem proves soundness and completeness of the unravelling Algorithm 1 in forward direction. Backward direction is analogous.

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

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 $\rho$ from $v$ to the current node only if:
5: **if** $k = 0$ **then**
6: Apply 0-termination criterion of Definition 4.18
7: **else**
8: **if** The last descriptor element $d$ of (the descriptor sequence of) the current track $\rho$ is $k$-indistinguishable from a previous occurrence of $d$ **then**
9: do *not* return $\rho$ 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 $\rho$ *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 4.18
15: **else**
16: **if** There exist two $k$-indistinguishable occurrences of a descriptor element $d$ in (the descriptor sequence of) the current track $\rho$ **then**
17: do *not* return $\rho$
18: Do not visit tracks of length greater than $\tau(|W|, k)$

**Theorem 4.24.** *Let $\mathcal{K}$ be a finite Kripke structure, $v$ be a state in $W$, and $k \in \mathbb{N}$. For every track $\rho$ of $\mathcal{K}$, with* $\text{fst}(\rho) = v$ *and* $|\rho| \geq 2$*, the unravelling algorithm returns a track $\rho'$ of $\mathcal{K}$, with* $\text{fst}(\rho') = v$*, such that $\rho$ and $\rho'$ have the same $B_k$-descriptor and* $|\rho'| \leq \tau(|W|, k)$.

The proof of Theorem 4.24 is given in A.2.3. It basically shows how a "contracted variant" of a track $\rho$ is (indirectly) computed by Algorithm 1.

As an example, in place of the track $\rho$ of Figure 4.2, the algorithm returns the following contracted track: $\rho' = v_0 v_1 v_2 v_3 v_3 v_2 v_3 v_3 v_2 v_3 v_2 v_3 v_2 v_1 v_3 v_2 v_3 v_2 v_1 v_2 v_1 v_3 v_2$. It can be checked that $\rho'$ does not contain any pair of 3-indistinguishable occurrences of a descriptor element and that $\rho$ and $\rho'$ 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, 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 occurrences of 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 $\rho_{ds}$ for $\rho$ 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 $\overline{\rho} \in \text{Trk}_{\mathcal{K}}$. Since $\rho(0, j+1)$ and $\rho(0, i+1)$ have the same $B_k$-descriptor, by Proposition 4.3 for any $\overline{\rho} \in \text{Trk}_{\mathcal{K}}$ 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 $\mathcal{K}$.

## 4.3 The model checking algorithm

Building on the unravelling Algorithm 1, we can easily define the model checking procedure `ModCheck`$(\mathcal{K}, \psi)$ (Algorithm 2). In particular $u$.`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 $u$.

`ModCheck`$(\mathcal{K}, \psi)$ exploits the procedure `Check`$(\mathcal{K}, k, \psi, \tilde{\rho})$ (Algorithm 3), which checks a formula $\psi$ of B-nesting depth $k$ against a track $\tilde{\rho}$ of the Kripke structure $\mathcal{K}$.

**Algorithm 2** `ModCheck`$(\mathcal{K}, \psi)$

```
1: k ← Nest_B(ψ)
2: u ← New(Unrav(K, init_state(K), k, FORWARD))
3: while u.hasMoreTracks() do
4:     ρ̃ ← u.getNextTrack()
5:     if Check(K, k, ψ, ρ̃) = 0 then
6:         return 0: "K, ρ̃ ⊭ ψ"                ◁ Counterexample
7: return 1: "K ⊨ ψ"
```

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 A.2.4).

**Lemma 4.25.** *Let $\psi$ 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 `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 Algorithm 3, 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 $\rho$, it is possible to check $\hat{\rho}_1\cdot\rho$ (since, by the right extension Proposition 4.2, they have the same $B_{k-1}$-descriptor).

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

**Theorem 4.26.** *Let $\psi$ be an HS$[A, \overline{A}, B, \overline{B}, \overline{E}]$ formula and $\mathcal{K}$ be a finite Kripke structure. `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 4.25, it follows that `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 `ModCheck`$(\mathcal{K}, \psi)$ returns 1.

On the other hand, if `ModCheck`$(\mathcal{K}, \psi) = 1$, then for any track $\rho$ with $\mathrm{fst}(\rho) = w_0$ *returned by the unravelling algorithm*, `Check`$(\mathcal{K}, \mathrm{Nest}_B(\psi), \psi, \rho)$ returns 1 and, by

**Algorithm 3** `Check`$(\mathcal{K}, k, \psi, \tilde{\rho})$

```
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:     for each v ∈ W such that (v, fst(ρ̃)) is an edge of K do
48:         if Check(K, k, φ, v · ρ̃) = 1 then
49:             return 1
50:         u ← New(Unrav(K, v, k, BACKWARD))
51:         while u.hasMoreTracks() do
52:             ρ ← u.getNextTrack()
53:             if Check(K, k, φ, ρ · ρ̃) = 1 then
54:                 return 1
55:     return 0
```

Lemma 4.25, we have $\mathcal{K},\rho \models \psi$. Assume now that a track $\tilde{\rho}$ with $\mathrm{fst}(\tilde{\rho}) = w_0$ is *not* returned by the unravelling algorithm. By Theorem 4.24, 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 as $\tilde{\rho}$ (with $k = \mathrm{Nest_B}(\psi)$). Since $\mathcal{K},\tilde{\rho} \models \psi \iff \mathcal{K},\overline{\rho} \models \psi$ (by Theorem 4.4) 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 $\psi$, thus $\mathcal{K} \models \psi$. □

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 $\psi$ is the input formula) simultaneously active calls to `Check`, the total space needed by the considered algorithms is $\big(|\psi|+1\big)\cdot O(|W|+\mathrm{Nest_B}(\psi))\cdot\tau(|W|,\mathrm{Nest_B}(\psi))$ 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.

As a particular case, formulas $\psi$ of the fragment $HS[A,\overline{A},B,\overline{E}]$ can be checked in polynomial space, because for these formulas $\mathrm{Nest_B}(\psi) = 0$ (we will come back to this HS fragment in the next chapter).

## 4.4 NEXP-hardness of model checking for the fragment $HS[A,\overline{A},B,\overline{B},\overline{E}]$

We conclude this chapter by proving that the model checking problem for formulas of the fragment $HS[A,\overline{A},B,\overline{B},\overline{E}]$, interpreted over finite Kripke structures, is NEXP-hard when a succinct encoding of formulas is exploited (Theorem 4.27). Like in the proof of Theorem 3.12, we allow to write $\langle \mathrm{B}\rangle^k\psi$ standing for $k$ repetitions of $\langle \mathrm{B}\rangle$ before $\psi$, with $k$ represented in binary. The same can be done for all other HS modalities. Additionally, $\bigwedge_{i=l,\cdots,r}\psi(i)$ 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 $\mathrm{expand}(\psi)$ the expanded form of $\psi$: all exponents $k$ have to be eliminated from $\psi$ by explicitly repeating $k$ times each HS modality with such an exponent, and big conjunctions must be replaced by conjunctions of formulas without indexes.

**Theorem 4.27.** *The model checking problem for* succinct $HS[A,\overline{A},B,\overline{B},\overline{E}]$ *formulas against finite Kripke structures is NEXP-hard.*

*Proof.* Since this proof is very similar to that of Theorem 3.12, we only outline which elements have to be changed.

- We 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 $n$ ($L$ belongs to NEXP).

- As a consequence, $M$ uses at most $2^{n^k}-3$ cells on its tape, so the size of a configuration is $2^{n^k}$. Moreover the number of configurations is $2^{n^k}-3$. Therefore the computation table is now a matrix of $2^{n^k}-3$ rows and $2^{n^k}$ columns.

- Finally $\psi_{window}$ changes in this way (obviously the $\psi_{window}$ described in the proof of 3.12 is not suitable, as it contains modality $E$):

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

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

In the proof of Theorem 3.12, in order to consider the positions in a track corresponding to the same cell of two consecutive rows of the computation table, we can "isolate" a subinterval of such a track in between these two positions, thanks to the joint use of modalities $B$ and $E$. In this proof, the lack of $E$ forces us to consider pairs of positions by taking into account two prefixes of the track (with suitable lengths): if we had considered an EXPSPACE Turing machine, the number of rows of the table could have been doubly exponential in $n$: this would have forced to have indexes in $\psi_{window}$—which accounts for all the cells of the computation table—that are doubly exponential in $n$ (thus encoded with an exponential quantity of bits), and the formula could not have been generated in polynomial time. Therefore we have to restrict ourselves to NEXP. □

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

In the next chapter, we shall analyze some more HS fragments and we will obtain as a by-product that the model checking problem for formulas of $HS[A,\overline{A},B,\overline{B},\overline{E}]$ is PSPACE-hard if the described succinct encoding is *not* allowed.

# 5 Some well-behaved fragments of HS

## Contents



In this chapter, we identify some well-behaved fragments of the HS logic, namely, $\forall HS[A,\overline{A},B,E]$ (and $\exists HS[A,\overline{A},B,E]$), $HS[A,\overline{A},\overline{B},\overline{E}]$, and $HS[A,\overline{A}]$, which are still expressive enough to capture meaningful interval properties of state-transition systems and whose model checking problem exhibits a considerably lower computational complexity.

In Section 5.1, we deal with the fragment $\forall HS[A,\overline{A},B,E]$, including formulas of $HS[A,\overline{A},B,E]$ in which only universal modalities are allowed and negation can be applied to propositional formulas only. We first provide a coNP model checking algorithm for $\forall HS[A,\overline{A},B,E]$, and then we show that the model checking problem for the pure propositional fragment $HS[\mathsf{Prop}]$ is coNP-hard. The two results allow us to conclude that the model checking problem for both $HS[\mathsf{Prop}]$ and $\forall HS[A,\overline{A},B,E]$ is coNP-complete. In addition, upper and lower bounds to the complexity of the problem for $HS[A,\overline{A}]$ (the logic of temporal neighbourhood) directly follow. Recall that the model checking algorithm for $HS[A,\overline{A},B,\overline{B},\overline{E}]$ of Section 4.3 can check formulas of $HS[A,\overline{A},\overline{B},\overline{E}]$ in polynomial space. Since $HS[A,\overline{A}]$ is a fragment of $HS[A,\overline{A},\overline{B},\overline{E}]$ and $HS[\mathsf{Prop}]$ is a fragment of $HS[A,\overline{A}]$, complexity of model checking for $HS[A,\overline{A}]$ is in between coNP and PSPACE.

In Section 5.2, we focus our attention on $HS[A,\overline{A},\overline{B},\overline{E}]$ and we prove that the model checking problem for $HS[A,\overline{B}]$ is PSPACE-hard. PSPACE-completeness of $HS[A,\overline{A},\overline{B},\overline{E}]$ (and $HS[A,\overline{B}]$) immediately follows. From this, we get for free a lower bound to the complexity of the model checking problem for $HS[A,\overline{A},B,\overline{B},\overline{E}]$ in the non-succinct case, which turns out to be PSPACE-hard.

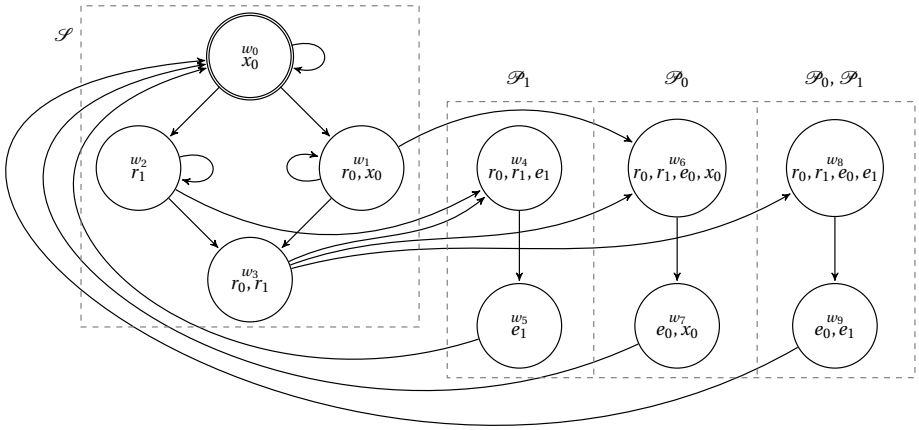



**Figure 5.1:** A simple state-transition system.

The following simple example shows the HS fragments we consider in this chapter at work.

**Example 5.1.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$, with $\mathcal{AP} = \{r_0, r_1, e_0, e_1, x_0\}$, be the Kripke structure of Figure 5.1, that models the interactions between a scheduler $\mathcal{S}$ and two processes, $\mathcal{P}_0$ and $\mathcal{P}_1$, which possibly ask for a shared resource. At the initial state $w_0$, $\mathcal{S}$ has not received any request from the processes yet, while in $w_1$ (resp., $w_2$) only $\mathcal{P}_0$ (resp., $\mathcal{P}_1$) has sent a request, and thus $r_0$ (resp., $r_1$) holds. As long as at most one process has sent a request, $\mathcal{S}$ is not forced to allocate the resource ($w_1$ and $w_2$ have self loops). At $w_3$, both $\mathcal{P}_0$ and $\mathcal{P}_1$ are waiting for the shared resource, and hence both $r_0$ and $r_1$ hold there. State $w_3$ has transitions only towards $w_4$, $w_6$, and $w_8$. At $w_4$ (resp., $w_6$) $\mathcal{P}_1$ (resp., $\mathcal{P}_0$) can access the resource: $e_1$ (resp., $e_0$) holds in $w_4 w_5$ (resp., $w_6 w_7$). However, a faulty transition may be taken from $w_3$: in $w_8$ and $w_9$ both $\mathcal{P}_0$ and $\mathcal{P}_1$ are using the resource (both $e_0$ and $e_1$ hold in $w_8 w_9$). Finally, from $w_5$, $w_7$, and $w_9$ the system can only move to $w_0$, where $\mathcal{S}$ waits for new requests from $\mathcal{P}_0$ and $\mathcal{P}_1$.*

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

*In the following formulas notice that $\mathcal{K} \models [A]\psi$ (equivalently, $\mathcal{K} \models [E]\psi$) iff $\psi$ holds over any (reachable) computation sub-interval.*

*It can be easily checked that $\mathcal{K} \not\models [E]\neg(e_0 \wedge e_1)$ (this formula is in $\forall HS[A, \overline{A}, B, E]$), that is, mutual exclusion is not guaranteed, as the faulty transition leading to $w_8$ may be taken at $w_3$, and then both $\mathcal{P}_0$ and $\mathcal{P}_1$ access the resource in $w_8 w_9$ ($e_0 \wedge e_1$ holds).*

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

*It also holds that $\mathcal{K} \models [A]\big(r_0 \wedge r_1 \rightarrow [A](e_0 \vee e_1 \vee \bigwedge_{p \in \mathcal{P}} \neg p)\big)$ (in $HS[A, \overline{A}]$ and $HS[A, \overline{A}, \overline{B}, \overline{E}]$). Indeed, if both processes send a request to $\mathcal{S}$ (state $w_3$), then it immediately allocates the resource. Formally, if $r_0 \wedge r_1$ holds over some tracks (the only possibilities are $w_3 w_4$, $w_3 w_6$, and $w_3 w_8$), then in any possible subsequent interval of length 2 $e_0 \vee e_1$ holds, that is, $\mathcal{P}_0$ or $\mathcal{P}_1$ are executed, or $\bigwedge_{p \in \mathcal{P}} \neg p$ holds, if we consider*

*tracks longer than 2. On the contrary, if only one process asks for the resource, then $\mathcal{S}$ can arbitrarily delay its allocation, that is, $\mathcal{K} \not\models [A]\big(r_0 \rightarrow [A](e_0 \vee \bigwedge_{p\in\mathcal{P}} \neg p)\big)$.*

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

## 5.1 The fragments $\forall HS[A,\overline{A},B,E]$, $HS[A,\overline{A}]$, and $HS$[Prop]

In this section, we take into consideration the universal and existential fragments of $HS[A,\overline{A},B,E]$, respectively denoted by $\forall HS[A,\overline{A},B,E]$ and $\exists HS[A,\overline{A},B,E]$, whose formulas are defined as follows:

$$\psi ::= \beta \mid \psi \wedge \psi \mid [A]\psi \mid [B]\psi \mid [E]\psi \mid [\overline{A}]\psi$$

$$\text{(resp., } \psi ::= \beta \mid \psi \vee \psi \mid \langle A\rangle\psi \mid \langle B\rangle\psi \mid \langle E\rangle\psi \mid \langle \overline{A}\rangle\psi\text{)},$$

where

$$\beta ::= p \mid \beta \vee \beta \mid \beta \wedge \beta \mid \neg\beta \mid \bot \mid \top \text{ with } p \in \mathcal{AP}.$$

The intersection of $\forall HS[A,\overline{A},B,E]$ and $\exists HS[A,\overline{A},B,E]$ is the set of all and only pure propositional formulas ($HS$[Prop]). Negations occur in pure propositional formulas only, and formulas with modalities can be combined only by conjunctions (in $\forall HS[A,\overline{A},B,E]$) or disjunctions (in $\exists HS[A,\overline{A},B,E]$). The negation of any formula of $\forall HS[A,\overline{A},B,E]$ can be transformed into an equivalent $\exists HS[A,\overline{A},B,E]$ formula (of at most double length), and vice versa, by using De Morgan's laws and the equivalences $[X]\psi \equiv \neg\langle X\rangle\neg\psi$ and $\neg\neg\psi \equiv \psi$.

We now outline a *non-deterministic* algorithm to decide the model checking problem for a $\forall HS[A,\overline{A},B,E]$ formula $\psi$. The algorithm searches for a counterexample to $\psi$. As we already pointed out, $\neg\psi$ is equivalent to a suitable formula $\psi'$ of the dual fragment $\exists HS[A,\overline{A},B,E]$. Hence, the algorithm looks for an initial track of the Kripke structure that satisfies $\psi'$.

For the satisfiability check, we use the non-deterministic procedure `Check` (Algorithm 4), which does not exploit (directly or indirectly) neither $BE_k$- descriptors nor $B_k$-descriptors, but only descriptor elements.

Recall that (Proposition 4.17, Definition 4.18) if a descriptor element $d$ is witnessed, then there exists a track of length at most $2+|W|^2$ associated with it, and thus to generate a (all) witnessed descriptor element(s) with initial state $v$, we just need to non-deterministically visit the unravelling of $\mathcal{K}$ from $v$ up to depth $2+|W|^2$.

The procedure `Check` takes as input a formula $\psi$ of $\exists HS[A,\overline{A},B,E]$ and a witnessed descriptor element $d = (v_{in}, S, v_{fin})$ and it returns **Yes** if and only if there exists a track $\rho \in \mathrm{Trk}_{\mathcal{K}}$ associated with $d$ such that $\mathcal{K},\rho \models \psi$. The procedure is recursively defined as follows.

If it is called on a Boolean combination $\beta$ of proposition letters (base of the recursion), $VAL(\beta, d)$ evaluates $\beta$ over $d$ in the standard way. The evaluation can be performed in deterministic polynomial time, and if $VAL(\beta, d)$ returns $\top$, then there exists a track associated with $d$ (of length at most quadratic in $|W|$) that satisfies $\beta$.

If $\psi$ has the form $\psi' \vee \psi''$, the procedure non-determistically calls itself on $\psi'$ or $\psi''$ (the control construct **Either** $c_1$ **Or** $c_2$ **EndOr** denotes a non-deterministic choice between commands $c_1$ and $c_2$).

If $\psi$ has the form $\langle \mathrm{A} \rangle \psi'$ (resp., $\langle \overline{\mathrm{A}} \rangle \psi'$), the procedure looks for a new descriptor element for a track starting from the final state (resp., leading to the initial state) of the current descriptor element $d$. To this end, we use the procedure $\texttt{aDescrEl}(\mathcal{K}, v, \mathrm{FORW})$ (resp., $\texttt{aDescrEl}(\mathcal{K}, v, \mathrm{BACKW})$) which non-deterministically returns a descriptor element $(v'_{in}, S', v'_{fin})$, with $v'_{in} = v$ (resp., $v'_{fin} = v$), witnessed in $\mathcal{K}$ by exploring forward (resp., backward) the unravelling of $\mathcal{K}$ from $v'_{in}$ (resp., from $v'_{fin}$). Its complexity is polynomial in $|W|$, since it needs to examine the unravelling of $\mathcal{K}$ from $v$ up to depth $2 + |W|^2$.

If $\psi$ has the form $\langle \mathrm{B} \rangle \psi'$, the procedure looks for a new descriptor element $d_1$ and eventually calls itself on $\psi'$ and $d_1$ only if the current descriptor element $d$ results from the "concatenation" of $d_1$ with a suitable descriptor element $d_2$: if $d_1 = (v'_{in}, S', v'_{fin})$ and $d_2 = (v''_{in}, S'', v''_{fin})$, then $\texttt{concat}(d_1, d_2)$ returns the descriptor element $(v'_{in}, S' \cup \{v'_{fin}, v''_{in}\} \cup S'', v''_{fin})$. Notice that if $\rho_1$ and $\rho_2$ are tracks associated with $d_1$ and $d_2$, respectively, then $\rho_1 \cdot \rho_2$ is associated with $\texttt{concat}(d_1, d_2)$.

The following theorem proves soundness and completeness of the `Check` procedure.

**Theorem 5.2.** *For any $\exists HS[A, \overline{A}, B, E]$ formula $\psi$ and any witnessed descriptor element $d = (v_{in}, S, v_{fin})$, the procedure $\texttt{Check}(\mathcal{K}, \psi, d)$ has a successful computation iff there exists a track $\rho$ associated with $d$ such that $\mathcal{K}, \rho \models \psi$.*

*Proof.* (Soundness) The proof is by induction on the structure of $\psi$. In the proof we assume that $d$ is $(v_{in}, S, v_{fin})$.

- $\psi$ is a boolean combination of propositions $\beta$: let $\rho$ be a witness track for $d$; if $\texttt{check}(\mathcal{K}, \beta, d)$ has a successful computation it means that $VAL(\beta, d)$ is true, hence $\mathcal{K}, \rho \models \psi$.

- $\psi = \varphi_1 \vee \varphi_2$: if $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation, it follows that for some $i \in \{1, 2\}$, $\texttt{check}(\mathcal{K}, \varphi_i, d)$ has a successful computation. By inductive hypothesis, there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ described by $d$ such that $\mathcal{K}, \rho \models \varphi_i$. Thus $\mathcal{K}, \rho \models \varphi_1 \vee \varphi_2$.

- $\psi = \langle \mathrm{A} \rangle \varphi$: if $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation, then there exists a witnessed $d' = (v'_{in}, S', v'_{fin})$ with $v'_{in} = v_{fin}$, such that $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. By inductive hypothesis, there exists a track $\rho'$ described by $d'$ such that $\mathcal{K}, \rho' \models \varphi$. If $\rho$ is a track described by $d$ (which is witnessed by hypothesis), we have $\mathrm{lst}(\rho) = \mathrm{fst}(\rho') = v_{fin}$ and, by definition, $\mathcal{K}, \rho \models \psi$.

- $\psi = \langle \mathrm{B} \rangle \varphi$: if $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation, there are two possible cases:

  - there exists $d' = (v_{in}, S', v'_{fin})$ witnessed by a track with $(v'_{fin}, v_{fin}) \in \delta$ such that $(v_{in}, S' \cup \{v'_{fin}\}, v_{fin}) = d$ and $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. By inductive hypothesis, there exists a track $\rho'$ described by $d'$ such that $\mathcal{K}, \rho' \models \varphi$. Hence $\mathcal{K}, \rho' \cdot v_{fin} \models \psi$ and $\rho' \cdot v_{fin}$ is associated with $d$.

**Algorithm 4** `Check`$(\mathcal{K},\psi,(v_{in},S,v_{fin}))$

**if** $\psi=\beta$ **then** ◁ *$\beta$ is a Boolean combination of propositions*
  **if** $VAL(\beta,(v_{in},S,v_{fin}))=\top$ **then**
    Yes
  **else**
    No
**else if** $\psi=\varphi_1\vee\varphi_2$ **then**
  **Either**
    **return** `Check`$(\mathcal{K},\varphi_1,(v_{in},S,v_{fin}))$
  **Or**
    **return** `Check`$(\mathcal{K},\varphi_2,(v_{in},S,v_{fin}))$
  **EndOr**
**else if** $\psi=\langle\mathrm{A}\rangle\varphi$ **then**
  $(v_{fin},S',v'_{fin})\leftarrow$ `aDescrEl`$(\mathcal{K},v_{fin},\mathrm{FORW})$
  **return** `Check`$(\mathcal{K},\varphi,(v_{fin},S',v'_{fin}))$
**else if** $\psi=\langle\overline{\mathrm{A}}\rangle\varphi$ **then**
  $(v'_{in},S',v_{in})\leftarrow$ `aDescrEl`$(\mathcal{K},v_{in},\mathrm{BACKW})$
  **return** `Check`$(\mathcal{K},\varphi,(v'_{in},S',v_{in}))$
**else if** $\psi=\langle\mathrm{B}\rangle\varphi$ **then**
  $(v'_{in},S',v'_{fin})\leftarrow$ `aDescrEl`$(\mathcal{K},v_{in},\mathrm{FORW})$ ◁ $v'_{in}=v_{in}$
  **Either**
    **if** $(v'_{in},S'\cup\{v'_{fin}\},v_{fin})=(v_{in},S,v_{fin})$ and $(v'_{fin},v_{fin})$ is an edge of $\mathcal{K}$ **then**
      **return** `Check`$(\mathcal{K},\varphi,(v'_{in},S',v'_{fin}))$
    **else**
      No
  **Or**
    $(v''_{in},S'',v''_{fin})\leftarrow$ `aDescrEl`$(\mathcal{K},v''_{in},\mathrm{FORW})$, where $(v'_{fin},v''_{in})$ is an edge of $\mathcal{K}$ non-deterministically chosen
    **if** `concat`$\left((v'_{in},S',v'_{fin}),(v''_{in},S'',v''_{fin})\right)=(v_{in},S,v_{fin})$ **then**
      **return** `Check`$(\mathcal{K},\varphi,(v'_{in},S',v'_{fin}))$
    **else**
      No
  **EndOr**
**else if** $\psi=\langle\mathrm{E}\rangle\varphi$ **then**
  $(v'_{in},S',v'_{fin})\leftarrow$ `aDescrEl`$(\mathcal{K},v_{fin},\mathrm{BACKW})$ ◁ $v'_{fin}=v_{fin}$
  **Either**
    **if** $(v_{in},\{v'_{in}\}\cup S',v'_{fin})=(v_{in},S,v_{fin})$ and $(v_{in},v'_{in})$ is an edge of $\mathcal{K}$ **then**
      **return** `Check`$(\mathcal{K},\varphi,(v'_{in},S',v'_{fin}))$
    **else**
      No
  **Or**
    $(v''_{in},S'',v''_{fin})\leftarrow$ `aDescrEl`$(\mathcal{K},v''_{fin},\mathrm{BACKW})$, where $(v''_{fin},v'_{in})$ is an edge of $\mathcal{K}$ non-deterministically chosen
    **if** `concat`$\left((v''_{in},S'',v''_{fin}),(v'_{in},S',v'_{fin})\right)=(v_{in},S,v_{fin})$ **then**
      **return** `Check`$(\mathcal{K},\varphi,(v'_{in},S',v'_{fin}))$
    **else**
      No
  **EndOr**

– there exist $d' = (v_{in}, S', v'_{fin})$ witnessed by a track and $d'' = (v''_{in}, S'', v''_{fin})$ witnessed by a track such that $(v'_{fin}, v''_{in}) \in \delta$, $\texttt{concat}(d', d'') = d$ and $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. By induction hypothesis, there exists a track $\rho'$ described by $d'$ such that $\mathcal{K}, \rho' \models \varphi$. Therefore $\mathcal{K}, \rho' \cdot \rho'' \models \psi$, where $\rho''$ is any track associated with $d''$, and $\rho' \cdot \rho''$ is associated with $d$.

The case for $\psi = \langle\overline{\text{A}}\rangle\varphi$ (resp. $\psi = \langle\text{E}\rangle\varphi$) can be treated as $\psi = \langle\text{A}\rangle\varphi$ (resp. $\psi = \langle\text{B}\rangle\varphi$).

(Completeness) The proof is by induction on the structure of $\psi$.

- $\psi$ is a boolean combination of propositions $\beta$: if $\rho$ is described by $d$ and it holds that $\mathcal{K}, \rho \models \beta$, then $VAL(\beta, d) = \top$, thus $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \varphi_1 \vee \varphi_2$: if there exists a track $\rho$ described by $d$ such that $\mathcal{K}, \rho \models \varphi_1 \vee \varphi_2$, then $\mathcal{K}, \rho \models \varphi_i$, for some $i \in \{1, 2\}$. By inductive hypothesis, $\texttt{check}(\mathcal{K}, \varphi_i, d)$ has a successful computation, hence $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \langle\text{A}\rangle\varphi$: if there exists a track $\rho$ described by $d$ such that $\mathcal{K}, \rho \models \langle\text{A}\rangle\varphi$, by definition there exists a track $\overline{\rho}$ with $\text{fst}(\overline{\rho}) = \text{lst}(\rho) = v_{fin}$ such that $\mathcal{K}, \overline{\rho} \models \varphi$. If $d' = (v_{fin}, S', v'_{fin})$ is the descriptor element of $\overline{\rho}$, then by inductive hypothesis $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. Now there is a computation where the non-deterministic call to $\texttt{aDescrEl}(\mathcal{K}, v_{fin}, \text{FORW})$ returns the descriptor element $d'$ of $\overline{\rho}$ and $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \langle\text{B}\rangle\varphi$: if there exists a track $\rho$ described by $d$ such that $\mathcal{K}, \rho \models \langle\text{B}\rangle\varphi$, there are two possible cases:

  – $\mathcal{K}, \overline{\rho} \models \varphi$ with $\rho = \overline{\rho} \cdot v_{fin}$ for $\overline{\rho} \in \text{Trk}_{\mathcal{K}}$. If $d' = (v_{in}, S', v'_{fin})$ describes $\overline{\rho}$, by inductive hypothesis $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. Now, there is a computation where $\texttt{aDescrEl}(\mathcal{K}, v_{in}, \text{FORW})$ returns $d'$; clearly, $(v'_{fin}, v_{fin}) \in \delta$ and $(v_{in}, S' \cup \{v'_{fin}\}, v_{fin}) = d$, and $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation.

  – $\mathcal{K}, \overline{\rho} \models \varphi$ with $\rho = \overline{\rho} \cdot \tilde{\rho}$ for some $\overline{\rho}, \tilde{\rho} \in \text{Trk}_{\mathcal{K}}$. Let $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v''_{in}, S'', v''_{fin})$ be the descriptor elements of $\overline{\rho}$ and $\tilde{\rho}$, respectively. Obviously we have $\texttt{concat}(d', d'') = d$.
  By inductive hypothesis, $\texttt{check}(\mathcal{K}, \varphi, d')$ has a successful computation. Since $\overline{\rho}$ and $\tilde{\rho}$ are witnessed, there exists a computation where the calls to $\texttt{aDescrEl}(\mathcal{K}, v_{in}, \text{FORW})$ and to $\texttt{aDescrEl}(\mathcal{K}, v''_{in}, \text{FORW})$ return non-deterministically $d'$ and $d''$, respectively, and $(v'_{fin}, v''_{in}) \in \delta$ is chosen non-deterministically. Hence $\texttt{check}(\mathcal{K}, \psi, d)$ has a successful computation.

The case for $\psi = \langle\overline{\text{A}}\rangle\varphi$ (respectively $\psi = \langle\text{E}\rangle\varphi$) can be treated as $\psi = \langle\text{A}\rangle\varphi$ (respectively $\psi = \langle\text{B}\rangle\varphi$). □

The procedure $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ of Algorithm 5 has a successful computation iff $\mathcal{K} \not\models \psi$, where $\psi$ is a $\forall HS[A, \overline{A}, B, E]$ formula, $\texttt{to}\exists HS[A, \overline{A}, B, E](\neg\psi)$ is the $\exists HS[A, \overline{A}, B, E]$ formula equivalent to $\neg\psi$, and $w_0$ is the initial state of $\mathcal{K}$.

If $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ has a successful computation it means that there exists a witnessed descriptor element $d = (v_{in}, S, v_{fin})$, where $v_{in}$ is the initial state of

**Algorithm 5** `ProvideCounterex`$(\mathcal{K},\psi)$

1: $(v_{in},S,v_{fin}) \leftarrow$ `aDescrEl`$(\mathcal{K},w_0,\text{FORW})$ ◁ $v_{in}=w_0$
2: **return** `Check`$(\mathcal{K},$`to`$\exists HS[A,\overline{A},B,E](\neg\psi),(v_{in},S,v_{fin}))$

$\mathcal{K}$, such that `check`$(\mathcal{K},$`to`$\exists HS[A,\overline{A},B,E](\neg\psi),d)$ has a successful computation. So there exists a track $\rho$ associated with $d$ such that $\mathcal{K},\rho\models\neg\psi$. Thus $\mathcal{K}\not\models\psi$.

On the other hand, if $\mathcal{K}\not\models\psi$ then there exists an initial track $\rho$ such that $\mathcal{K},\rho\models\neg\psi$. Let $d$ be the descriptor element of $\rho$: `check`$(\mathcal{K},$`to`$\exists HS[A,\overline{A},B,E](\neg\psi),d)$ has a successful computation. Since $d$ is witnessed by an initial track, some non-deterministic instance of `aDescrEl`$(\mathcal{K},w_0,\text{FORW})$ returns $d$; hence `ProvideCounterex`$(\mathcal{K},\psi)$ has a successful computation.

As for the complexity, `ProvideCounterex`$(\mathcal{K},\psi)$ runs in non-deterministic polynomial time (it is in NP) since the number of recursive invocations of the procedure `Check` is $O(|\psi|)$ and each invocation requires time polynomial in $|W|$ while generating descriptor elements. As a consequence, the model checking problem for $\forall HS[A,\overline{A},B,E]$ belongs to the complexity class coNP.

We conclude the section by proving that the model checking problem for the fragment $\forall HS[A,\overline{A},B,E]$ is coNP-complete. Such a result is an easy corollary of the following theorem.

**Theorem 5.3.** *Let $\mathcal{K}$ be a finite Kripke structure and $\beta\in HS$[Prop] be a Boolean combination of proposition letters. The problem of deciding whether $\mathcal{K}\not\models\beta$ is NP-hard (under a LOGSPACE reduction).*

*Proof.* We provide a LOGSPACE reduction from the NP-complete SAT problem to our problem. Let $\beta$ be a Boolean formula over a set of variables $\{x_1,\ldots,x_n\}$. We build a Kripke structure $\mathcal{K}=(\mathcal{AP},W,\delta,\mu,w_0)$, where:

- $\mathcal{AP}=\{x_1,\ldots,x_n\}$,
- $W=\{w_0\}\cup\{w_i^\ell \mid \ell\in\{\top,\bot\},\ 1\le i\le n\}$,
- $\delta=\{(w_0,w_1^\top),(w_0,w_1^\bot)\}\cup\{(w_i^\ell,w_{i+1}^m)\mid \ell,m\in\{\top,\bot\},\ 1\le i\le n-1\}\cup$ $\{(w_n^\top,w_n^\top),(w_n^\bot,w_n^\bot)\}$,
- $\mu(w_0)=\mathcal{AP}$,
- for all $1\le i\le n$, $\mu(w_i^\top)=\mathcal{AP}$ and $\mu(w_i^\bot)=\mathcal{AP}\setminus\{x_i\}$.

Figure 5.2 provides an example of such a Kripke structure for $\mathcal{AP}=\{x_1,x_2,x_3,x_4\}$.

It is immediate to see that any initial track $\rho$ of any length induces a truth assignment to the variables of $\mathcal{AP}$: for any $x_i\in\mathcal{AP}$, $x_i$ evaluates to $\top$ iff $x_i\in\bigcap_{w\in\text{states}(\rho)}\mu(w)$. Vice versa, for any possible truth assignment to the variables in $\mathcal{AP}$, there exists an initial track $\rho$ that induces such an assignment: we include in the track the state $x_i^\top$ if $x_i$ is assigned to $\top$, $x_i^\bot$ otherwise.

Let $\gamma=\neg\beta$. It holds that $\beta$ is satisfiable iff there exists an initial track $\rho\in\text{Trk}_\mathcal{K}$ such that $\mathcal{K},\rho\models\beta$, that is, iff $\mathcal{K}\not\models\gamma$. To conclude, it suffices to observe that $\mathcal{K}$ can be built with logarithmic working space. □

It immediately follows that checking whether $\mathcal{K}\not\models\beta$ for $\beta\in HS$[Prop] is NP-complete, so model checking for formulas of $HS$[Prop] is coNP-complete. Moreover,

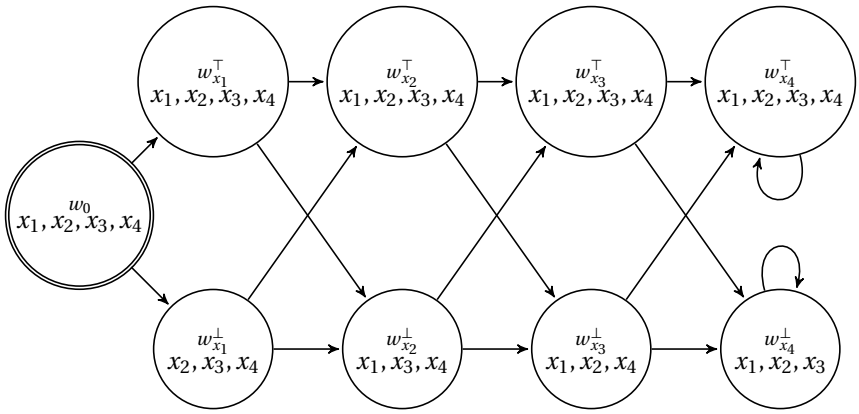



**Figure 5.2:** Kripke structure associated with a SAT formula with variables $x_1$, $x_2$, $x_3$, $x_4$.

since a Boolean combination of proposition letters in $HS$[Prop] is also a formula of $\forall HS[A, \overline{A}, B, E]$, $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ is at least as hard as checking whether $\mathcal{K} \not\models \beta$ for $\beta \in HS$[Prop]. Thus, $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ is NP-complete, and hence the model checking problem for $\forall HS[A, \overline{A}, B, E]$ is coNP-complete.

Finally, from the lower bound for $HS$[Prop], it immediately follows that model checking for $HS[A, \overline{A}]$ is coNP-hard (and we already know it is in PSPACE from Section 4.3).

## 5.2 The fragment $HS[A, \overline{A}, \overline{B}, \overline{E}]$

As we said, the model checking algorithm for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ of Section 4.3 can check formulas of $HS[A, \overline{A}, \overline{B}, \overline{E}]$ in polynomial (not exponential) space. Here, we prove that such an algorithm is asymptotically optimal by showing that model checking for $HS[A, \overline{B}]$ is a PSPACE-hard problem (Theorem 5.4). PSPACE-completeness of $HS[A, \overline{A}, \overline{B}, \overline{E}]$ (and $HS[A, \overline{B}]$) immediately follows. As a by-product, we have that model checking for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ is PSPACE-hard, as well.

**Theorem 5.4.** *The model checking problem for $HS[A, \overline{B}]$ formulas over finite Kripke structures is PSPACE-hard (under LOGSPACE reductions).*

*Proof.* We provide a reduction from the problem of determining the truth of a *fully-quantified* boolean formula in *prenex normal form* (the QBF problem, for short), which is known to be PSPACE-complete (see [Sip12, Pap94]), to the model checking problem for $HS[A, \overline{B}]$ formulas over Kripke structures. We consider a QBF formula $\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots, x_1)$, where $Q_i \in \{\exists, \forall\}$, $i = 1, \cdots, n$ and $\phi(x_n, x_{n-1}, \cdots, x_1)$ is a quantifier free Boolean formula. Let $Var = \{x_n, \ldots, x_1\}$ be the set of Boolean variables of $\psi$. We define the following Kripke structure $\mathcal{K}_{QBF}^{Var} = (\mathcal{AP}, W, \delta, \mu, w_0)$:

- $\mathcal{AP} = Var \cup \{start\} \cup \{x_{i\,aux} \mid 1 \leq i \leq n\}$;
- $W = \{w_{x_i}^{\ell} \mid 1 \leq i \leq n, \ell \in \{\bot_1, \bot_2, \top_1, \top_2\}\} \cup \{w_0, w_1, sink\}$.

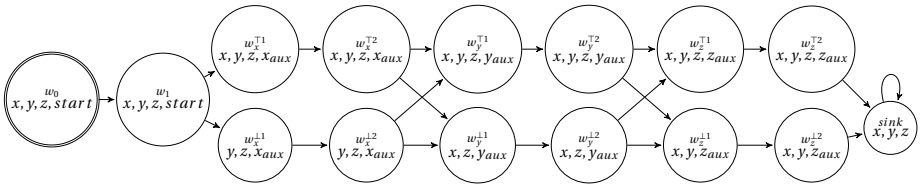



**Figure 5.3:** Kripke structure $\mathcal{K}_{QBF}^{x,y,z}$ associated with a QBF formula with variables $x$, $y$, $z$.

- For $n=0$, $\delta = \{(w_0,w_1),(w_1,sink),(sink,sink)\}$.
  For $n \geq 1$,

$$\begin{aligned}\delta = \{&(w_0,w_1),(w_1,w_{x_n}^{\top_1}),(w_1,w_{x_n}^{\bot_1}),(sink,sink)\} \cup \\ &\{(w_{x_i}^{\top_1},w_{x_i}^{\top_2}),(w_{x_i}^{\bot_1},w_{x_i}^{\bot_2}) \mid 1 \leq i \leq n\} \cup \\ &\{(w_{x_i}^{\ell},w_{x_{i-1}}^{m}) \mid \ell \in \{\bot_2,\top_2\},\ m \in \{\bot_1,\top_1\},\ n \leq i \leq 2\} \cup \\ &\{(w_{x_1}^{\top_2},sink),(w_{x_1}^{\bot_2},sink)\};\end{aligned}$$

- $\mu(w_0) = \mu(w_1) = Var \cup \{start\}$;
  $\mu(w_{x_i}^{\ell}) = Var \cup \{x_{i\,aux}\}$, for $1 \leq i \leq n$ and $\ell \in \{\top_1,\top_2\}$;
  $\mu(w_{x_i}^{\ell}) = (Var \setminus \{x_i\}) \cup \{x_{i\,aux}\}$, for $1 \leq i \leq n$ and $\ell \in \{\bot_1,\bot_2\}$;
  $\mu(sink) = Var$.

For an example of such a Kripke structure for $Var = \{x,y,z\}$, see Figure 5.3.
From $\psi$, we derive the following $HS[A,\overline{B}]$ formula $\xi = start \rightarrow \xi_n$, where

$$\xi_i = \begin{cases} \phi(x_n, x_{n-1}, \cdots x_1) & i = 0 \\ \langle \overline{\mathrm{B}} \rangle \big( (\langle \mathrm{A} \rangle x_{i\,aux}) \wedge \xi_{i-1} \big) & i > 0 \wedge Q_i = \exists \\ [\overline{B}] \big( (\langle \mathrm{A} \rangle x_{i\,aux}) \rightarrow \xi_{i-1} \big) & i > 0 \wedge Q_i = \forall \end{cases}.$$

Clearly, $\mathcal{K}_{QBF}^{Var}$ and $\xi$ can be built by using logarithmic working space.

It can be proved (the proof is in the appendix, Section A.3.1) that the quantified Boolean formula $\psi$ is true if and only if $\mathcal{K}_{QBF}^{Var} \models \xi$. □

# 6 Conclusions

In this final chapter we summarize the results achieved so far (see Figure 6.1 for a visual report).

In Chapter 3 we have proved that the (full) HS model checking problem is decidable with a non-elementary upper bound, by following [MMPP14]. The cornerstone of the decidability proof is the notion of $BE_k$-descriptor, which allows us to obtain a finite representation of a possibly infinite set of equivalent tracks. Since the number of $BE_k$-descriptors is always finite, the decidability of the model checking problem for HS over finite Kripke structures easily follows. In addition, we have provided a notion, correspondence between descriptors, which precisely captures $k$-equivalence between tracks. Finally, we have proved that the HS model checking problem is EXPSPACE-hard, provided that a succinct encoding of formulas is used (otherwise we can only prove that it is PSPACE-hard).

In Chapter 4 we have showed that formulas of the fragment $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ (and of the symmetric $HS[A, \overline{A}, E, \overline{B}, \overline{E}]$) can be checked with exponential working space. The proposed algorithm rests on a contraction method that allows us to restrict the verification of the input formula to a finite subset of tracks of bounded size, called track representatives. These are determined by evaluating $k$-indistinguishability among descriptor elements, a notion which allows both to determine if two tracks are associated with the same $B_k$-descriptor (without building it explicitly), and to calculate an upper bound to the length of representatives. In the end, we have proved that the model checking problem for $HS[A, \overline{A}, B, \overline{B}, \overline{E}]$ is NEXP-hard, provided that a succinct encoding of formulas is used.

In Chapter 5 we have clarified the complexity picture of some HS fragments, $\forall HS[A, \overline{A}, B, E]$, $HS[A, \overline{A}]$ and $HS[A, \overline{A}, \overline{B}, \overline{E}]$, whose formulas are expressive enough to capture typical interval properties of transition systems, while keeping the model checking complexity lower, comparable to that of point-based temporal logics.

Clearly, our analysis of HS model checking is innovative, thus many problems are still open. For example, the exact complexity of the full logic is unknown: it ranges from PSPACE to non-ELEMENTARY (in the non-succinct case), quite a big gap! Even some apparently easy fragments are still unsolved (e.g., $HS[A, \overline{A}]$ is in between coNP and PSPACE). It is evident that if $B$ and $E$ modalities are *not* involved, model checking for such fragments is in PSPACE. As soon as one of these two is added, the expressiveness of fragments increases (e.g., metric properties about the length of

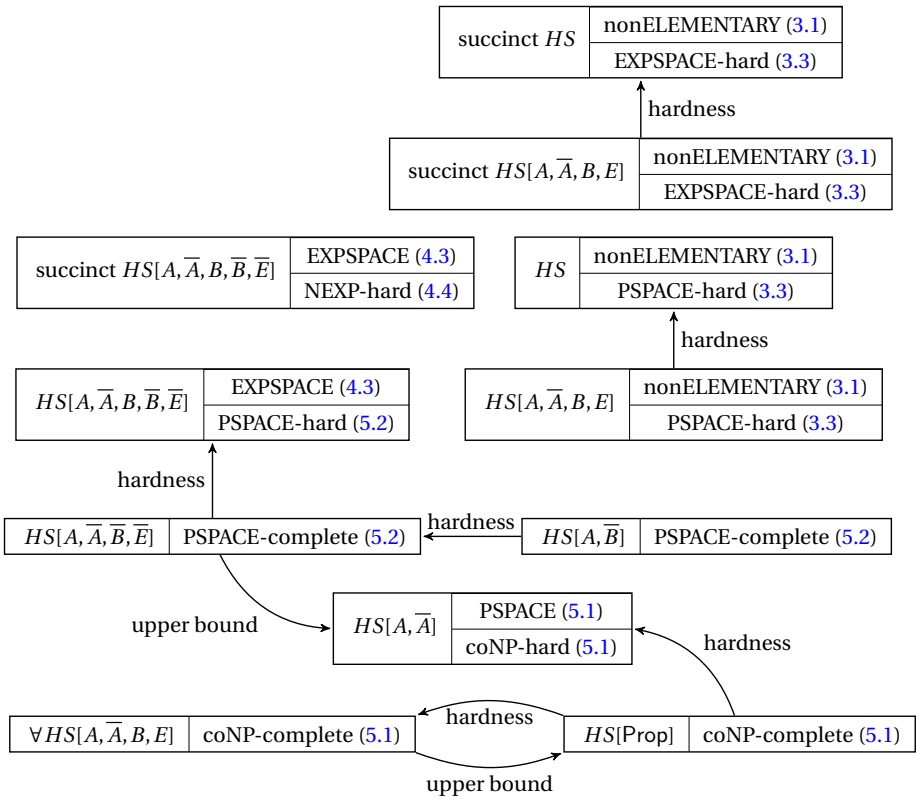



**Figure 6.1:** Complexity of the model checking problem for the HS fragments we have studied. The number of the section in which we deal with a fragment is shown next to it.

tracks can be expressed by exploiting $B$ or $E$ operators), but so too does complexity: $HS[A,\overline{A},B,\overline{B},\overline{E}]$, $HS[A,\overline{A},E,\overline{B},\overline{E}]$ and full HS hardly appear to be in PSPACE (but this is just a conjecture).

Finally, we have always understood the homogeneity assumption, starting from the very definition of HS semantics over induced abstract interval models. However, in this way we can not benefit of the full expressive power of HS. Thus an obvious research direction would be to redefine HS semantics relaxing such an hypothesis and then study if and how complexity and decidability results change, and/or adapt the current notions and machinery to the new, more expressive semantics.

# A Appendix

## Contents



## A.1 Proofs of Chapter 3

### A.1.1 Proof of Lemma 3.9

*Proof.* The proof is by induction on $n \geq 0$. Let $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ be the $BE_k$-descriptors for $\rho$ and $\rho'$, respectively.

Base case ($n = 0$). Since $\mathcal{K}, \rho \models p \iff \mathcal{K}, \rho' \models p$, for any $p \in \mathcal{AP}$, the roots of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are labelled by the same set of proposition letters and the descriptors are corresponding up to depth 0.

Inductive step ($n \geq 1$). We preliminarily show that if $\rho$ and $\rho'$ are $k$-equivalent with respect to all formulas $\varphi$ with $\mathrm{Nest}(\varphi) \leq n$, then for any track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$, there is a track $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, such that, for *all* HS formulas $\psi$, with $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$, $\mathcal{K}, \overline{\rho} \models \psi \iff \mathcal{K}, \overline{\rho}' \models \psi$. The proof is by contradiction. Suppose that there is a track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$, such that, for all tracks $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, there exists a formula $\psi$, with $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$, such that $\mathcal{K}, \overline{\rho} \models \psi$ and $\mathcal{K}, \overline{\rho}' \not\models \psi$. Let $H$ be the set of those tracks $\hat{\rho}$ such that $\mathrm{fst}(\hat{\rho}) = \mathrm{lst}(\rho')$. $H$ can be partitioned into a finite number

of classes, say $s \geq 1$, each one containing $k$-descriptor equivalent tracks of $H$ (recall that $k$-descriptor equivalence is an equivalence relation of finite index). Now, let $\{\overline{\rho}'_1, \overline{\rho}'_2, \ldots, \overline{\rho}'_s\}$ be a set of track representatives, chosen one for each equivalence class induced by $\sim_k$ on $H$ (for all $1 \leq i < j \leq s$, $\overline{\rho}'_i$ and $\overline{\rho}'_j$ have distinct $BE_k$-descriptors). By Theorem 3.2, tracks which are $k$-descriptor equivalent satisfy the same set of formulas $\psi'$, with $\mathrm{Nest}_{\mathrm{BE}}(\psi') \leq k$. So there are formulas $\psi_1, \ldots, \psi_s$ such that, for all $1 \leq i \leq s$, $\mathrm{Nest}(\psi_i) \leq n-1$, $\mathrm{Nest}_{\mathrm{BE}}(\psi_i) \leq k$, $\mathcal{K}, \overline{\rho} \models \psi_i$, and $\mathcal{K}, \overline{\rho}'_i \not\models \psi_i$. It easily follows that $\mathcal{K}, \overline{\rho} \models \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s$ and, for all $1 \leq i \leq s$, $\mathcal{K}, \overline{\rho}'_i \models \neg\psi_1 \vee \neg\psi_2 \vee \cdots \vee \neg\psi_s$. Hence, $\mathcal{K}, \rho \models \langle \mathrm{A} \rangle(\psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s)$ and $\mathcal{K}, \rho' \models [A](\neg\psi_1 \vee \neg\psi_2 \vee \cdots \vee \neg\psi_s)$, that is, $\mathcal{K}, \rho' \not\models \langle \mathrm{A} \rangle(\psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s)$, which is a contradiction.

Thus, we have proved that for any track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$, there exists a track $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, such that, for *all* HS formulas $\psi$, with $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k$, $\mathcal{K}, \overline{\rho} \models \psi \iff \mathcal{K}, \overline{\rho}' \models \psi$. By the inductive hypothesis, $\overline{\rho}$ and $\overline{\rho}'$ are associated with corresponding $BE_k$-descriptors up to depth $n-1$. Symmetrically, we can show that for any track $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, there exists $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$, such that $\overline{\rho}'$ and $\overline{\rho}$ are associated with corresponding $BE_k$-descriptors up to depth $n-1$. In this way, we have proved the condition for modality $A$ of Definition 3.8. The conditions for modalities $\overline{A}$, $\overline{B}$, and $\overline{E}$ can be proved in a very similar way. In particular, as a consequence of the fact that $\rho$ and $\rho'$ are $k$-equivalent with respect to all HS formulas $\psi$, with $\mathrm{Nest}(\psi) \leq n$ and $n \geq 1$, it holds that $\mathcal{K}, \rho \models \langle \overline{\mathrm{A}} \rangle \top \iff \mathcal{K}, \rho' \models \langle \overline{\mathrm{A}} \rangle \top$. It follows that $\mathcal{D}_{BE_k}$ has an $\overline{A}$-successor if and only if $\mathcal{D}'_{BE_k}$ has one. The same holds for $\overline{E}$-successors.

Let us now consider the condition for modality $B$ of Definition 3.8.

First of all, we show that for any track $\overline{\rho} \in \mathrm{Pref}(\rho)$, there exists a track $\overline{\rho}' \in \mathrm{Pref}(\rho')$ such that for *all* formulas $\psi$ of HS, having $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k-1$, $\mathcal{K}, \overline{\rho} \models \psi \iff \mathcal{K}, \overline{\rho}' \models \psi$. The proof is again by contradiction. Suppose that there exists a track $\overline{\rho} \in \mathrm{Pref}(\rho)$ such that, for all tracks $\overline{\rho}' \in \mathrm{Pref}(\rho')$, there exists a formula $\psi$, with $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k-1$, such that $\mathcal{K}, \overline{\rho} \models \psi$ and $\mathcal{K}, \overline{\rho}' \not\models \psi$. Now, let us consider the tracks $\overline{\rho}'_1, \overline{\rho}'_2, \cdots, \overline{\rho}'_s$ (for some $s \in \mathbb{N}$) which are prefixes of $\rho'$ and are associated with distinct subtrees of depth $k-1$ of the $BE_k$-descriptor for $\rho'$ (the number of these tracks is obviously finite). So there are formulas $\psi_1, \ldots, \psi_s$ such that, for all $1 \leq i \leq s$, $\mathrm{Nest}(\psi_i) \leq n-1$, $\mathrm{Nest}_{\mathrm{BE}}(\psi_i) \leq k-1$, $\mathcal{K}, \overline{\rho} \models \psi_i$, and $\mathcal{K}, \overline{\rho}'_i \not\models \psi_i$. Thus, $\mathcal{K}, \overline{\rho} \models \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s$ and for all $i$, $\mathcal{K}, \overline{\rho}'_i \models \neg\psi_1 \vee \neg\psi_2 \vee \cdots \vee \neg\psi_s$. Hence we have $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle(\psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s)$ and $\mathcal{K}, \rho' \models [B](\neg\psi_1 \vee \neg\psi_2 \vee \cdots \vee \neg\psi_s)$, that is $\mathcal{K}, \rho' \not\models \langle \mathrm{B} \rangle(\psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s)$, which leads to a contradiction.

We have proved that for any track $\overline{\rho} \in \mathrm{Pref}(\rho)$, there exists a track $\overline{\rho}' \in \mathrm{Pref}(\rho')$ such that, for *all* formulas $\psi$ of HS, having $\mathrm{Nest}(\psi) \leq n-1$ and $\mathrm{Nest}_{\mathrm{BE}}(\psi) \leq k-1$, $\mathcal{K}, \overline{\rho} \models \psi \iff \mathcal{K}, \overline{\rho}' \models \psi$. By the inductive hypothesis, $\overline{\rho}$ and $\overline{\rho}'$ are associated with corresponding $BE_{k-1}$-descriptors up to depth $n-1$. Symmetrically, we can show that for any track $\overline{\rho}' \in \mathrm{Pref}(\rho')$, there exists a track $\overline{\rho} \in \mathrm{Pref}(\rho)$ such that $\overline{\rho}'$ and $\overline{\rho}$ are associated with corresponding $BE_{k-1}$-descriptors up to depth $n-1$.

In this way, we have proved the condition for modality $B$ of Definition 3.8. The condition for modality $E$ can be proved in a symmetrical way. □

### A.1.2 Proof of Lemma 3.10

*Proof.* The proof is by induction on $n \geq 0$.

Base case ($n = 0$). Consider the descriptors $\mathcal{D}_{BE_k}$, $\mathcal{D}'_{BE_k}$, $\mathcal{D}_{BE_k}|_{k-1}$, and $\mathcal{D}'_{BE_k}|_{k-1}$. Since the roots of $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are labelled by the same set of proposition letters,

the roots of $\mathcal{D}_{BE_k}|_{k-1}$ and $\mathcal{D}'_{BE_k}|_{k-1}$ are labelled by the same set of proposition letters as well.

Inductive step ($n > 0$). Let $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$ be two witnesses for $\mathcal{D}_{BE_k}$ and for $\mathcal{D}'_{BE_k}$, respectively (and thus for $\mathcal{D}_{BE_k}|_{k-1}$ and $\mathcal{D}'_{BE_k}|_{k-1}$, respectively). Consider a track $\tilde{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\tilde{\rho}) = \mathrm{lst}(\rho)$. The $BE_k$-descriptor $\mathcal{D}_{\tilde{BE}_k}$ for $\tilde{\rho}$ is an $A$-successor of $\mathcal{D}_{BE_k}$, and $\mathcal{D}_{\tilde{BE}_k}|_{k-1}$ is an $A$-successor of $\mathcal{D}_{BE_k}|_{k-1}$. Since $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are corresponding up to depth $n$, there exists a track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho')$, described by $\overline{\mathcal{D}_{BE_k}}$, such that $\mathcal{D}_{\tilde{BE}_k}$ and $\overline{\mathcal{D}_{BE_k}}$ are corresponding up to depth $n-1$. By the inductive hypothesis, $\mathcal{D}_{\tilde{BE}_k}|_{k-1}$ and $\overline{\mathcal{D}_{BE_k}}|_{k-1}$ are corresponding up to depth $n-1$ (and, obviously, $\overline{\mathcal{D}_{BE_k}}|_{k-1}$ is an $A$-successor of $\mathcal{D}'_{BE_k}|_{k-1}$).

Let us consider now a track $\hat{\rho}$, with $(\mathrm{lst}(\rho), \mathrm{fst}(\hat{\rho})) \in \delta$ and $\rho \cdot \hat{\rho} \in \mathrm{Trk}_{\mathcal{K}}$. The $BE_k$-descriptor $\mathcal{D}_{\hat{BE}_k}$ of $\rho \cdot \hat{\rho}$ is a $\overline{B}$-successor of $\mathcal{D}_{BE_k}$ and $\mathcal{D}_{\hat{BE}_k}|_{k-1}$ is a $\overline{B}$-successor of $\mathcal{D}_{BE_k}|_{k-1}$. Since $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are corresponding up to depth $n$, there exists a track $\check{\rho}$ such that $(\mathrm{lst}(\rho'), \mathrm{fst}(\check{\rho})) \in \delta$, $\rho' \cdot \check{\rho}$ is described by $\mathcal{D}_{\check{BE}_k}$, and $\mathcal{D}_{\hat{BE}_k}$ and $\mathcal{D}_{\check{BE}_k}$ are corresponding up to depth $n-1$. By the inductive hypothesis, $\mathcal{D}_{\hat{BE}_k}|_{k-1}$ and $\mathcal{D}_{\check{BE}_k}|_{k-1}$ are corresponding up to depth $n-1$ (and, obviously, $\mathcal{D}_{\check{BE}_k}|_{k-1}$ is a $\overline{B}$-successor of $\mathcal{D}'_{BE_k}|_{k-1}$).

Finally (only for cases with $k \geq 2$), let us consider a subtree of depth $k-2$ linked to the root of $\mathcal{D}_{BE_k}|_{k-1}$ via a $B$-edge. In this case, there exists (at least) a subtree of $\mathcal{D}_{BE_k}$, say $\mathcal{S}_{k-1}$, such that $\mathcal{S}_{k-1}|_{k-2}$ is the considered subtree of $\mathcal{D}_{BE_k}|_{k-1}$. Since $\mathcal{D}_{BE_k}$ and $\mathcal{D}'_{BE_k}$ are corresponding up to depth $n$, there exists a subtree $\mathcal{S}'_{k-1}$ of $\mathcal{D}'_{BE_k}$, connected to the root of $\mathcal{D}'_{BE_k}$ via a $B$-edge, corresponding to $\mathcal{S}_{k-1}$ up to depth $n-1$. By the inductive hypothesis $\mathcal{S}_{k-1}|_{k-2}$ and $\mathcal{S}'_{k-1}|_{k-2}$ are corresponding up to depth $n-1$ (the latter is a subtree of $\mathcal{D}'_{BE_k}|_{k-1}$ connected to the root of $\mathcal{D}'_{BE_k}|_{k-1}$ via a $B$-edge).

The remaining cases can be dealt with analogously. □

## A.2 Proofs of Chapter 4

### A.2.1 Proof of Lemma 4.1

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 $k \in \mathbb{N}$.

$k = 0$: if $\rho_1$ and $\rho_2$ are associated with the descriptor element $(v_{in}, S, v_{fin})$, and $\rho'_1$ and $\rho'_2$ with $(v'_{in}, S', v'_{fin})$, then $\rho_1 \cdot \rho'_1$ and $\rho_2 \cdot \rho'_2$ are both represented by the descriptor element $(v_{in}, S \cup \{v_{fin}, v'_{in}\} \cup S', v'_{fin})$.

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

- if $\rho$ is a proper prefix of $\rho_1$, since $\rho_1$ and $\rho_2$ have the same $B_k$-descriptor, there exists a prefix $\overline{\rho}$ of $\rho_2$ associated with the same subtree as $\rho$ of depth $k-1$ in the descriptor of $\rho_1$ (and $\rho_2$);
- for $\rho = \rho_1$, it holds that $\rho_1$ and $\rho_2$ have the same $B_{k-1}$-descriptor because they have the same $B_k$-descriptor;

- if $\rho$ 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 $\rho'_1$, 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 us 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 those of $\rho_1$ and $\rho_2$'s $B_{k-1}$-descriptor, (possibly) with the addition of the $B_{k-2}$-descriptor of $\rho_1$ (which is equal to that of $\rho_2$). 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 $\rho'_1$ of length at least 2, and $\rho'_1$ and $\rho'_2$ have the same $B_k$-descriptor, there exists a prefix $\tilde{\rho}_2$ of $\rho'_2$ associated with the same subtree of depth $k-1$ as $\tilde{\rho}_1$ (in the $B_k$-descriptor of $\rho'_1$). 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 $\rho_2 \cdot \rho'_2$ having the same $B_{k-1}$-descriptor. The inverse can be shown by symmetry. Thus $\mathcal{D}_{B_k}$ is equal to $\mathcal{D}'_{B_k}$. □

### A.2.2 Proof of Theorem 4.20

*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, $Q_{-2}(u+1) = \mathcal{C} \setminus \{d, d'\}$, $Q_{-1}(u) = \{d, d'\}$, $\emptyset = Q_0(u) = Q_1(u) = \cdots = Q_s(u)$, $c(u) >_{lex} c(u+1)$, 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 $d$. Therefore $d \in Q_{-1}(i-1)$ ($\rho_{ds}(u, i-1)$ 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 $d$ 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 $d$, $\rho_{ds}(j)$, $j < i-1$, 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 $j < i' < i$ s.t. $\rho_{ds}(i') = d'' \in \mathcal{C}$ 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 4.16 $\rho_{ds}(j')$ 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 4.14, 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 $\overline{t}$-indistinguishable, for $1 \leq \overline{t} \leq t$, 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 $j$, $\overline{t} = 1$ 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 $(\overline{t}-1)$-indistinguishable from another occurrence of $d'$ before $j$. Analogously to the case 5, $\rho_{ds}(i')$ must be $(\overline{t}-1)$-indistinguishable from the last occurrence of $d'$ before $j$, say $\rho_{ds}(j')$ with $j' < j$ (it's a consequence of Property 4.14). But two consecutive occurrences of $d'$ in $\rho_{ds}(j', i')$ must then be at most $(\overline{t}-1)$-indistinguishable (if all pairs of occurrences of $d'$ in $\rho_{ds}(j', i')$ were

$\overline{t}$-indistinguishable, $\rho_{ds}(i')$ and $\rho_{ds}(j')$ would be $\overline{t}$-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 $d' \in Q_{\overline{t}-2}(\ell-1)$ and $d' \in Q_{\overline{t}-1}(\ell)$. As a consequence, we have $d \in Q_{\overline{t}-1}(\ell), \cdots, d \in Q_{\overline{t}-1}(i-1)$ (all descriptor elements in $\rho_{ds}(j,i)$ are at least $(\overline{t}-1)$-indistinguishable from other occurrences before $j$) and finally $d \in Q_{\overline{t}}(i)$, implying that $c(i-1) >_{lex} c(i)$.

□

### A.2.3 Proof of Theorem 4.24

*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 4.18. So let's assume $k \geq 1$. The proof is by induction on $\ell = |\rho|$.

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

(Case $\ell > 2$) If in $\rho_{ds}$ there are no pairs of $k$-indistinguishable occurrences of some descriptor element, the termination criterion of Algorithm 1 can never be applied. Thus $\rho$ 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 $\rho$ can be split into 3 parts: $\rho_{ds} = \rho_{ds1} \cdot \rho_{ds2} \cdot \rho_{ds3}$ where $\rho_{ds1}$ ends with a Type-1 descriptor element and it does not contain pairs of $k$-indistinguishable occurrences of a descriptor element, $\rho_{ds2}$ is a subsequence of $\rho_{ds}$ associated with a cluster $\mathcal{C}$ of (Type-2) descriptor elements with at least a pair of $k$-indistinguishable occurrences of descriptor elements, and $\rho_{ds3}$ (if it is not the empty sequence) begins with a Type-1 descriptor element. Namely, $\rho_{ds2}$ is the "leftmost" subsequence of $\rho_{ds}$ 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 $i$ and $j$ 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 4.14, there exists a pair of indexes $i', j'$ 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 us consider the one with the lower index $i'$ (namely, precisely the pair which is found earlier by the unravelling algorithm). By Theorem 4.15, 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 4.2, 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 1 does not return $\tilde{\rho}_2$ and, due to the backtrack step, neither $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ is returned. But since $\text{lst}(\tilde{\rho}_1) = \text{lst}(\tilde{\rho}_2)$ ($\rho_{ds2}(j')$ 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 $\rho''$ of $\mathcal{K}$ is returned such that $\rho'$ and $\rho''$ have the same $B_k$-descriptor, and $|\rho''| \leq \tau(|W|, k)$. $\rho$ has in turn the same $B_k$-descriptor as $\rho''$. □

### A.2.4 Proof of Lemma 4.25

*Proof.* The proof is by induction on the structure of $\psi$. (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 4.24 the unravelling procedure returns $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\overline{\rho}) = \mathrm{fst}(\rho)$ and $\overline{\rho}$ and $\rho$ have the same $B_k$-descriptor, thus $\mathcal{K}, \overline{\rho} \models \varphi$. By inductive hypothesis, $\mathtt{Check}(\mathcal{K}, k, \varphi, \overline{\rho}) = 1$, hence $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$.

  Vice versa, if $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\tilde{\rho}) = \mathrm{fst}(\rho)$ and $\mathtt{Check}(\mathcal{K}, k, \varphi, \rho) = 1$. 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 the 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, $\mathtt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. Since all prefixes of $\tilde{\rho}$ are checked, $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. *Note that, by definition of descriptor, 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 $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, then for some track $\rho \in \mathrm{Pref}(\tilde{\rho})$, we have $\mathtt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. 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 $\rho$ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ for which $\mathcal{K}, \tilde{\rho} \cdot \rho \models \varphi$. If $|\rho| = 1$, since by inductive hypothesis $\mathtt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$, then $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. Otherwise, the unravelling algorithm returns a track $\overline{\rho}$ with the same $B_k$-descriptor as $\rho$. Thus, by the left extension Proposition 4.3, $\tilde{\rho} \cdot \rho$ 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) $\mathtt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \overline{\rho}) = 1$ implying that $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. *Notice that, given two tracks $\rho, \rho'$ of $\mathcal{K}$, if we are considering $\overline{\rho}$ as the track representative of the $B_k$-descriptor of $\rho$, and the unravelling algorithm returns $\overline{\rho}'$ as the representative of the $B_k$-descriptor of $\rho'$, since by Lemma 4.1 $\rho \cdot \rho'$ 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 $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists $\rho$ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ and $\mathtt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$. 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 $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$.

□

## A.3 Proofs of Chapter 5

### A.3.1 Proof of Theorem 5.4

We preliminary introduce some definitions.

Given $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an HS formula $\psi$, we define $\mathcal{K}_{|p\ell(\psi)}$ as the Kripke structure $(\overline{\mathcal{AP}}, W, \delta, \overline{\mu}, w_0)$ where $p\ell(\psi)$ is the set of proposition letters occurring in $\psi$, $\overline{\mathcal{AP}} = \mathcal{AP} \cap p\ell(\psi)$ and for all $w \in W$, $\overline{\mu}(w) = \mu(w) \cap p\ell(\psi)$. Intuitively this is the Kripke structure obtained from $\mathcal{K}$ by restricting the labelling of every state to $p\ell(\psi)$.

Given $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $v \in W$, $reach(\mathcal{K}, v)$ is defined as the Kripke structure $(\mathcal{AP}, W', \delta', \mu', v)$, such that

$$W' = \{w \in W \mid \text{there exists } \rho \in \mathrm{Trk}_{\mathcal{K}} \text{ with } \mathrm{fst}(\rho) = v \text{ and } \mathrm{lst}(\rho) = w\},$$

$\delta' = \delta \cap (W' \times W')$ and for all $w \in W'$, $\mu'(w) = \mu(w)$; $reach(\mathcal{K}, v)$ is thus the subgraph of $\mathcal{K}$ of the states reachable from $v$. Notice that the initial state of $reach(\mathcal{K}, v)$ is $v$.

We say that $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w_0')$ are isomorphic (we denote this by $\mathcal{K} \sim \mathcal{K}'$) if and only if there exists a *bijection* $f : W \mapsto W'$ such that:

- $f(w_0) = w_0'$;
- for all $u, v \in W$, $(u, v) \in \delta \iff (f(u), f(v)) \in \delta'$;
- for all $v \in W$, $\mu(v) = \mu'(f(v))$.

Finally we introduce the following shorthand: $\mathcal{L}(\mathcal{K}, \rho)$ is equal to $\sigma(\rho)$, where $\mathcal{K}$ is the Kripke structure $(\mathcal{AP}, W, \delta, \mu, w_0)$, $\rho \in \mathrm{Trk}_{\mathcal{K}}$ and $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$ is the abstract interval model induced by $\mathcal{K}$.

The following lemma holds:

**Lemma A.1.** *Given an $HS[A, \overline{B}]$ formula $\psi$, two Kripke structures $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w_0')$, and two tracks $\rho \in \mathrm{Trk}_{\mathcal{K}}$, $\rho' \in \mathrm{Trk}_{\mathcal{K}'}$ such that*

$$\mathcal{L}(\mathcal{K}_{\restriction p\ell(\psi)}, \rho) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\psi)}, \rho') \text{ and } reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho')),$$

*then $\mathcal{K}, \rho \models \psi \iff \mathcal{K}', \rho' \models \psi$.*

This lemma intuitively states that for any $HS[A, \overline{B}]$ formula $\psi$, if the same set of propositions, restricted to $p\ell(\psi)$, holds over two tracks $\rho$ and $\rho'$ of two Kripke structures, and the subgraphs of the reachable states from $\mathrm{lst}(\rho)$ and $\mathrm{lst}(\rho')$ of the structures are isomorphic, then $\rho$ and $\rho'$ are equivalent with respect to $\psi$.

*Proof.* By induction on the structural complexity of $\psi$.

- $\psi = p$ is a proposition letter. $p\ell(p) = \{p\}$. If $\mathcal{K}, \rho \models p$ then $p \in \mathcal{L}(\mathcal{K}, \rho)$ and hence $p \in \mathcal{L}(\mathcal{K}_{\restriction p\ell(\psi)}, \rho)$. By hypothesis it follows that $p \in \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\psi)}, \rho')$, so $p \in \mathcal{L}(\mathcal{K}', \rho')$ and $\mathcal{K}', \rho' \models p$.
- $\psi = \neg\phi$. $p\ell(\phi) = p\ell(\psi)$. If $\mathcal{K}, \rho \models \neg\phi$ then $\mathcal{K}, \rho \not\models \phi$; by inductive hypothesis, $\mathcal{K}', \rho' \not\models \phi$, so $\mathcal{K}', \rho' \models \neg\phi$.
- $\psi = \phi_1 \wedge \phi_2$. If $\mathcal{K}, \rho \models \phi_1 \wedge \phi_2$ then $\mathcal{K}, \rho \models \phi_1$; by hypothesis we have that $\mathcal{L}(\mathcal{K}_{\restriction p\ell(\psi)}, \rho) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\psi)}, \rho')$ and $reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho'))$. Being $p\ell(\phi_1) \subseteq p\ell(\psi)$, we get that $\mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi_1)}, \rho) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi_1)}, \rho')$ and

  $$reach(\mathcal{K}_{\restriction p\ell(\phi_1)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{\restriction p\ell(\phi_1)}, \mathrm{lst}(\rho')).$$

  By inductive hypothesis, $\mathcal{K}', \rho' \models \phi_1$. By reasoning in a symmetric way for $\phi_2$, the thesis follows.
- $\psi = \langle \mathrm{A} \rangle \phi$. If $\mathcal{K}, \rho \models \langle \mathrm{A} \rangle \phi$ then there exists $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$ and $\mathcal{K}, \overline{\rho} \models \phi$. Obviously $p\ell(\psi) = p\ell(\phi)$. Since by hypothesis it holds that $reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho'))$, we consider $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}'}$ such that $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, $|\overline{\rho}| = |\overline{\rho}'|$ and for all $0 \leq i < |\overline{\rho}|$, $f(\overline{\rho}(i)) = \overline{\rho}'(i)$, where $f$ is the (an) isomorphism between $reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho))$ and $reach(\mathcal{K}'_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho'))$. It immediately follows that $\mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi)}, \overline{\rho}')$.

  We have to prove that $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho})) \sim reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$ as well. Indeed, the restriction of $f$ to the states of $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, say $f'$, is

an isomorphism between $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$ and $reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$ (notice that $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$ is a subgraph of $reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho))$): first, $f(\mathrm{lst}(\overline{\rho})) = f'(\mathrm{lst}(\overline{\rho})) = \mathrm{lst}(\overline{\rho}')$; then if $w$ is any state of $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, $f(w) = f'(w) = w'$ must be a state of $reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$: since there exists a track from $\mathrm{lst}(\overline{\rho})$ to $w$, then there exists an isomorphic track (w.r.t. $f$) from $\mathrm{lst}(\overline{\rho}')$ to $w'$. Moreover, if $(w, \overline{w}) \in \delta$, then $\overline{w}$ is another state of $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, hence $(w', f(\overline{w})) \in \delta'$ and $f(\overline{w}) = f'(\overline{w})$ is a state of $reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$, as well. Reasoning by symmetry, we can conclude that for any two states of $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, $v, v'$, it holds that $(v, v')$ is an edge iff $(f'(v), f'(v'))$ is an edge of $reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$.

By inductive hypothesis, $\mathcal{K}', \overline{\rho}' \models \phi$. So $\mathcal{K}', \rho' \models \langle \mathrm{A} \rangle \phi$.

- $\psi = \langle \overline{\mathrm{B}} \rangle \phi$. If $\mathcal{K}, \rho \models \langle \overline{\mathrm{B}} \rangle \phi$, then $\mathcal{K}, \rho \cdot \overline{\rho} \models \phi$, where $\rho \cdot \overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ and $\overline{\rho}$ is either a single state or a proper track. Obviously $p\ell(\psi) = p\ell(\phi)$. Analogously to the previous case, we consider $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}'}$ such that $|\overline{\rho}| = |\overline{\rho}'|$ and for all $0 \leq i < |\overline{\rho}|$, $f(\overline{\rho}(i)) = \overline{\rho}'(i)$, where $f$ is the isomorphism between $reach(\mathcal{K}_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho))$ and $reach(\mathcal{K}'_{\restriction p\ell(\psi)}, \mathrm{lst}(\rho'))$. Since $f(\mathrm{lst}(\rho)) = \mathrm{lst}(\rho')$, by definition of isomorphism $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$ implies $(\mathrm{lst}(\rho'), \mathrm{fst}(\overline{\rho}')) \in \delta'$. It immediately follows that

  $$\mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi)}, \overline{\rho}') \text{ and } reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho})) \sim reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\overline{\rho}')).$$

  Finally,

  $$\mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi)}, \rho \cdot \overline{\rho}) = \mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi)}, \rho) \cap \mathcal{L}(\mathcal{K}_{\restriction p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi)}, \rho') \cap \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi)}, \overline{\rho}') = \mathcal{L}(\mathcal{K}'_{\restriction p\ell(\phi)}, \rho' \cdot \overline{\rho}').$$

  and obviously $reach(\mathcal{K}_{\restriction p\ell(\phi)}, \mathrm{lst}(\rho \cdot \overline{\rho})) \sim reach(\mathcal{K}'_{\restriction p\ell(\phi)}, \mathrm{lst}(\rho' \cdot \overline{\rho}'))$. By inductive hypothesis, $\mathcal{K}', \rho' \cdot \overline{\rho}' \models \phi$, therefore $\mathcal{K}', \rho' \models \langle \overline{\mathrm{B}} \rangle \phi$.

□

We are now ready to give the proof of Theorem 5.4.

*Proof.* We prove that

$$\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)$$

is a true quantified boolean formula if and only if $\mathcal{K}_{QBF}^{x_n, \cdots, x_1} \models \xi$ by induction on the number of variables $n$ of $\psi$. In the following, $\phi(x_n, x_{n-1}, \cdots x_1)\{x_i/v\}$, with $v \in \{\top, \bot\}$, denotes the formula obtained from $\phi(x_n, x_{n-1}, \cdots x_1)$ by replacing all of the occurrences of $x_i$ with $v$. For the purpose of the proof, it is worth noticing that $\mathcal{K}_{QBF}^{x_n, x_{n-1}, \cdots, x_1}$ and $\mathcal{K}_{QBF}^{x_{n-1}, \cdots, x_1}$ are isomorphic when restricted to the states $w_{x_{n-1}}^{\top 1}$, $w_{x_{n-1}}^{\top 2}$, $w_{x_{n-1}}^{\bot 1}$, $w_{x_{n-1}}^{\bot 2}$, $\cdots$, $w_{x_1}^{\top 1}$, $w_{x_1}^{\top 2}$, $w_{x_1}^{\bot 1}$, $w_{x_1}^{\bot 2}$, $sink$ (i.e. the leftmost part of both Kripke structures is eliminated), and the labeling of states is suitably restricted as well. Moreover only the track $w_0 w_1$ satisfies $start$ and, for $i = n, \cdots, 1$, the letter $x_{i\,aux}$ holds only over the two tracks $w_{x_i}^{\top 1} w_{x_i}^{\top 2}$ and $w_{x_i}^{\bot 1} w_{x_i}^{\bot 2}$.

(Case $n = 0$) $\psi$ equals $\phi$ and it does not have variables; the states of $\mathcal{K}_{QBF}^{\emptyset}$ are $W = \{w_0, w_1, sink\}$ and $\xi = start \rightarrow \phi$.

Let us assume $\phi$ is true. All initial tracks of length greater than 2 trivially satisfy $\xi$, as $start$ does not hold on them. As for $w_0 w_1$, it is true that $\mathcal{K}_{QBF}^{\emptyset}, w_0 w_1 \models \phi$,

since $\phi$ is true (its truth does not depend on the propositions which hold on $w_0 w_1$, because it has no variables). Thus $\mathcal{K}^{\emptyset}_{QBF} \models \xi$. Vice versa, if $\mathcal{K}^{\emptyset}_{QBF} \models \xi$ then in particular $\mathcal{K}^{\emptyset}_{QBF}, w_0 w_1 \models \phi$. But $\phi$ does not have variables, hence it is true.

(Case $n \geq 1$) Let us consider the QBF formula

$$\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1);$$

we distinguish two cases, $Q_n = \exists$ and $Q_n = \forall$ and for both we prove the two implications.

- $Q_n = \exists$:

($\Rightarrow$) if $\psi$ is true, then by definition, if all occurrences of $x_n$ in $\phi(x_n, x_{n-1}, \cdots x_1)$ are replaced with some $v \in \{\top, \bot\}$, we get $\phi'(x_{n-1}, \cdots x_1) = \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/v\}$ such that

$$\psi' = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi'(x_{n-1}, \cdots x_1)$$

is a true quantified boolean formula. By inductive hypothesis $\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF} \models \xi'$, where $\xi' = start \rightarrow \xi'_{n-1}$ is derived from $\psi'$ and $\xi'_{n-1} = \xi_{n-1}\{x_n/v\}$. It follows that $\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF}, w'_0 w'_1 \models \xi'_{n-1}$, where $w'_0$ and $w'_1$ are the two "leftmost" states of $\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF}$ (corresponding to $w_0$ and $w_1$ of $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}$).

Let us prove that $\mathcal{K}^{x_n,\cdots,x_1}_{QBF} \models \xi$. We consider a generic initial track $\rho$ in $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}$: if it does not satisfy $start$ then trivially $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, \rho \models \xi$, otherwise $\rho = w_0 w_1$ and we have to show that $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 \models \langle \overline{\mathrm{B}} \rangle ((\langle \mathrm{A} \rangle x_{n\,aux}) \wedge \xi_{n-1})$ $(= \xi_n)$. We consider $w_0 w_1 w^{\top 1}_{x_n}$ if $v = \top$ (and $w_0 w_1 w^{\bot 1}_{x_n}$ otherwise). In the former case (the latter is symmetric) we have to prove that $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} \models (\langle \mathrm{A} \rangle x_{n\,aux}) \wedge \xi_{n-1}$. Trivially $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} \models \langle \mathrm{A} \rangle x_{n\,aux}$, thus it remains to show that $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} \models \xi_{n-1}$.

As we proved, from the inductive hypothesis it follows that $\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF}, w'_0 w'_1 \models \xi'_{n-1} (= \xi_{n-1}\{x_n/\top\})$. Now, since

- $p\ell(\xi_{n-1}\{x_n/\top\}) = \{x_1, \cdots, x_{n-1}, x_{1\,aux}, \cdots, x_{n-1\,aux}\}$,
- $\mathcal{L}(\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w'_0 w'_1) = \{x_{n-1}, \cdots, x_1\}$,
- $\mathcal{L}(\mathcal{K}^{x_n,\cdots,x_1}_{QBF}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n}) = \{x_{n-1}, \cdots, x_1\}$,
- $reach(\mathcal{K}^{x_n,\cdots,x_1}_{QBF}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w^{\top 2}_{x_n}) \sim reach(\mathcal{K}^{x_{n-1},\cdots,x_1}_{QBF}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w'_1)$,

by Lemma A.1, $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} \models \xi'_{n-1}$. So $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} \models \xi_{n-1}$ as $x_n$ is in the labelling of the track $w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n}$ and of any $\overline{\rho}$ such that $w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} \in \mathrm{Pref}(\overline{\rho})$.

Now, if $n = 1$, $\xi_{n-1} = \phi(x_n)$, and we have $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} \models \xi_{n-1}$.

If $n > 1$, either $\xi_{n-1} = \langle \overline{\mathrm{B}} \rangle ((\langle \mathrm{A} \rangle x_{n-1\,aux}) \wedge \xi_{n-2})$ or $\xi_{n-1} = [\overline{B}]((\langle \mathrm{A} \rangle x_{n-1\,aux}) \rightarrow \xi_{n-2})$: in the first case, since $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} \models \langle \overline{\mathrm{B}} \rangle ((\langle \mathrm{A} \rangle x_{n-1\,aux}) \wedge \xi_{n-2})$ then there are only two possibilities: $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} w^{\top 1}_{x_{n-1}} \models (\langle \mathrm{A} \rangle x_{n-1\,aux}) \wedge \xi_{n-2}$ or $\mathcal{K}^{x_n,\cdots,x_1}_{QBF}, w_0 w_1 w^{\top 1}_{x_n} w^{\top 2}_{x_n} w^{\bot 1}_{x_{n-1}} \models (\langle \mathrm{A} \rangle x_{n-1\,aux}) \wedge \xi_{n-2}$. In

both cases $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models \langle\overline{\mathrm{B}}\rangle((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2})$ by definition. Otherwise $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} w_{x_n}^{\top 2} \models [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \rightarrow \xi_{n-2})$. It follows that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} w_{x_n}^{\top 2} w_{x_{n-1}}^{\top 1} \models \xi_{n-2}$ *and* $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} w_{x_n}^{\top 2} w_{x_{n-1}}^{\perp 1} \models \xi_{n-2}$. As a consequence $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models [\overline{B}]((\neg \langle \mathrm{A}\rangle\, x_{n-1\,aux}) \vee \xi_{n-2})(= \xi_{n-1})$ (recall that the only successor of $w_{x_n}^{\top 1}$ in $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}$ is $w_{x_n}^{\top 2}$ and in particular $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} w_{x_n}^{\top 2} \models \neg \langle \mathrm{A}\rangle\, x_{n-1\,aux}$).

($\Leftarrow$) If $\mathcal{K}_{QBF}^{x_n,\cdots,x_1} \models \xi$, then $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 \models \langle\overline{\mathrm{B}}\rangle(\langle \mathrm{A}\rangle\, x_{n\,aux} \wedge \xi_{n-1})$. A possibility is that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models (\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1}$ (the other, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\perp 1} \models (\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1}$, is symmetric). Hence $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models \xi_{n-1}\{x_n/\top\}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} w_{x_n}^{\top 2} \models \xi_{n-1}\{x_n/\top\}$ (as before). Thus (by Lemma A.1) $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}, w_0' w_1' \models \xi_{n-1}\{x_n/\top\}(= \xi_{n-1}')$ and $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1} \models start \rightarrow \xi_{n-1}'$, i.e. $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1} \models \xi'$. By inductive hypothesis

$$\psi' = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/\top\}$$

is a true quantified boolean formula. Thus

$$\psi = \exists x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)$$

is a true quantified boolean formula.

- $Q_n = \forall$:

($\Rightarrow$) the formulas

$$\psi' = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/\top\}$$

and

$$\psi'' = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/\bot\}$$

are both true quantified boolean formulas. We now show that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 \models [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n\,aux}) \rightarrow \xi_{n-1})$. We just need to prove that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\perp 1} \models \xi_{n-1}$. By reasoning as in the $\exists$ case, the thesis follows.

($\Leftarrow$) If $\mathcal{K}_{QBF}^{x_n,\cdots,x_1} \models \xi$, then $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 \models [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n\,aux}) \rightarrow \xi_{n-1})$. Therefore $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\top 1} \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0 w_1 w_{x_n}^{\perp 1} \models \xi_{n-1}$. Reasoning as in the $\exists$ case and by applying the inductive hypothesis twice, we get that

$$Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/\top\}$$

and

$$Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)\{x_n/\bot\}$$

are true quantified boolean formulas, thus

$$\forall x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots x_1)$$

is a true quantified boolean formula.

□

# Bibliography


[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[BCC+99] A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu. Symbolic model checking using sat procedures instead of bdds. In *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, pages 317–320. ACM, 1999.

[BCC+03] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu. Bounded model checking. *Advances in computers*, 58:117–148, 2003.

[BCM+90] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 1020 states and beyond. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on e*, pages 428–439, Jun 1990.

[BDG+14] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. The dark side of interval temporal logic: marking the undecidability border. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):41–83, 2014.

[BGMS09] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic*, 161(3):289–304, 2009.

[BGMS10] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation*, 20(1):133–166, 2010.

[BMSS11a] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. Optimal tableau systems for propositional neighborhood logic over all, dense, and discrete linear orders. In *Proc. of TABLEAUX*, pages 73–87, 2011.

[BMSS11b] 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*, pages 387–396. IEEE Comp. Society Press, 2011.

[BT03] H. Bowman and S. J. Thompson. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation*, 13(2):195–239, 2003.

[CE81] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1981.

[CGP02] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2002.

[CH04] Z. Chaochen and M. R. Hansen. *Duration Calculus - A Formal Approach to Real-Time Systems*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[CM90] O. Coudert and J.C. Madre. A unified framework for the formal verification of sequential circuits. In *Computer-Aided Design, 1990. ICCAD-90. Digest of Technical Papers., 1990 IEEE International Conference on*, pages 126–129, Nov 1990.

[DGMS11] D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Interval temporal logics: a journey. *Bull. of the EATCS*, 105:73–99, 2011.

[Gab87] D.M. Gabbay. The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems. In *TLS'87*, LNCS 398, pages 409–448. Springer, 1987.

[GMS04] V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

[GT99] F. Giunchiglia and P. Traverso. Planning as model checking. In *Proc. of the 5th ECP*, pages 1–20, 1999.

[HS91] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[Lan06] M. Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2006.

[LM13] A. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *Proc. of IJCAI*, 2013.

[LM14] A. R. Lomuscio and J. Michaliszyn. Decidability of model checking multi-agent systems against a class of EHS specifications. In *Proc. of ECAI*, pages 543–548, 2014.

[Lod00] K. Lodaya. Sharpening the undecidability of interval temporal logic. In *Proc. of ASIAN*, LNCS 1961, pages 290–298, 2000.

[LR06] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *Proc. of the 12th TACAS*, pages 450–454, 2006.

[McM93] K. L. McMillan. Symbolic model checking. In *Symbolic Model Checking*, pages 25–60. Springer US, 1993.

[MM14] J. Marcinkowski and J. Michaliszyn. The undecidability of the logic of subintervals. *Fundamenta Informaticae*, 131(2):217–240, 2014.

[MMPP14] A. Montanari, A. Murano, G. Perelli, and A Peron. Checking interval properties of computations. In *Proc. of TIME*, pages 59–68, 2014.

[MMS12] D. Della Monica, A. Montanari, and P. Sala. The importance of the past in interval temporal logics: The case of propositional neighborhood logic. In *Logic Programs, Norms and Action*, pages 79–102, 2012.

[Mos83] B. Moszkowski. *Reasoning About Digital Circuits*. PhD thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.

[MPS10] A. Montanari, G. Puppis, and P. Sala. Maximal decidable fragments of Halpern and Shoham's modal logic of intervals. In *ICALP'10 (2)*, LNCS 6199, pages 345–356, 2010.

[MPS14] A. Montanari, G. Puppis, and P. Sala. Decidability of the interval temporal logic $\mathsf{A}, \overline{\mathsf{A}}, \mathsf{B}, \overline{\mathsf{B}}$ over the rationals. In *Prof. of MFCS*, pages 451–463, 2014.

[MS12] A. Montanari and P. Sala. An optimal tableau system for the logic of temporal neighborhood over the reals. In *Proc. of TIME*, pages 39–46, 2012.

[Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[Pel93] D. Peled. All from one, one for all: on model checking using representatives. In Costas Courcoubetis, editor, *Computer Aided Verification*, volume 697 of *Lecture Notes in Computer Science*, pages 409–423. Springer Berlin Heidelberg, 1993.

[Pnu77] A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.

[Pnu81] A. Pnueli. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science*, 13:45–60, 1981.

[Pra05] I. Pratt-Hartmann. Temporal prepositions and their logic. *Artificial Intelligence*, 166(1-2):1–36, 2005.

[QS81] J.P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in CESAR. In *SP'81*, LNCS 137, pages 337–351. Springer, 1981.

[Roe80] P. Roeper. Intervals and tenses. *Journal of Philosophical Logic*, 9:451–469, 1980.

[Sip12] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 3rd edition, 2012.

[Ven90] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.

[Ven91] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

[VW86] M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86*, pages 332–344. IEEE Comp. Society Press, 1986.