Interval vs. Point Temporal Logic Model Checking: An Expressiveness Comparison

LAURA BOZZELLI, University of Napoli "Federico II", Italy
ALBERTO MOLINARI and ANGELO MONTANARI, University of Udine, Italy
ADRIANO PERON, University of Napoli "Federico II", Italy
PIETRO SALA, University of Verona, Italy

In recent years, model checking with interval temporal logics is emerging as a viable alternative to model checking with standard point-based temporal logics, such as LTL, CTL, CTL*, and the like. The behavior of the system is modeled by means of (finite) Kripke structures, as usual. However, while temporal logics which are interpreted "point-wise" describe how the system evolves state-by-state, and predicate properties of system states, those which are interpreted "interval-wise" express properties of computation stretches, spanning a sequence of states. A proposition letter is assumed to hold over a computation stretch (interval) if and only if it holds over each component state (homogeneity assumption). A natural question arises: is there any advantage in replacing points by intervals as the primary temporal entities, or is it just a matter of taste?

In this article, we study the expressiveness of Halpern and Shoham's interval temporal logic (HS) in model checking, in comparison with those of LTL, CTL, and CTL*. To this end, we consider three semantic variants of HS: the state-based one, introduced by Montanari et al. in [30, 34], that allows time to branch both in the past and in the future, the computation-tree-based one, that allows time to branch in the future only, and the trace-based variant, that disallows time to branch. These variants are compared among themselves and to the aforementioned standard logics, getting a complete picture. In particular, we show that HS with trace-based semantics is equivalent to LTL (but at least exponentially more succinct), HS with computation-tree-based semantics is equivalent to finitary CTL*, and HS with state-based semantics is incomparable with all of them (LTL, CTL, and CTL*).

CCS Concepts: • **Theory of computation → Logic and verification**; **Modal and temporal logics**; **Verification by model checking**;

Additional Key Words and Phrases: Interval temporal logics, expressiveness, model checking

ACM Reference format:
Laura Bozzelli, Alberto Molinari, Angelo Montanari, Adriano Peron, and Pietro Sala. 2018. Interval vs. Point Temporal Logic Model Checking: An Expressiveness Comparison. *ACM Trans. Comput. Logic* 20, 1, Article 4 (December 2018), 31 pages.
https://doi.org/10.1145/3281028

This work is an extended and revised version of [8].

The work has been supported by the GNCS project *Formal Methods for Verification and Synthesis of Discrete and Hybrid Systems*. The work by A. Molinari and A. Montanari has also been supported by the project *(PRID) ENCASE - Efforts in the uNderstanding of Complex interActing SystEms*.

Authors' addresses: L. Bozzelli and A. Peron, University of Napoli "Federico II", via Claudio 21, IT-80125 Napoli; emails: lr.bozzelli@gmail.com, adrperon@unina.it; A. Molinari and A. Montanari, University of Udine, via delle Scienze 206, IT-33100 Udine; emails: molinari.alberto@gmail.com, angelo.montanari@uniud.it; P. Sala, University of Verona, strada le Grazie 15, IT-37134 Verona; email: pietro.sala@univr.it.

1529-3785/2018/12-ART4 $15.00
https://doi.org/10.1145/3281028

1 INTRODUCTION

Point-based temporal logics (PTLs) provide a standard framework for the specification of the behavior of reactive systems that makes it possible to describe how a system evolves state-by-state ("point-wise" view). PTLs have been successfully employed in *model checking* (MC), which enables one to automatically verify complex finite-state systems usually modeled as finite propositional Kripke structures. The MC methodology considers two types of PTLs—*linear* and *branching*—which differ in the underlying model of time. In linear PTLs, like LTL [37], each moment in time has a unique possible future: formulas are interpreted over paths of a Kripke structure, and thus they refer to a single computation of the system. In branching PTLs, like CTL and CTL* [17], each moment in time may evolve into several possible futures: formulas are interpreted over states of the Kripke structure, hence referring to all the possible system computations.

Interval temporal logics (ITLs) have been proposed as an alternative setting for reasoning about time [20, 36, 42]. Unlike standard PTLs, they assume intervals, instead of points, as their primitive entities. ITLs allow one to specify relevant temporal properties that involve, e.g., actions with duration, accomplishments, and temporal aggregations, which are inherently "interval-based," and thus cannot be naturally expressed by PTLs. ITLs have been applied in various areas of computer science, including formal verification, computational linguistics, planning, and multi-agent systems [26, 36, 38]. *Halpern and Shoham's modal logic of time intervals* (referred to as HS) [20] is the most popular among the ITLs. It features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen's relations [1]), apart from equality. Its *satisfiability problem* turns out to be highly undecidable for all interesting (classes of) linear orders [20]; the same happens with most of its fragments [11, 25, 29], but there are some noteworthy exceptions like the logic of temporal neighborhood $\overline{\mathsf{AA}}$, over all relevant (classes of) linear orders [13, 14], and the logic of sub-intervals D, over the class of dense linear orders [12, 35].

In this article, we focus on the *MC problem* for HS. In order to check interval properties of computations, one needs to collect information about states into computation stretches, that is, finite paths of the Kripke structure (*traces* for short). Each trace is interpreted as an interval, whose labeling is defined on the basis of the labeling of the component states. Such an approach to HS MC has been simultaneously and independently proposed by Montanari et al. in [30, 34] and by Lomuscio and Michaliszyn in [26, 27].

In [30, 34], Montanari et al. assume a *state-based* semantics, according to which intervals/traces are "forgetful" of the history leading to their initial state. Since the initial (final, respectively) state of an interval may feature several predecessors (successors, respectively), such an interpretation induces a branching reference both in the future and in the past. A graphical account of the state-based semantics can be found in Figure 1; a detailed explanation will be given in the following. The other fundamental choice done in [30, 34] concerns the labeling of intervals: a natural principle, known as the *homogeneity assumption*, is adopted, which states that a proposition letter holds over an interval if and only if it holds over each component state (such an assumption turns out to be the most appropriate choice for many practical applications). In this setting, the MC problem for full HS turns out to be decidable. More precisely, it is **EXPSPACE**-hard [7], while the only known upper bound is non-elementary [30].[1] The exact complexity of MC for almost all the meaningful syntactic fragments of HS, which ranges from **co-NP** to $\mathbf{P^{NP}}$, **PSPACE**, and beyond, has been determined in a subsequent series of papers [7, 9, 10, 30–33].

In [26, 27], Lomuscio and Michaliszyn address the MC problem for some fragments of HS extended with epistemic modalities. Their semantic assumptions are different from those made in [30,

[1]Here and in the following we refer to the *combined complexity* of MC (which accounts for both the size of the Kripke structure and of the formula at the same time).

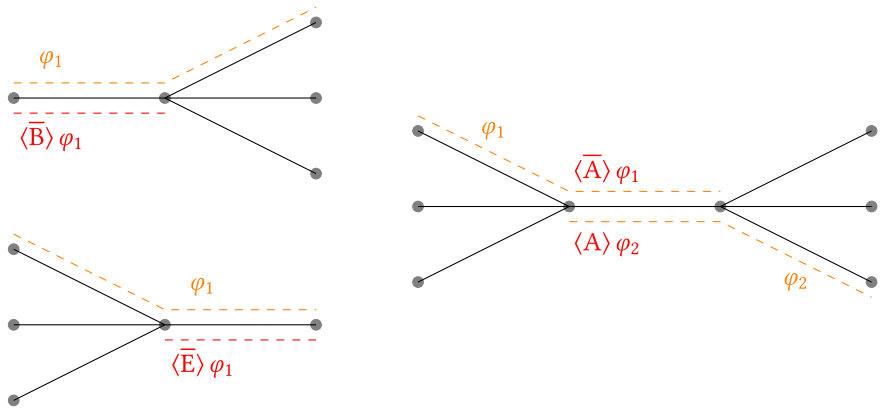


Fig. 1. State-based semantic variant $\mathrm{HS}_{\mathrm{st}}$: past and future are branching.

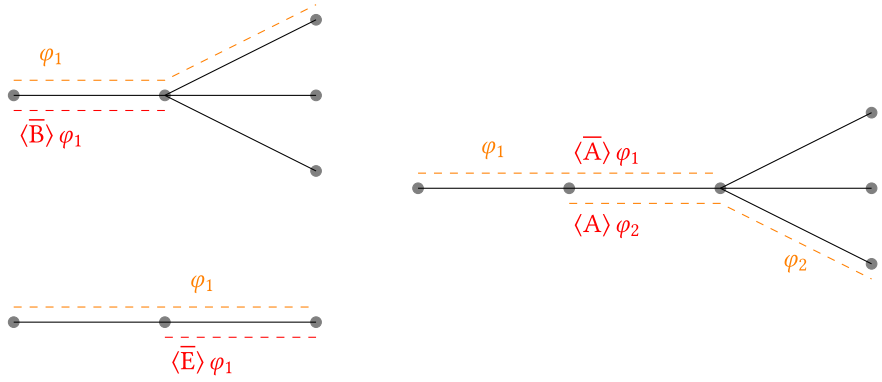


Fig. 2. Computation-tree-based semantic variant $\mathrm{HS}_{\mathrm{ct}}$: future is branching, past is linear, finite, and cumulative.

34]: the fragments are interpreted over the unwinding of the Kripke structure (*computation-tree-based* semantics—see Figure 2 for a graphical account), and the interval labeling takes into account only the endpoints of intervals. In [26], they focus on the HS fragment BE of Allen's relations *started-by* and *finished-by*, extended with epistemic modalities. They consider a *restricted form of MC (local MC)*, which checks the specification against a single (finite) initial computation interval, and prove that it is **PSPACE**-complete. In [27], they demonstrate that the picture drastically changes with other fragments of HS that allow one to access infinitely many intervals. In particular, they prove that the MC problem for the HS fragment $\mathrm{A}\overline{\mathrm{B}}$ of Allen's relations *meets* and *starts*, extended with epistemic modalities, is decidable with a non-elementary upper bound. The decidability status of MC for full epistemic HS is not known.

To summarize, the MC problem for HS (and its fragments) has been extensively studied under the state-based and the computation-tree-based semantics, mainly focusing on complexity issues. What is missing is a formal comparison of the expressiveness of HS MC and MC for standard point-based temporal logics. A comparison of the expressiveness of the MC problem for HS under the state-based and the computation-tree-based semantics is missing as well.

Our Contribution. In this article, we study the expressiveness of HS, in the context of MC, in comparison with that of the standard PTLs LTL, CTL, and CTL*. The analysis is carried on enforcing the homogeneity assumption.

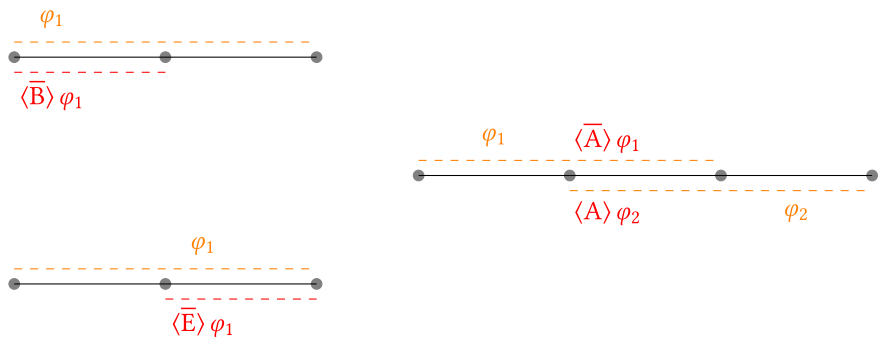


Fig. 3. Trace-based semantic variant HS_{lin}: neither past nor future are branching.

We prove that HS endowed with the state-based semantics proposed in [30, 34] (hereafter denoted as HS_{st}) is not comparable with LTL, CTL, and CTL*. On the one hand, the result supports the intuition that HS_{st} gains some expressiveness by the ability of branching in the past. On the other hand, HS_{st} does not feature the possibility of forcing the verification of a property over an infinite path, thus implying that the formalisms are not comparable. With the aim of having a more "effective" comparison base, we consider two other semantic variants of HS, namely, the *computation-tree-based semantic variant* (denoted as HS_{ct}) and the *trace-based* one (HS_{lin}).

The state-based (see Figure 1) and computation-tree-based (see Figure 2) approaches rely on a branching-time setting and differ in the nature of past. In the latter approach, past is linear: each interval may have several possible futures, but only a unique past. Moreover, past is assumed to be finite and cumulative, that is, the story of the current situation increases with time, and is never forgotten. The trace-based approach relies on a linear-time setting (see Figure 3), where the infinite paths (computations) of the given Kripke structure are the main semantic entities. Branching is neither allowed in the past nor in the future. Note that the linear-past (rather than branching) approach is more suited to the specification of dynamic behaviors, because it considers states in a computation tree, while the branching-past approach considers machine states, where past is not very meaningful for the specification of behavioral constraints [23].

The variant HS_{ct} is a natural candidate for an expressiveness comparison with the branching time logics CTL and CTL*. The most interesting and technically involved result is the characterization of the expressive power of HS_{ct}: HS_{ct} turns out to be expressively equivalent to finitary CTL*, that is, the variant of CTL* with quantification over finite paths. As for CTL, a non-comparability result can be stated.

The variant HS_{lin} is a natural candidate for an expressiveness comparison with LTL. We prove that HS_{lin} and LTL are equivalent (this result holds true even for a very small fragment of HS_{lin}), but the former is at least exponentially more succinct than the latter.

We complete the picture with a comparison of the three semantic variants HS_{st}, HS_{ct}, and HS_{lin}. We prove that, as expected, HS_{lin} is not comparable with either of the branching versions, HS_{ct} and HS_{st}. The interesting result is that, on the other hand, HS_{ct} is strictly included in HS_{st}: this supports HS_{st}, adopted in [7, 9, 30–33], as a reasonable and adequate semantic choice. The complete picture of the expressiveness results is reported in Figure 4 (the symbols $\neq$, $\equiv$, and $<$ denote incomparability, equivalence, and strict inclusion, respectively).

Structure of the Article. In Section 2, we introduce basic notation and preliminary notions. In Section 2.1 we define Kripke structures and interval structures, in Section 2.2 we recall the well-known PTLs LTL, CTL, and CTL*, and in Section 2.3 we present the interval temporal logic HS. Then, in Section 2.4 we define the three semantic variants of HS (HS_{st}, HS_{ct}, and HS_{lin}). Finally,

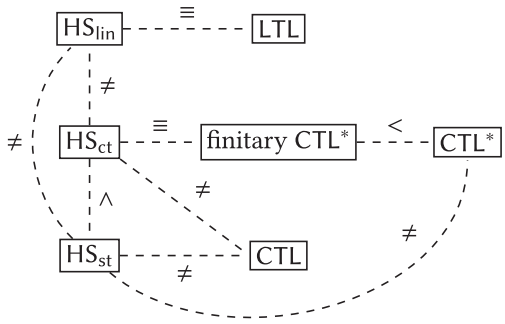


Fig. 4. Overview of the expressiveness results.

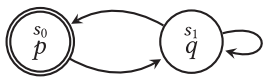


Fig. 5. The Kripke structure $\mathcal{K}$.

in Section 2.5 we provide a detailed example which gives an intuitive account of the three semantic variants and highlights their differences. In the next three sections, we analyze and compare their expressiveness. In Section 3, we show the expressive equivalence of LTL and $\mathsf{HS}_{\mathsf{lin}}$. Then, in Section 4 we prove the expressive equivalence of $\mathsf{HS}_{\mathsf{ct}}$ and finitary CTL*. Finally, in Section 5, we compare the expressiveness of $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{ct}}$, and $\mathsf{HS}_{\mathsf{lin}}$. Conclusions summarize the work done and outline some directions for future research.

2 PRELIMINARIES

In this section, we introduce the notation and some fundamental notions that will be extensively used in the rest of the article. Let $(\mathbb{N}, <)$ be the set of natural numbers equipped with the standard linear ordering. For all $i, j \in \mathbb{N}$, with $i \leq j$, we denote by $[i, j]$ the set of natural numbers h such that $i \leq h \leq j$. Let Σ be an alphabet and w be a non-empty finite or infinite word over Σ. We denote by $|w|$ the length of w ($|w| = \infty$ if w is infinite). For all $i, j \in \mathbb{N}$, with $i \leq j$, $w(i)$ denotes the i-th letter of w, while $w[i, j]$ denotes the finite subword of w given by $w(i) \cdots w(j)$. If w is finite and $|w| = n + 1$, we define $\mathrm{fst}(w) = w(0)$ and $\mathrm{lst}(w) = w(n)$. The sets of all proper prefixes and suffixes of w are $\mathrm{Pref}(w) = \{w[0, i] \mid 0 \leq i \leq n - 1\}$ and $\mathrm{Suff}(w) = \{w[i, n] \mid 1 \leq i \leq n\}$, respectively. The set of all the finite words over Σ is denoted by Σ^*, and $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$, where ε is the empty word.

2.1 Kripke Structures and Interval Structures

Systems are usually modeled as Kripke structures. Let $\mathcal{AP}$ be a finite set of proposition letters, which represent predicates decorating the states of the given system.

Definition 2.1 (Kripke Structure). A *Kripke structure* over a finite set $\mathcal{AP}$ of proposition letters is a tuple $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$, where S is a set of states, $\delta \subseteq S \times S$ is a left-total transition relation, $\mu : S \to 2^{\mathcal{AP}}$ is a total labeling function assigning to each state s the set of proposition letters that hold over it, and $s_0 \in S$ is the initial state. For $(s, s') \in \delta$, we say that s' is a successor of s, and s is a predecessor of s'. Finally, we say that $\mathcal{K}$ is finite if S is finite.

For example, Figure 5 depicts the finite Kripke structure $\mathcal{K} = (\{p, q\}, \{s_0, s_1\}, \delta, \mu, s_0)$, where $\delta = \{(s_0, s_1), (s_1, s_0), (s_1, s_1)\}$, $\mu(s_0) = \{p\}$, $\mu(s_1) = \{q\}$. The initial state s_0 is marked by a double circle.

Let $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$ be a Kripke structure. An infinite path π of $\mathcal{K}$ is an infinite word over S such that $(\pi(i), \pi(i + 1)) \in \delta$ for all $i \geq 0$. A *trace* (or finite path) of $\mathcal{K}$ is a non-empty prefix

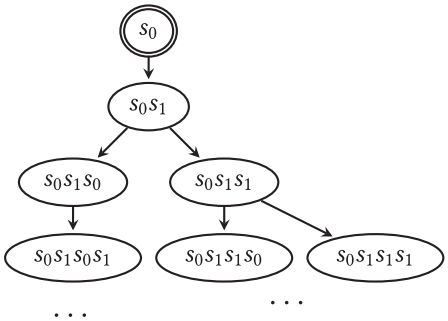


Fig. 6. Computation tree $C(\mathcal{K})$ of the Kripke structure $\mathcal{K}$ of Figure 5.

of some infinite path of $\mathcal{K}$. A finite or infinite path is *initial* if it starts from the initial state of $\mathcal{K}$. Let $\mathrm{Trk}_{\mathcal{K}}$ be the (*infinite*) set of all traces of $\mathcal{K}$ and $\mathrm{Trk}^0_{\mathcal{K}}$ be the set of initial traces of $\mathcal{K}$. For a trace ρ, states(ρ) denotes the set of states occurring in ρ, that is, states$(\rho) = \{\rho(0), \ldots, \rho(n)\}$, where $|\rho| = n + 1$.

We now introduce the notion of *D-tree structure*, namely, an infinite tree-shaped Kripke structure with branches over a set D of directions.

Definition 2.2 (D-tree Structure). Given a set D of directions, a *D-tree structure* (over $\mathcal{AP}$) is a Kripke structure $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$ such that $s_0 \in D$, S is a prefix closed subset of D^+, and δ is the set of pairs $(s, s') \in S \times S$ such that there exists $d \in D$ for which $s' = s \cdot d$ (note that δ is completely specified by S). The states of a D-tree structure are called *nodes*.

A Kripke structure $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$ induces an S-tree structure, called the *computation tree of* $\mathcal{K}$, denoted by $C(\mathcal{K})$, which is obtained by unwinding $\mathcal{K}$ from the initial state (note that the directions are the set of states of $\mathcal{K}$). Formally, $C(\mathcal{K}) = (\mathcal{AP}, \mathrm{Trk}^0_{\mathcal{K}}, \delta', \mu', s_0)$, where the set of nodes is the set of initial traces of $\mathcal{K}$ and for all $\rho, \rho' \in \mathrm{Trk}^0_{\mathcal{K}}$, $\mu'(\rho) = \mu(\mathrm{lst}(\rho))$ and $(\rho, \rho') \in \delta'$ if and only if $\rho' = \rho \cdot s$ for some $s \in S$. See Figure 6 for an example.

Given a strict partial ordering $\mathbb{S} = (X, <)$, an *interval* in $\mathbb{S}$ is an ordered pair $[x, y]$ such that $x, y \in X$ and $x \leq y$. The interval $[x, y]$ denotes the subset of X given by the set of points $z \in X$ such that $x \leq z \leq y$. We denote by $\mathbb{I}(\mathbb{S})$ the set of intervals in $\mathbb{S}$.

Definition 2.3 (Interval Structure). An *interval structure IS* over $\mathcal{AP}$ is a pair $IS = (\mathbb{S}, \sigma)$ such that $\mathbb{S} = (X, <)$ is a strict partial ordering and $\sigma : \mathbb{I}(\mathbb{S}) \to 2^{\mathcal{AP}}$ is a labeling function assigning a set of proposition letters to each interval over $\mathbb{S}$.

2.2 Standard Temporal Logics

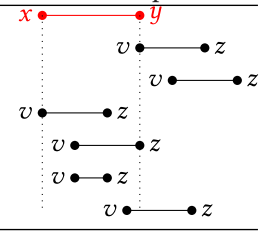
In this subsection, we recall the standard propositional temporal logics CTL*, CTL, and LTL [17, 37]. Given a set of proposition letters $\mathcal{AP}$, the formulas φ of CTL* are defined as follows:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \exists\varphi,$$

where $p \in \mathcal{AP}$, X and U are the "next" and "until" temporal modalities, and $\exists$ is the existential path quantifier.[2] We also use the standard shorthands $\forall\varphi := \neg\exists\neg\varphi$ ("universal path quantifier"), $\mathsf{F}\varphi := \top \mathsf{U} \varphi$ ("eventually" or "in the future"), and its dual $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$ ("always" or "globally"). Hereafter, we denote by $|\varphi|$ the size of φ, that is, the number of its symbols/subformulas.

[2]Hereafter, we denote by $\exists/\forall$ the existential/universal path quantifiers (instead of by the usual E/A), in order not to confuse them with the HSmodalities E/A.

Table 1. Allen's Relations and Corresponding HS Modalities

Allen relation	HS	Definition w.r.t. interval structures	Example
MEETS	$\langle A \rangle$	$[x,y]\mathcal{R}_A[v,z] \iff y = v$	
BEFORE	$\langle L \rangle$	$[x,y]\mathcal{R}_L[v,z] \iff y < v$	
STARTED-BY	$\langle B \rangle$	$[x,y]\mathcal{R}_B[v,z] \iff x = v \wedge z < y$	
FINISHED-BY	$\langle E \rangle$	$[x,y]\mathcal{R}_E[v,z] \iff y = z \wedge x < v$	
CONTAINS	$\langle D \rangle$	$[x,y]\mathcal{R}_D[v,z] \iff x < v \wedge z < y$	
OVERLAPS	$\langle O \rangle$	$[x,y]\mathcal{R}_O[v,z] \iff x < v < y < z$	

The logic CTL is the fragment of CTL* where each temporal modality is immediately preceded by a path quantifier, whereas LTL corresponds to the path-quantifier-free fragment of CTL*.

Given a Kripke structure $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$, an infinite path π of $\mathcal{K}$, and a position $i \geq 0$ along π, the satisfaction relation $\mathcal{K}, \pi, i \models \varphi$ for CTL*, written simply $\pi, i \models \varphi$ when $\mathcal{K}$ is clear from the context, is defined as follows (Boolean connectives are treated as usual):

$$
\begin{array}{ll}
\pi, i \models p & \Leftrightarrow p \in \mu(\pi(i)), \\
\pi, i \models \mathsf{X}\varphi & \Leftrightarrow \pi, i+1 \models \varphi, \\
\pi, i \models \varphi_1 \mathsf{U} \varphi_2 & \Leftrightarrow \text{for some } j \geq i : \pi, j \models \varphi_2 \text{ and } \pi, k \models \varphi_1 \text{ for all } i \leq k < j, \\
\pi, i \models \exists \varphi & \Leftrightarrow \text{for some infinite path } \pi' \text{ starting from } \pi(i),\ \pi', 0 \models \varphi.
\end{array}
$$

The MC problem is defined as follows: $\mathcal{K}$ is a model of φ, written $\mathcal{K} \models \varphi$, if for all initial infinite paths π of $\mathcal{K}$, it holds that $\mathcal{K}, \pi, 0 \models \varphi$.

We also consider a variant of CTL*, called *finitary* CTL*, where the path quantifier $\exists$ of CTL* is replaced by the finitary path quantifier $\exists_f$. In this setting, path quantification ranges over the traces (finite paths) starting from the current state. The satisfaction relation $\rho, i \models \varphi$, where ρ is a trace and i is a position along ρ, is similar to that given for CTL* with the only difference of finiteness of paths, and the fact that for a formula $\mathsf{X}\varphi$, $\rho, i \models \mathsf{X}\varphi$ if and only if $i + 1 < |\rho|$ and $\rho, i + 1 \models \varphi$. A Kripke structure $\mathcal{K}$ is a model of a finitary CTL* formula if for each initial trace ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho, 0 \models \varphi$.

The MC problem for both CTL* and LTL is **PSPACE**-complete [18, 40]. It is not difficult to show that, as it happens with finitary LTL [15], MC for finitary CTL* is **PSPACE**-complete as well.

2.3 The Interval Temporal Logic HS

An interval algebra was proposed by Allen in [1] to reason about intervals and their relative order, while a systematic logical study of interval representation and reasoning was done a few years later by Halpern and Shoham, that introduced the interval temporal logic HS featuring one modality for each Allen relation, but equality [20]. Table 1 depicts 6 of the 13 Allen's relations, together with the corresponding HS (existential) modalities. The other 7 relations are the 6 inverse relations (given a binary relation $\mathcal{R}$, the inverse relation $\overline{\mathcal{R}}$ is such that $b\overline{\mathcal{R}}a$ if and only if $a\mathcal{R}b$) and equality.

For a set of proposition letters $\mathcal{AP}$, the formulas ψ of HS are defined as follows:

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle \psi,$$

where $p \in \mathcal{AP}$ and $X \in \{A, L, B, E, D, O, \overline{A}, \overline{L}, \overline{B}, \overline{E}, \overline{D}, \overline{O}\}$. For any modality $\langle X \rangle$, the dual universal modality $[X]\psi$ is defined as $\neg\langle X \rangle \neg\psi$. For any subset of Allen's relations $\{X_1, \ldots, X_n\}$, $\mathsf{X}_1 \cdots \mathsf{X}_n$ denotes the HS fragment featuring (universal and existential) modalities for $X_1, \ldots, X_n$ only.

We assume the *non-strict semantic version of HS*, which admits intervals consisting of a single point.[3] Under such an assumption, all HS modalities can be expressed in terms of $\langle B \rangle$, $\langle E \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ [42]. As an example, $\langle A \rangle$ can be expressed in terms of $\langle E \rangle$ and $\langle \overline{B} \rangle$ as $\langle A \rangle \varphi := ([E]\bot \wedge (\varphi \vee \langle \overline{B} \rangle \varphi)) \vee \langle E \rangle([E]\bot \wedge (\varphi \vee \langle \overline{B} \rangle \varphi))$. We also use the derived operator $\langle G \rangle$ of HS (and its dual $[G]$), which allows one to select arbitrary subintervals of a given interval, and is defined as $\langle G \rangle \psi := \psi \vee \langle B \rangle \psi \vee \langle E \rangle \psi \vee \langle B \rangle \langle E \rangle \psi$.

HS can be viewed as a multi-modal logic with $\langle B \rangle$, $\langle E \rangle$, $\langle \overline{B} \rangle$, and $\langle \overline{E} \rangle$ as primitive modalities and its semantics can be defined over a multi-modal Kripke structure, called *abstract interval model*, where intervals are treated as atomic objects and Allen's relations as binary relations over intervals.

Definition 2.4 (Abstract Interval Model [30]). An *abstract interval model* over $\mathcal{AP}$ is a tuple $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where $\mathbb{I}$ is a set of worlds, $B_{\mathbb{I}}$ and $E_{\mathbb{I}}$ are two binary relations over $\mathbb{I}$, and $\sigma : \mathbb{I} \to 2^{\mathcal{AP}}$ is a labeling function assigning a set of proposition letters to each world.

Let $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$ be an abstract interval model. In the interval setting, $\mathbb{I}$ is interpreted as a set of intervals, $B_{\mathbb{I}}$ and $E_{\mathbb{I}}$ as Allen's relations B (*started-by*) and E (*finished-by*), respectively, and σ assigns to each interval in $\mathbb{I}$ the set of proposition letters that hold over it. Given an interval $I \in \mathbb{I}$, the truth of an HS formula over I is inductively defined as follows (the Boolean connectives are treated as usual):

- $\mathcal{A}, I \models p$ if and only if $p \in \sigma(I)$, for any $p \in \mathcal{AP}$;
- $\mathcal{A}, I \models \langle X \rangle \psi$, for $X \in \{B, E\}$, if and only if 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 $\overline{X} \in \{\overline{B}, \overline{E}\}$, if and only if there exists $J \in \mathbb{I}$ such that $J \, X_{\mathbb{I}} \, I$ and $\mathcal{A}, J \models \psi$.

The next definition shows how to derive an abstract interval model from an interval structure.

Definition 2.5 (Abstract Interval Model Induced by an Interval Structure). An interval structure $\mathcal{IS} = (\mathbb{S}, \sigma)$, with $\mathbb{S} = (X, <)$, *induces* the abstract interval model $\mathcal{A}_{\mathcal{IS}} = (\mathcal{AP}, \mathbb{I}(\mathbb{S}), B_{\mathbb{I}(\mathbb{S})}, E_{\mathbb{I}(\mathbb{S})}, \sigma)$, where $[x, y] \, B_{\mathbb{I}(\mathbb{S})} \, [v, z]$ iff $x = v$ and $z < y$, and $[x, y] \, E_{\mathbb{I}(\mathbb{S})} \, [v, z]$ iff $y = z$ and $x < v$.

For an interval I and an HS formula ψ, we write $\mathcal{IS}, I \models \psi$ to mean that $\mathcal{A}_{\mathcal{IS}}, I \models \psi$.

2.4 Three Semantic Variants of HS for MC

In this section, we define the three variants of HS semantics $\mathsf{HS}_{\mathsf{st}}$ (state-based), $\mathsf{HS}_{\mathsf{ct}}$ (computation-tree-based), and $\mathsf{HS}_{\mathsf{lin}}$ (trace-based) for model checking HS formulas against Kripke structures. For each variant, the related (finite) MC problem consists of deciding whether or not a finite Kripke structure is a model of an HS formula under such a semantic variant.

Let us start with the *state-based variant* [30, 34], where an abstract interval model is naturally associated with a given Kripke structure $\mathcal{K}$ by considering the set of intervals as the set $\mathrm{Trk}_{\mathcal{K}}$ of traces of $\mathcal{K}$.

Definition 2.6 (Abstract Interval Model Induced by a Kripke Structure). The *abstract interval model induced by a Kripke structure* $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$ is $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$, where $\mathbb{I} = \mathrm{Trk}_{\mathcal{K}}$, $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Pref}(\rho)\}$, $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Suff}(\rho)\}$, and $\sigma : \mathbb{I} \to 2^{\mathcal{AP}}$ is such that $\sigma(\rho) = \bigcap_{s \in \mathrm{states}(\rho)} \mu(s)$, for all $\rho \in \mathbb{I}$.

According to the definition of σ, $p \in \mathcal{AP}$ holds over $\rho = s_1 \cdots s_n$ if and only if it holds over all the states $s_1, \ldots, s_n$ of ρ. This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval if and only if it holds over all its subintervals [39].

[3] All the results we prove in the article hold for the strict version as well.

Definition 2.7 (State-Based HS—$\mathsf{HS}_{\mathsf{st}}$*).* Let $\mathcal{K}$ be a Kripke structure and ψ be an HS formula. A trace $\rho \in \mathrm{Trk}_{\mathcal{K}}$ satisfies ψ under the state-based semantic variant, denoted as $\mathcal{K}, \rho \models_{\mathsf{st}} \psi$, if it holds that $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$. Moreover, $\mathcal{K}$ *is a model of* ψ *under the state-based semantic variant*, denoted as $\mathcal{K} \models_{\mathsf{st}} \psi$, if for all *initial* traces $\rho \in \mathrm{Trk}^0_{\mathcal{K}}$, it holds that $\mathcal{K}, \rho \models_{\mathsf{st}} \psi$.

We now introduce the *computation-tree-based semantic variant*, where we simply consider the abstract interval model *induced by the computation tree* of the Kripke structure. Notice that since each state in a computation tree has a unique predecessor (with the exception of the initial state), this HS variant enforces a linear reference in the past.

Definition 2.8 (Computation-Tree-Based HS—$\mathsf{HS}_{\mathsf{ct}}$*).* A Kripke structure $\mathcal{K}$ *is a model of an* HS *formula* ψ *under the computation-tree-based semantic variant*, written $\mathcal{K} \models_{\mathsf{ct}} \psi$, if $C(\mathcal{K}) \models_{\mathsf{st}} \psi$.

Finally, we define the *trace-based semantic variant*, which exploits the interval structures induced by the infinite paths of the Kripke structure.

Definition 2.9 (Interval Structure Induced by an Infinite Path). For a Kripke structure $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$ and an infinite path $\pi = \pi(0)\pi(1)\cdots$ of $\mathcal{K}$, the *interval structure induced by* π is $IS_{\mathcal{K},\pi} = ((\mathbb{N}, <), \sigma)$, where for each interval $[i, j]$, $\sigma([i, j]) = \bigcap_{h=i}^{j} \mu(\pi(h))$.

Definition 2.10 (Trace-Based HS—$\mathsf{HS}_{\mathsf{lin}}$*).* A Kripke structure $\mathcal{K}$ *is a model of an* HS *formula* ψ *under the trace-based semantic variant*, denoted as $\mathcal{K} \models_{\mathsf{lin}} \psi$, if and only if for each initial infinite path π and for each initial interval $[0, i]$, it holds that $IS_{\mathcal{K},\pi}, [0, i] \models \psi$.

In the next sections, we compare the expressiveness of the logics $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{ct}}$, $\mathsf{HS}_{\mathsf{lin}}$, LTL, CTL, and CTL* when interpreted over finite Kripke structures. Given two logics L_1 and L_2, and two formulas $\varphi_1 \in L_1$ and $\varphi_2 \in L_2$, we say that φ_1 in L_1 is *equivalent* to φ_2 in L_2 if, for every finite Kripke structure $\mathcal{K}$, $\mathcal{K}$ is a model of φ_1 in L_1 if and only if $\mathcal{K}$ is a model of φ_2 in L_2. We say that L_2 *is subsumed by* L_1, denoted as $L_1 \geq L_2$, if for each formula $\varphi_2 \in L_2$, there exists a formula $\varphi_1 \in L_1$ such that φ_1 in L_1 is equivalent to φ_2 in L_2. Moreover, L_1 is *as expressive as* L_2 (or L_1 and L_2 have *the same expressive power*), written $L_1 \equiv L_2$, if both $L_1 \geq L_2$ and $L_2 \geq L_1$. We say that L_1 is *(strictly) more expressive than* L_2 if $L_1 \geq L_2$ and $L_2 \not\geq L_1$. Finally, L_1 and L_2 are *expressively incomparable* if both $L_1 \not\geq L_2$ and $L_2 \not\geq L_1$.

2.5 An Example: A Vending Machine

In this section, we give an example highlighting the differences among the HS semantic variants $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{ct}}$, and $\mathsf{HS}_{\mathsf{lin}}$.

The Kripke structure of Figure 7 represents a *vending machine*, which can dispense water, hot dogs, and candies. In state s_0 (the initial one), no coin has been inserted into the machine (hence, the proposition letter $p_{\$=0}$ holds there). Three edges, labeled by "ins_$1," "ins_$2,"and "ins_$0.50," connect s_0 to s_1, s_2, and s_3, respectively. Edge labels do not convey semantic value (they are neither part of the structure definition nor associated with proposition letters) and are simply used for an easy reference to edges. In s_1 (s_2, s_3, respectively) the proposition letter $p_{\$=1}$ ($p_{\$=2}$, $p_{\$=0.50}$, respectively) holds, representing the fact that 1 Dollar (2, 0.50 Dollars, respectively) has been inserted into the machine. The cost of a bottle of water (a candy, a hot dog, respectively) is $0.50 ($1, $2. respectively). A state s_i, for $i = 1, 2, 3$, is connected to a state s_j, for $j = 4, 5, 6$, only if the available credit allows one to buy the corresponding item. Then, edges labeled by "dispensed" connect s_4, s_5, and s_6 to s_7. In s_7, the machine gives change, and can nondeterministically move back to s_0 (ready for dispensing another item), or to s_8, where it begins an automatic maintenance activity (p_{maint} holds there). Afterward, state s_9 is reached, where maintenance ends. From there, if the maintenance activity fails (edge "maint_failed"), s_8 is reached again (another maintenance cycle is attempted); otherwise, maintenance concludes successfully ("maint_success") and s_0 is reached.

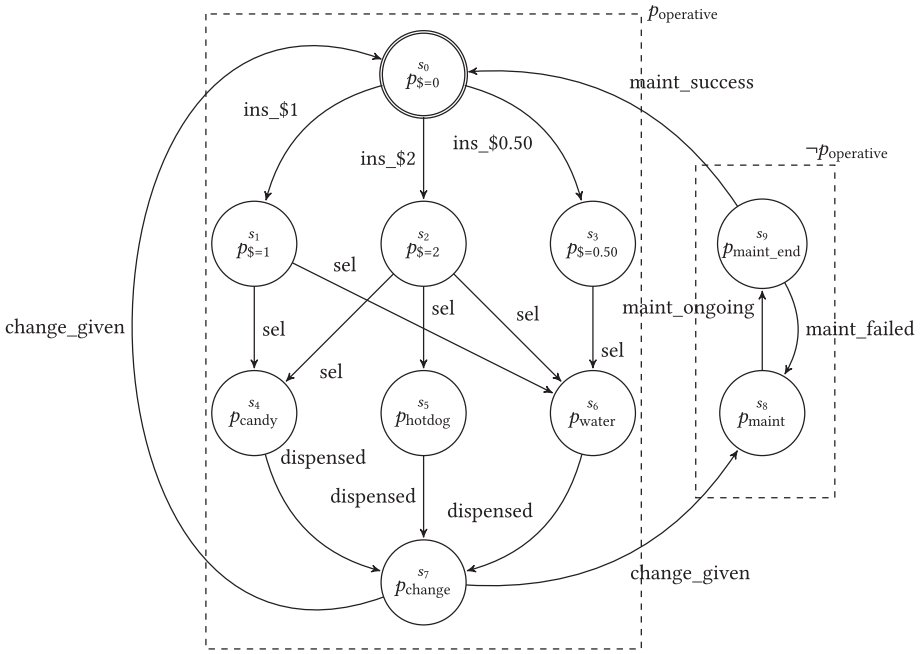


Fig. 7. Kripke structure representing a vending machine.

Since the machine is operating in states $s_0, \ldots, s_7$, and under maintenance in s_8 and s_9, $p_{\text{operative}}$ holds over the former, and it does not on the latter.

In the following, we will make use of the B formulas $length_n$, with $n \geq 1$: for any given n, $length_n$ characterizes the intervals of length n, and is defined as follows:

$$length_n := (\underbrace{\langle \mathrm{B} \rangle \ldots \langle \mathrm{B} \rangle}_{n-1} \top) \ \wedge \ (\underbrace{[B] \ldots [B]}_{n} \bot).$$

We now give some examples of properties we can formalize under all, or some, of the HS semantic variants $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{ct}}$, and $\mathsf{HS}_{\mathsf{lin}}$.

—In any run of length 50, during which the machine never enters maintenance mode, it dispenses at least a hot dog, a bottle of water, and a candy.

$$\mathcal{K} \not\models (p_{\text{operative}} \wedge length_{=50}) \longrightarrow \Big((\langle \mathrm{B} \rangle \langle \mathrm{E} \rangle p_{\text{hotdog}}) \wedge (\langle \mathrm{B} \rangle \langle \mathrm{E} \rangle p_{\text{water}}) \wedge (\langle \mathrm{B} \rangle \langle \mathrm{E} \rangle p_{\text{candy}}) \Big).$$

Clearly this property is false, as the machine can possibly dispense only one or two kinds of items. We start by observing that the above formula is equivalent in all of the three semantic variants of HS: since modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ only allow one to "move" from an interval to its subintervals, $\mathsf{BE}_{\mathsf{lin}}$, $\mathsf{BE}_{\mathsf{st}}$, and $\mathsf{BE}_{\mathsf{ct}}$ coincide (for this reason, we have omitted the subscript from the symbol $\models$). Homogeneity plays a fundamental role here: asking $p_{\text{operative}}$ to be true implies that such a letter is true along the whole trace (thus s_8 and s_9 are always avoided). It is worth observing that the same property can be expressed in LTL, for instance as follows:

$$\bigwedge_{i \in \{0, \ldots, 49\}} \mathsf{X}^i p_{\text{operative}} \wedge \bigvee_{i,j,k \in \{1, \ldots, 48\}, i \neq j \neq k \neq i} (\mathsf{X}^i p_{\text{hotdog}}) \wedge (\mathsf{X}^j p_{\text{water}}) \wedge (\mathsf{X}^k p_{\text{candy}}).$$

The length of this LTL formula is *exponential* in the number of items (in this case, three), whereas the length of the above HS one is only linear. As a matter of fact, we will prove (Theorem 3.5) that BE is at least exponentially more succinct than LTL.

—If the credit is \$0.50, then no hot dog or candy may be provided.

$$\mathcal{K} \models (\langle \mathrm{E} \rangle p_{\$=0.50}) \longrightarrow \neg \langle \mathrm{A} \rangle (length_{=2} \wedge \langle \mathrm{E} \rangle (p_{\mathrm{hotdog}} \vee p_{\mathrm{candy}})).$$

We observe that a trace satisfies $\langle \mathrm{E} \rangle p_{\$=0.50}$ if and only if it ends in s_3. This property is satisfied under all of the three semantic variants, even though the nature of future differs among them (recall Figures 1, 2, and 3). As we have already mentioned, a linear setting (rather than branching) is suitable for the specification of dynamic behaviors, because it considers states *of a computation*; conversely, a branching approach focuses on machine states (and thus on the structure of a system).

In this case, only the state s_6 can be reached from s_3, regardless of the nature of future. For this reason, $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{ct}}$, and $\mathsf{HS}_{\mathsf{lin}}$ behave in the same way.

—Let us exemplify now a difference between $\mathsf{HS}_{\mathsf{st}}$ (and $\mathsf{HS}_{\mathsf{ct}}$) and $\mathsf{HS}_{\mathsf{lin}}$.

$$\begin{array}{l} \mathcal{K} \models_{\mathsf{st}} \\ \mathcal{K} \models_{\mathsf{ct}} (\langle \mathrm{E} \rangle p_{\mathrm{maint\_end}}) \longrightarrow \langle \mathrm{A} \rangle \langle \mathrm{E} \rangle p_{\mathrm{operative}}, \\ \mathcal{K} \not\models_{\mathsf{lin}} \end{array}$$

This is a structural property, requiring that when the machine enters state s_9 (where maintenance ends), it can become again operative reaching state s_0 (s_9 is not a lock state for the system). This is clearly true when future is branching and it is not when future is linear: $\mathsf{HS}_{\mathsf{lin}}$ refers to system computations, and some of these may ultimately loop between s_8 and s_9.

—Conversely, some properties make sense only if they are predicated over computations. This is the case, for instance, of fairness.

$$\begin{array}{l} \mathcal{K} \models_{\mathsf{st}} \\ \mathcal{K} \models_{\mathsf{ct}} ([A] \langle \mathrm{A} \rangle \langle \mathrm{E} \rangle p_{\mathrm{maint}}) \longrightarrow [A] \langle \mathrm{A} \rangle \langle \mathrm{E} \rangle p_{\mathrm{operative}} \\ \mathcal{K} \not\models_{\mathsf{lin}} \end{array}$$

Assuming the trace-based semantics, the property requires that if a system computation enters infinitely often into maintenance mode, it will infinitely often enter operation mode. Again, this is not true, as some system computations may ultimately loop between s_8 and s_9 (hence, they are not fair). On the contrary, such a property is trivially true under $\mathsf{HS}_{\mathsf{st}}$ or $\mathsf{HS}_{\mathsf{ct}}$, as, for any initial trace ρ, it holds that $\mathcal{K}, \rho \models \langle \mathrm{A} \rangle \langle \mathrm{E} \rangle p_{\mathrm{operative}}$.

—We conclude with a property showing the difference between linear and branching *past*, that is, between $\mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{lin}}$ (and $\mathsf{HS}_{\mathsf{ct}}$). The requirement is the following: the machine may dispense water with any amount of (positive) credit.

$$\begin{array}{l} \mathcal{K} \models_{\mathsf{st}} \\ \mathcal{K} \not\models_{\mathsf{ct}} (\langle \mathrm{E} \rangle p_{\mathrm{water}}) \longrightarrow \langle \mathrm{E} \rangle \left(p_{\mathrm{water}} \wedge \bigwedge_{p \in \{p_{\$=2}, p_{\$=1}, p_{\$=0.50}\}} \langle \overline{\mathrm{A}} \rangle (length_{=2} \wedge \langle \mathrm{B} \rangle p) \right) \\ \mathcal{K} \not\models_{\mathsf{lin}} \end{array}$$

Again, this one is a structural property, that cannot be expressed in $\mathsf{HS}_{\mathsf{lin}}$ or $\mathsf{HS}_{\mathsf{ct}}$, as these refer to a specific computation in the past. Conversely, it is true under $\mathsf{HS}_{\mathsf{st}}$, since s_6 is backward reachable in one step by s_1, s_2, and s_3.

3 EQUIVALENCE BETWEEN LTL AND $\mathsf{HS}_{\mathsf{LIN}}$

In this section, we show that $\mathsf{HS}_{\mathsf{lin}}$ is as expressive as LTL even for small syntactical fragments of $\mathsf{HS}_{\mathsf{lin}}$. To this end, we exploit the well-known equivalence between LTL and the first-order

fragment of monadic second-order logic over infinite words (FO for short). Recall that, given a countable set $\{x, y, z, \ldots\}$ of (position) variables, the FO formulas φ over a set of proposition letters $\mathcal{AP} = \{p, \ldots\}$ are defined as:

$$\varphi ::= \top \mid p(x) \mid x \leq y \mid x < y \mid \neg\, \varphi \mid \varphi \ \wedge \ \varphi \mid \exists x.\varphi.$$

We interpret FO formulas φ over infinite paths π of Kripke structures $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$. Given a variable *valuation* g, assigning to each variable a position $i \geq 0$, the satisfaction relation $(\pi, g) \models \varphi$ corresponds to the standard satisfaction relation $(\mu(\pi), g) \models \varphi$, where $\mu(\pi)$ is the infinite word over $2^{\mathcal{AP}}$ given by $\mu(\pi(0))\mu(\pi(1)) \cdots$. More precisely, $(\pi, g) \models \varphi$ is inductively defined as follows (we omit the standard rules for the Boolean connectives):

$$\begin{array}{ll} (\pi, g) \models p(x) & \Leftrightarrow p \in \mu(\pi(g(x))), \\ (\pi, g) \models x \ op \ y & \Leftrightarrow g(x) \ op \ g(y), \text{ for } op \in \{<, \leq\}, \\ (\pi, g) \models \exists x.\varphi & \Leftrightarrow (\pi, g[x \leftarrow i]) \models \varphi \text{ for some } i \geq 0, \end{array}$$

where $g[x \leftarrow i](x) = i$ and $g[x \leftarrow i](y) = g(y)$ for $y \neq x$. Note that the satisfaction relation depends only on the values assigned to the variables occurring free in the given formula φ. We write $\pi \models \varphi$ to mean that $(\pi, g_0) \models \varphi$, where $g_0(x) = 0$ for each variable x. An FO sentence is a formula with no free variables. The following is a well-known result (Kamp's theorem [21]).

Proposition 3.1. *Given an FO sentence φ over $\mathcal{AP}$, one can construct an LTL formula ψ such that, for all Kripke structures $\mathcal{K}$ over $\mathcal{AP}$ and infinite paths π, it holds that $\pi \models \varphi$ if and only if $\pi, 0 \models \psi$.*

Given a $\mathsf{HS}_{\mathsf{lin}}$ formula ψ, we now construct an FO sentence ψ_{FO} such that, for all Kripke structures $\mathcal{K}$, $\mathcal{K} \models_{\mathsf{lin}} \psi$ if and only if for each initial infinite path π of $\mathcal{K}$, $\pi \models \psi_{\mathsf{FO}}$.

We start by defining a mapping h assigning to each triple (φ, x, y), consisting of a HS formula φ and two distinct position variables x, y, an FO formula having as free variables x and y. The mapping h returns the FO formula defining the semantics of the HS formula φ interpreted over an interval bounded by the positions x and y.

The function h is homomorphic with respect to the Boolean connectives, and is defined for proposition letters and modal operators as follows (here z is a fresh position variable):

$$\begin{array}{ll} h(p, x, y) & = \forall z.((z \geq x \wedge z \leq y) \rightarrow p(z)), \\ h(\langle \mathrm{E} \rangle \psi, x, y) & = \exists z.(z > x \wedge z \leq y \wedge h(\psi, z, y)), \\ h(\langle \mathrm{B} \rangle \psi, x, y) & = \exists z.(z \geq x \wedge z < y \wedge h(\psi, x, z)), \\ h(\langle \overline{\mathrm{E}} \rangle \psi, x, y) & = \exists z.(z < x \wedge h(\psi, z, y)), \\ h(\langle \overline{\mathrm{B}} \rangle \psi, x, y) & = \exists z.(z > y \wedge h(\psi, x, z)). \end{array}$$

It is worth noting that homogeneity plays a crucial role in the definition of $h(p, x, y)$ (without it, a binary predicate would be necessary to encode the truth of p over $[x, y]$).

Given a Kripke structure $\mathcal{K}$, an infinite path π, an interval of positions $[i, j]$, and an $\mathsf{HS}_{\mathsf{lin}}$ formula ψ, by a straightforward induction on the structure of ψ, we can show that $IS_{\mathcal{K},\pi}, [i, j] \models \psi$ if and only if $(\pi, g) \models h(\psi, x, y)$ for any valuation such that $g(x) = i$ and $g(y) = j$.

Now, let us consider the FO sentence $h(\psi)$ given by $\exists x((\forall z.z \geq x) \wedge \forall y.h(\psi, x, y))$. Clearly, $\mathcal{K} \models_{\mathsf{lin}} \psi$ if and only if for each initial infinite path π of $\mathcal{K}$, $\pi \models h(\psi)$. By Proposition 3.1, it follows that one can construct an LTL formula $h'(\psi)$ such that $h'(\psi)$ in LTL is equivalent to ψ in $\mathsf{HS}_{\mathsf{lin}}$. Thus, we obtain the following expressiveness containment.

Theorem 3.2. $LTL \geq HS_{\mathsf{lin}}$.

Now we show that also the converse containment holds, that is, LTL can be translated into $\mathsf{HS}_{\mathsf{lin}}$. Actually, it is worth noting that for such a purpose the fragment AB of $\mathsf{HS}_{\mathsf{lin}}$, featuring only modalities for A and B, is expressive enough.

Theorem 3.3. *Given an LTL formula φ, one can construct in linear time an* AB *formula ψ such that φ in LTL is equivalent to ψ in* $\mathsf{AB}_{\mathsf{lin}}$.

Proof. Let $f : \mathsf{LTL} \to \mathsf{AB}$ be the mapping, homomorphic with respect to the Boolean connectives, defined as follows:

$$f(p) = p, \text{for each proposition letter } p,$$
$$f(\mathsf{X}\psi) = \langle \mathrm{A} \rangle (\mathit{length}_2 \wedge \langle \mathrm{A} \rangle (\mathit{length}_1 \wedge f(\psi))),$$
$$f(\psi_1 \mathsf{U} \psi_2) = \langle \mathrm{A} \rangle \Big(\langle \mathrm{A} \rangle (\mathit{length}_1 \wedge f(\psi_2)) \wedge [B](\langle \mathrm{A} \rangle (\mathit{length}_1 \wedge f(\psi_1))) \Big).$$

Given a Kripke structure $\mathcal{K}$, an infinite path π, a position $i \geq 0$, and an LTL formula ψ, by a straightforward induction on the structure of ψ we can show that $\pi, i \models \psi$ if and only if $IS_{\mathcal{K},\pi}, [i, i] \models f(\psi)$. Hence, $\mathcal{K} \models \psi$ if and only if $\mathcal{K} \models_{\mathsf{lin}} \mathit{length}_1 \to f(\psi)$. □

The next corollary follows immediately from Theorem 3.2 and Theorem 3.3.

Corollary 3.4. *HS_{lin} and LTL have the same expressive power.*

While there is no difference in the expressive power between LTL and $\mathsf{HS}_{\mathsf{lin}}$, things change if we consider succinctness. Whereas Theorem 3.3 shows that it is possible to convert any LTL formula into an equivalent $\mathsf{HS}_{\mathsf{lin}}$ one in linear time, the following theorem holds.

Theorem 3.5. *HS_{lin} is at least exponentially more succinct than LTL.*

Proof. To prove the statement, it suffices to provide an $\mathsf{HS}_{\mathsf{lin}}$ formula ψ for which there exists no LTL equivalent formula whose size is polynomial in $|\psi|$.

To this end, we restrict our attention to the fragment $\mathsf{BE}_{\mathsf{lin}}$. Since modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ only allow one to "move" from an interval to its subintervals, $\mathsf{BE}_{\mathsf{lin}}$ actually coincides with $\mathsf{BE}_{\mathsf{st}}$, whose MC is known to be hard for **EXPSPACE** [7]. Thus, in particular, it is possible to encode by means of a $\mathsf{BE}_{\mathsf{lin}}$ formula ψ_{cpt} the (unique) computation of a deterministic Turing machine using $b(n) \in O(2^n)$ bits that, when executed on input 0^n, for some natural number $n \geq 1$, counts in binary from 0 to $2^{2^n} - 1$, by repeatedly summing 1, and finally accepts. The length of ψ_{cpt} is *polynomial* in n, and the unique trace which satisfies it (that is, that encodes such a computation) has length $\ell(n) \geq b(n) \cdot 2^{2^n}$.

Conversely, it is known that LTL features a *single-exponential small-model property* [16], stating that, for every *satisfiable* LTL formula φ, there are $u, v \in S^*$ with $|u| \leq 2^{|\varphi|}$ and $|v| \leq |\varphi| \cdot 2^{|\varphi|}$, such that $u \cdot v^\omega, 0 \models \varphi$. This allows us to conclude (by an easy contradiction argument) that there is no polynomial-length (w.r.t. $|\psi_{\mathsf{cpt}}|$, and thus to n) LTL formula that can encode the aforementioned computation. An *exponential-length* LTL formula would be needed for such an encoding. □

Exactly the same argument can be used to show that $\mathsf{HS}_{\mathsf{lin}}$ is at least exponentially more succinct than the extension of LTL with past modalities (denoted in the following as LTL_P) [24].

4 A CHARACTERIZATION OF $\mathsf{HS}_{\mathsf{CT}}$

In this section, we will focus our attention on the computation-tree-based semantic variant $\mathsf{HS}_{\mathsf{ct}}$, showing that it is as expressive as *finitary* CTL*. As a matter of fact, the result can be proved to hold already for the syntactical fragment ABE which does not feature transposed modalities. In addition, we show that $\mathsf{HS}_{\mathsf{ct}}$ is subsumed by CTL*.

4.1 From Finitary CTL* to $\mathsf{HS}_{\mathsf{ct}}$

We first show that finitary CTL* is subsumed by $\mathsf{HS}_{\mathsf{ct}}$. As a preliminary fundamental step, we prove that when interpreted over finite words, the BE fragment of HS and LTL define the same class of finitary languages (Theorem 4.5).

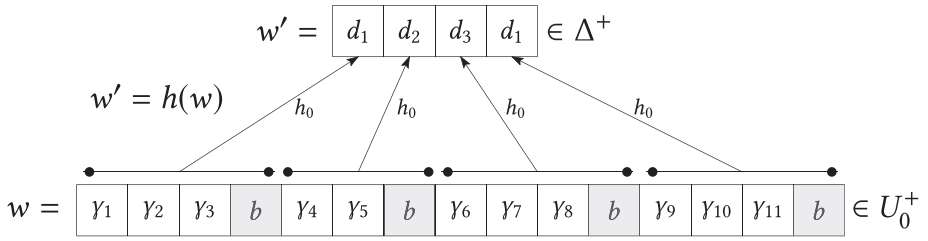


Fig. 8. Visual description of condition 3 of Definition 4.1 (LTL-closure).

For an LTL formula φ with proposition letters over an alphabet Σ (in our case Σ is $2^{\mathcal{AP}}$), let us denote by $L_{act}(\varphi)$ the set of non-empty finite words over Σ satisfying φ under the standard action-based semantics of LTL, interpreted over finite words (see [41]). A similar notion can be given for BE formulas φ with proposition letters in Σ (under the homogeneity assumption). Then, φ denotes a language, written $L_{act}(\varphi)$, of non-empty finite words over Σ inductively defined as

- $L_{act}(a) = a^+$, for $a \in \Sigma$ (we observe that this definition reflects the homogeneity assumption);
- $L_{act}(\neg\varphi) = \Sigma^+ \setminus L_{act}(\varphi)$;
- $L_{act}(\varphi_1 \wedge \varphi_2) = L_{act}(\varphi_1) \cap L_{act}(\varphi_2)$;
- $L_{act}(\langle \mathrm{B} \rangle \varphi) = \{w \in \Sigma^+ \mid \mathrm{Pref}(w) \cap L_{act}(\varphi) \neq \emptyset\}$;
- $L_{act}(\langle \mathrm{E} \rangle \varphi) = \{w \in \Sigma^+ \mid \mathrm{Suff}(w) \cap L_{act}(\varphi) \neq \emptyset\}$.

We prove that, under the action-based semantics, BE formulas and LTL formulas define the same class of finitary languages.

To prove that the finitary languages defined by LTL formulas are subsumed by those defined by BE formulas, we exploit an algebraic condition introduced by Wilke in [43], called *LTL-closure*, which gives, for a class of finitary languages, a sufficient condition to guarantee the inclusion of the class of LTL-definable languages. The converse inclusion, that is, the class of finitary languages defined by the fragment BE is subsumed by that defined by LTL, can be proved by a technique similar to that used in Section 3, and thus omitted.

We start by considering the former inclusion recalling from [43] a sufficient condition for a class of finitary languages to include the class of finitary languages which are LTL-definable.

Definition 4.1 (LTL-Closure). A class C of languages of finite words over finite alphabets is *LTL-closed* if and only if the following conditions are satisfied, where Σ and Δ are finite alphabets, $b \in \Sigma$ and $\Gamma = \Sigma \setminus \{b\}$:

(1) C is closed under language complementation and language intersection;
(2) if $L \in C$ with $L \subseteq \Gamma^+$, then $\Sigma^* bL$, $\Sigma^* b(L + \varepsilon)$, $Lb\Sigma^*$, $(L + \varepsilon)b\Sigma^*$ are in C;
(3) let $U_0 = \Gamma^* b$, $h_0 : U_0 \to \Delta$, and $h : U_0^+ \to \Delta^+$ be defined by $h(u_0 u_1 \cdots u_n) = h_0(u_0) \cdots h_0(u_n)$. Assume that for each $d \in \Delta$, the language $L_d = \{u \in \Gamma^+ \mid h_0(ub) = d\}$ is in C. Then, for each language $L \in C$ such that $L \subseteq \Delta^+$, the language $\Gamma^* b h^{-1}(L)\Gamma^*$ is in C.

In Figure 8, we graphically depict condition (3) of the definition of LTL-closure. In the proposed example, we have (i) for all i, $d_i \in \Delta$ and $\gamma_i \in \Gamma$, (ii) $w = (\gamma_1\gamma_2\gamma_3 b)(\gamma_4\gamma_5 b)(\gamma_6\gamma_7\gamma_8 b)(\gamma_9\gamma_{10}\gamma_{11} b) \in U_0^4$, and (iii) $w' = h(w) = h_0(\gamma_1\gamma_2\gamma_3 b)h_0(\gamma_4\gamma_5 b)h_0(\gamma_6\gamma_7\gamma_8 b)h_0(\gamma_9\gamma_{10}\gamma_{11} b) = d_1 d_2 d_3 d_1 \in \Delta^4$. For instance, $\gamma_1\gamma_2\gamma_3, \gamma_9\gamma_{10}\gamma_{11} \in L_{d_1}$ and $\gamma_4\gamma_5 \in L_{d_2}$.

The following result holds [43].

Theorem 4.2. *Any LTL-closed class C of finitary languages includes the class of LTL-definable finitary languages.*

Therefore, to prove that the finitary languages defined by BE formulas subsume those defined by LTL, as stated by Theorem 4.5 below, it suffices to prove that *the class of finitary languages definable by* BE *formulas is LTL-closed*, and to apply Theorem 4.2. We observe that, by definition, the class of BE-definable languages is obviously closed under language complementation and intersection (condition (1) of Definition 4.1). The fulfillment of conditions (2) and (3) of Definition 4.1 is then proved by the two following Lemmata 4.3 and 4.4, respectively.

Lemma 4.3. *Let Σ be a finite alphabet, $b \in \Sigma$, $\Gamma = \Sigma \setminus \{b\}$, $L \subseteq \Gamma^+$, and ψ be a* BE *formula over Γ such that $L_{act}(\psi) = L$. Then, there are* BE *formulas defining (under the action-based semantics) the languages bL, $\Sigma^* bL$, $\Sigma^* b(L + \varepsilon)$, Lb, $Lb\Sigma^*$, $(L + \varepsilon)b\Sigma^*$, and bLb.*

Proof. We focus on the cases for the languages bL, $\Sigma^* bL$, $\Sigma^* b$, and bLb (for the other languages, the proof is similar: $\Sigma^* b(L + \varepsilon) = \Sigma^* bL + \Sigma^* b$, Lb is symmetric to bL, $Lb\Sigma^*$ to $\Sigma^* bL$, and $(L + \varepsilon)b\Sigma^*$ to $\Sigma^* b(L + \varepsilon)$). Let ψ be a BE formula over Γ such that $L_{act}(\psi) = L$.

Language bL. The BE formula defining the language bL is the formula

$$(\neg length_1 \wedge \langle \mathrm{B} \rangle\, b \wedge [E](\neg b \wedge [B]\neg b)) \wedge h_b(\psi), \tag{1}$$

where the formula $h_b(\psi)$ is inductively defined on the structure of ψ in the following way. The mapping h_b is homomorphic with respect to the Boolean connectives, while for the atomic actions in Γ and the modalities $\langle \mathrm{E} \rangle$ and $\langle \mathrm{B} \rangle$, it is defined as follows:

— for all $a \in \Gamma$, $h_b(a) = a \vee (\langle \mathrm{B} \rangle\, b \wedge \langle \mathrm{E} \rangle\, a \wedge [E]a)$;
— $h_b(\langle \mathrm{B} \rangle\, \theta) = (\langle \mathrm{B} \rangle\, h_b(\theta) \wedge \neg \langle \mathrm{B} \rangle\, b) \vee \langle \mathrm{B} \rangle (h_b(\theta) \wedge \langle \mathrm{B} \rangle\, b)$;
— $h_b(\langle \mathrm{E} \rangle\, \theta) = (\langle \mathrm{E} \rangle\, h_b(\theta) \wedge \neg \langle \mathrm{B} \rangle\, b) \vee (\langle \mathrm{B} \rangle\, b \wedge \langle \mathrm{E} \rangle \langle \mathrm{E} \rangle\, h_b(\theta))$.

The first conjunct of the formula of (1) ensures that a word u' in the defined language has length at least 2 and it has the form bu without any occurrence of b in u. The second conjunct $h_b(\psi)$ ensures that u belongs to the language defined by ψ. For atomic actions and temporal modalities, $h_b(\psi)$ is a disjunction of two possible choices; the appropriate one is forced at top level by the first conjunct of the formula of (1), that constrains one and only one b to occur in the word in the first position.

By a straightforward structural induction on ψ, it can be shown that the following fact holds.

Claim 1. Let $u \in \Gamma^+$, $u' = bu$, and $|u| = n + 1$. Then, for all $i, j \in [0, n]$ with $i \leq j$, $u[i, j] \in L_{act}(\psi)$ if and only if $u'[\hat{i}, j + 1] \in L_{act}(h_b(\psi))$, where $\hat{i} = i$ if $i = 0$, and $\hat{i} = i + 1$ otherwise.

By Claim 1, for each $u \in \Gamma^+$, $u \in L_{act}(\psi)$ if and only if $bu \in L_{act}(h_b(\psi))$. Therefore, the formula of (1) captures the language $bL_{act}(\psi)$.

Languages $\Sigma^* bL$ and $\Sigma^* b$. Following the proof given for the case of the language bL, with $L \subseteq \Gamma^+$, one can construct a BE formula φ defining the language bL. Hence, the BE formula $\varphi \vee \langle \mathrm{E} \rangle\, \varphi$ defines $\Sigma^* bL$. The BE formula defining $\Sigma^* b$ is $b \vee \langle \mathrm{E} \rangle\, b$.

Language bLb. By the proof given for the language bL, with $L \subseteq \Gamma^+$, one can build a BE formula φ defining the language bL. The BE formula defining the language bLb is the formula

$$(\neg length_1 \wedge \neg length_2 \wedge \langle \mathrm{B} \rangle\, b \wedge \langle \mathrm{E} \rangle\, b \wedge [E][B]\neg b) \wedge k_b(\varphi), \tag{2}$$

where the formula $k_b(\varphi)$ is inductively defined on the structure of φ in the following way. The mapping k_b is homomorphic with respect to the Boolean connectives, while for the atomic actions in Σ and the modalities $\langle \mathrm{E} \rangle$ and $\langle \mathrm{B} \rangle$, it is defined as follows:

— for all $a \in \Gamma$, $k_b(a) = a \vee (\langle \mathrm{E} \rangle b \wedge \langle \mathrm{B} \rangle a \wedge [B]a)$;
— $k_b(b) = b$;
— $k_b(\langle \mathrm{B} \rangle \theta) = (\langle \mathrm{B} \rangle k_b(\theta) \wedge \neg \langle \mathrm{E} \rangle b) \vee (\langle \mathrm{E} \rangle b \wedge \langle \mathrm{B} \rangle \langle \mathrm{B} \rangle k_b(\theta))$;
— $k_b(\langle \mathrm{E} \rangle \theta) = (\langle \mathrm{E} \rangle k_b(\theta) \wedge \neg \langle \mathrm{E} \rangle b) \vee \langle \mathrm{E} \rangle (k_b(\theta) \wedge \langle \mathrm{E} \rangle b)$.

The first conjunct of the formula of (2) ensures that a word u' in the defined language has length at least 3 and it has the form bub without any occurrence of b in u. The second conjunct $k_b(\varphi)$ ensures that bu belongs to the language defined by φ. Similarly to the case of the language bL, for atomic actions (different from b) and temporal modalities, $k_b(\psi)$ is a disjunction of two possible choices; the appropriate one is forced at top level by the first conjunct of the formula of (2), that constrains one and only one b to occur in the word in the last position.

By a straightforward structural induction on φ, it can be shown that the following fact holds.

Claim 2. Let $u \in \Gamma^+$ and $|bu| = n + 1$. Then, for all $i, j \in [0, n]$ with $i \leq j$, $bu[i, j] \in L_{act}(\varphi)$ if and only if $bub[i, \hat{j}] \in L_{act}(k_b(\varphi))$ where $\hat{j} = j$ if $j < n$, and $\hat{j} = n + 1$ otherwise.

By Claim 2, for each $u \in \Gamma^+$, $bu \in L_{act}(\varphi)$ if and only if $bub \in L_{act}(k_b(\varphi))$ implying that the formula of (2) defines the language $L_{act}(\varphi)b$. This concludes the proof of the lemma. □

Lemma 4.4. *Let Σ and Δ be finite alphabets, $b \in \Sigma$, $\Gamma = \Sigma \setminus \{b\}$, $U_0 = \Gamma^* b$, $h_0 : U_0 \to \Delta$ and $h : U_0^+ \to \Delta^+$ be defined by $h(u_0 u_1 \cdots u_n) = h_0(u_0) \cdots h_0(u_n)$. Assume that, for each $d \in \Delta$, there is a* BE *formula capturing the language $L_d = \{u \in \Gamma^+ \mid h_0(ub) = d\}$. Then, for each* BE *formula φ over Δ, one can construct a* BE *formula over Σ capturing the language $\Gamma^* b h^{-1}(L_{act}(\varphi))\Gamma^*$.*

Proof. By hypothesis and Lemma 4.3, for each $d \in \Delta$ there exists a BE formula θ_d over Σ defining the language $bL_d b$, where $L_d = \{u \in \Gamma^+ \mid h_0(ub) = d\}$. Hence, there is a BE formula $\hat{\theta}_d$ over Σ capturing the language $b\hat{L}_d b$, where $\hat{L}_d = \{u \in \Gamma^* \mid h_0(ub) = d\}$ (note that $L_d = \hat{L}_d \setminus \{\varepsilon\}$).

Let φ be a BE formula over Δ. By structural induction over φ, we construct a BE formula φ^+ over Σ such that $L_{act}(\varphi^+) = \Gamma^* b h^{-1}(L_{act}(\varphi))\Gamma^*$. The formula φ^+ is defined as follows:

— $\varphi = d$ with $d \in \Delta$. We have that $L_{act}(d) = d^+$ and $\Gamma^* b h^{-1}(L_{act}(d))\Gamma^*$ is the set of finite words in $\Gamma^* b \Sigma^* b \Gamma^*$ such that each subword $u[i, j]$ of u which is in $b\Gamma^* b$ is in $b\hat{L}_d b$ as well. Using the formula $\psi_b := \neg length_1 \wedge \langle \mathrm{B} \rangle b \wedge \langle \mathrm{E} \rangle b \wedge [E][B]\neg b$ to define the language $b\Gamma^* b$, φ^+ is defined as follows:
$$\varphi^+ = (\langle \mathrm{G} \rangle \psi_b) \wedge [G](\psi_b \to \hat{\theta}_d).$$
— $\varphi = \neg\theta$. We have that
$$\Gamma^* b h^{-1}(L_{act}(\varphi))\Gamma^* = \Gamma^* b h^{-1}(\Delta^+ \setminus L_{act}(\theta))\Gamma^* = \Gamma^* b h^{-1}(\Delta^+)\Gamma^* \cap \overline{\Gamma^* b h^{-1}(L_{act}(\theta))\Gamma^*},$$
where $\Gamma^* b h^{-1}(\Delta^+)\Gamma^*$ restricts the set of "candidate" models to the well-formed ones. Thus, taking ψ_b as defined in the previous case, φ^+ is given by
$$\varphi^+ = (\langle \mathrm{G} \rangle \psi_b) \wedge [G]\left(\psi_b \to \bigvee_{d \in \Delta} \hat{\theta}_d\right) \wedge \neg\theta^+,$$
where, by the inductive hypothesis, $L_{act}(\theta^+) = \Gamma^* b h^{-1}(L_{act}(\theta))\Gamma^*$.
— $\varphi = \theta \wedge \psi$. We simply have $\varphi^+ = \theta^+ \wedge \psi^+$.
— $\varphi = \langle \mathrm{B} \rangle \theta$. First, we note that $\Gamma^* b h^{-1}(L_{act}(\langle \mathrm{B} \rangle \theta))\Gamma^*$ is the set of finite words in the language $\Gamma^* b h^{-1}(L_{act}(\theta))h^{-1}(\Delta^+)\Gamma^*$, which is included in the language $\Gamma^* b h^{-1}(\Delta^+)\Gamma^*$ defined by the formula $[G](\psi_b \to \bigvee_{d \in \Delta} \hat{\theta}_d)$. Note also that, by the inductive hypothesis, $\Gamma^* b h^{-1}(L_{act}(\theta))$ is included in the language of θ^+. Thus, φ^+ is given by
$$\varphi^+ = [G]\left(\psi_b \to \bigvee_{d \in \Delta} \hat{\theta}_d\right) \wedge (\xi \vee \langle \mathrm{B} \rangle \xi),$$

where $\xi = (\langle \mathrm{E} \rangle\, b) \wedge \langle \mathrm{B} \rangle(\theta^+ \wedge \langle \mathrm{E} \rangle\, b)$.

— $\varphi = \langle \mathrm{E} \rangle\, \theta$. $\Gamma^* b h^{-1}(L_{act}(\langle \mathrm{E} \rangle\, \theta))\Gamma^*$ is the set $\Gamma^* b h^{-1}(\Delta^+) h^{-1}(L_{act}(\theta))\Gamma^*$ included in the language $\Gamma^* b h^{-1}(\Delta^+)\Gamma^*$, symmetrically to the previous case. Thus, φ^+ is given by

$$\varphi^+ = [G]\left(\psi_b \rightarrow \bigvee_{d \in \Delta} \hat{\theta}_d\right) \wedge (\xi' \vee \langle \mathrm{E} \rangle\, \xi'),$$

where $\xi' = (\langle \mathrm{B} \rangle\, b) \wedge \langle \mathrm{E} \rangle(\theta^+ \wedge \langle \mathrm{B} \rangle\, b)$. □

Since, by Lemmata 4.3 and 4.4, the class of finitary languages definable by BE formulas is LTL-closed, by Theorem 4.2 we get the following result.

Theorem 4.5. *Let φ be an LTL formula over a finite alphabet Σ. Then, there exists a* BE *formula φ_{HS} over Σ such that $L_{act}(\varphi_{HS}) = L_{act}(\varphi)$.*

The result expressed in Theorem 4.5 above is used to prove that finitary CTL* is subsumed by the fragment ABE under the state-based semantics.

Theorem 4.6. *Let φ be a finitary CTL* formula over $\mathcal{AP}$. Then, there is an* ABE *formula φ_{HS} over $\mathcal{AP}$ such that for all Kripke structures $\mathcal{K}$ over $\mathcal{AP}$ and traces ρ, $\mathcal{K}, \rho, 0 \models \varphi$ if and only if $\mathcal{K}, \rho \models_{\mathrm{st}} \varphi_{HS}$.*

Proof. The proof is by induction on the nesting depth of modality $\exists_f$ in φ. In the base case, φ is a finitary LTL formula over $\mathcal{AP}$. Since what we need to deal with it is just the first part of the work we have to do for the inductive step, it is omitted and only the inductive step is detailed.

Let H be the non-empty set of sub-formulas of φ of the form $\exists_f \psi$ which do not occur in the scope of the path quantifier $\exists_f$, that is, the $\exists_f \psi$ formulas which are maximal with respect to the nesting depth of modality $\exists_f$. Then, φ can be seen as an LTL formula over the extended set of proposition letters $\overline{\mathcal{AP}} = \mathcal{AP} \cup H$. Let $\Sigma = 2^{\overline{\mathcal{AP}}}$ and $\overline{\varphi}$ be the LTL formula over Σ obtained from φ by replacing the occurrences of each proposition letter $p \in \overline{\mathcal{AP}}$ in φ with the formula $\bigvee_{P \in \Sigma\, :\, p \in P} P$, according to the LTL action-based semantics.

Given a Kripke structure $\mathcal{K}$ over $\mathcal{AP}$ with labeling μ and a trace ρ of $\mathcal{K}$, we denote by ρ_H the finite word over $2^{\overline{\mathcal{AP}}}$ of length $|\rho|$ defined as $\rho_H(i) = \mu(\rho(i)) \cup \{\exists_f \psi \in H \mid \mathcal{K}, \rho, i \models \exists_f \psi\}$, for all $i \in [0, |\rho| - 1]$. One can easily prove by structural induction on $\overline{\varphi}$ that $\mathcal{K}, \rho, 0 \models \varphi$ if and only if $\rho_H \in L_{act}(\overline{\varphi})$. By Theorem 4.5, there exists a BE formula $\overline{\varphi}_{\mathrm{HS}}$ over Σ such that $L_{act}(\overline{\varphi}) = L_{act}(\overline{\varphi}_{\mathrm{HS}})$.

Now, by the induction hypothesis, for each formula $\exists_f \psi \in H$, there exists an ABE formula ψ_{HS} such that for all Kripke structures $\mathcal{K}$ and traces ρ of $\mathcal{K}$, $\mathcal{K}, \rho, 0 \models \psi$ iff $\mathcal{K}, \rho \models_{\mathrm{st}} \psi_{\mathrm{HS}}$. Since ρ is arbitrary, $\mathcal{K}, \rho, i \models \exists_f \psi$ iff $\mathcal{K}, \rho[i, i], 0 \models \exists_f \psi$ iff $\mathcal{K}, \rho[i, i] \models_{\mathrm{st}} \langle \mathrm{A} \rangle\, \psi_{\mathrm{HS}}$, for each $i \geq 0$.

Let φ_{HS} be the ABE formula over $\mathcal{AP}$ obtained from the BE formula $\overline{\varphi}_{\mathrm{HS}}$ by replacing each occurrence of $P \in \Sigma$ in $\overline{\varphi}_{\mathrm{HS}}$ with the formula

$$[G]\left(\mathit{length}_1 \longrightarrow \bigwedge_{\exists_f \psi \in H \cap P} \langle \mathrm{A} \rangle\, \psi_{\mathrm{HS}} \;\wedge\; \bigwedge_{\exists_f \psi \in H \setminus P} \neg \langle \mathrm{A} \rangle\, \psi_{\mathrm{HS}} \;\wedge\; \bigwedge_{p \in \mathcal{AP} \cap P} p \;\wedge\; \bigwedge_{p \in \mathcal{AP} \setminus P} \neg p\right).$$

Since for all $i \geq 0$ and $\exists_f \psi \in H$, $\mathcal{K}, \rho, i \models \exists_f \psi$ if and only if $\mathcal{K}, \rho[i, i] \models_{\mathrm{st}} \langle \mathrm{A} \rangle\, \psi_{\mathrm{HS}}$, it is possible to prove by a straightforward induction on the structure of $\overline{\varphi}_{\mathrm{HS}}$ that, for any Kripke structure $\mathcal{K}$ and trace ρ of $\mathcal{K}$ we have $\mathcal{K}, \rho \models_{\mathrm{st}} \varphi_{\mathrm{HS}}$ if and only if $\rho_H \in L_{act}(\overline{\varphi}_{\mathrm{HS}})$.

Therefore, since $\mathcal{K}, \rho, 0 \models \varphi$ if and only if $\rho_H \in L_{act}(\overline{\varphi})$ and $L_{act}(\overline{\varphi}) = L_{act}(\overline{\varphi}_{\mathrm{HS}})$, $\mathcal{K}, \rho, 0 \models \varphi$ if and only if $\mathcal{K}, \rho \models_{\mathrm{st}} \varphi_{\mathrm{HS}}$, for any Kripke structure $\mathcal{K}$ and trace ρ of $\mathcal{K}$. □

Since the fragment ABE of HS does not feature any modalities unraveling a Kripke structure backward (namely, $\langle \overline{\mathrm{A}} \rangle$ and $\langle \overline{\mathrm{E}} \rangle$), the computation-tree-based semantics coincides with the

state-based one (recall Figures 1 and 2), and thus the next corollary immediately follows from Theorem 4.6.

Corollary 4.7. *Finitary CTL* is subsumed by both HS_{st} and HS_{ct}.*

4.2 From HS_{ct} to Finitary CTL*

We show now that HS_{ct} is subsumed by both CTL* and its finitary variant. To prove this result, we first introduce a hybrid and linear-past extension of CTL*, called *hybrid* CTL^*_{lp}, and its finitary variant, called *finitary hybrid* CTL^*_{lp}.

Besides standard modalities, hybrid logics make use of explicit variables and quantifiers that bind them [4]. Variables and binders allow us to easily mark points in a path, which will be considered as starting and ending points of intervals, thus permitting a natural encoding of HS_{ct}. Actually, we will show that the restricted use of variables and binders exploited in our encoding does not increase the expressive power of (finitary) CTL* (as it happens for an unrestricted use), thus proving the desired result. We start defining *hybrid* CTL^*_{lp}.

For a countable set $\{x, y, z, \ldots\}$ of (position) variables, the set of formulas φ of hybrid CTL^*_{lp} over $\mathcal{AP}$ is defined as follows:

$$\varphi ::= \top \mid p \mid x \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow x.\varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \mathsf{X}^{-}\varphi \mid \varphi \mathsf{U}^{-}\varphi \mid \exists\varphi,$$

where X^- ("previous") and U^- ("since") are the past counterparts of the "next" and "until" modalities X and U, and $\downarrow x$ is the *downarrow binder operator* [4], which binds x to the current position along the given initial infinite path. We also use the standard shorthands $F^-\varphi := \top \mathsf{U}^-\varphi$ ("eventually in the past") and its dual $G^-\varphi := \neg \mathsf{F}^-\neg\varphi$ ("always in the past"). As usual, a sentence is a formula with no free variables.

Let $\mathcal{K}$ be a Kripke structure and φ be a hybrid CTL^*_{lp} formula. For an *initial* infinite path π of $\mathcal{K}$, a variable valuation g, that assigns to each variable x a position along π, and $i \geq 0$, the satisfaction relation $\pi, g, i \models \varphi$ is defined as follows (we omit the clauses for Boolean connectives, for U and X):

$$\begin{array}{ll}
\pi, g, i \models \mathsf{X}^{-}\varphi & \Leftrightarrow i > 0 \text{ and } \pi, g, i-1 \models \varphi, \\
\pi, g, i \models \varphi_1 \mathsf{U}^{-} \varphi_2 & \Leftrightarrow \text{ for some } j \leq i, \pi, g, j \models \varphi_2 \text{ and } \pi, g, k \models \varphi_1 \text{ for all } j < k \leq i, \\
\pi, g, i \models \exists\varphi & \Leftrightarrow \text{for some initial infinite path } \pi' \text{ such that } \pi'[0, i] = \pi[0, i],\ \pi', g, i \models \varphi, \\
\pi, g, i \models x & \Leftrightarrow g(x) = i, \\
\pi, g, i \models \downarrow x.\varphi & \Leftrightarrow \pi, g[x \leftarrow i], i \models \varphi,
\end{array}$$

where $g[x \leftarrow i](x) = i$ and $g[x \leftarrow i](y) = g(y)$ for $y \neq x$. A Kripke structure $\mathcal{K}$ is a model of a formula φ if $\pi, g_0, 0 \models \varphi$, for every initial infinite path π of $\mathcal{K}$, with g_0 the variable evaluation assigning 0 to each variable. Note that the path quantification is "memoryful," that is, it ranges over infinite paths that start at the root and visit the current node of the computation tree. Clearly, the semantics for the syntactical fragment CTL* coincides with the standard one. If we disallow the use of variables and binder modalities, we obtain the logic CTL^*_{lp}, a well-known linear-past extension of CTL* which is as expressive as CTL* [22]. We also consider the finitary variant of hybrid CTL^*_{lp}, where the path quantifier $\exists$ is replaced with the finitary path quantifier $\exists_f$. This logic corresponds to an extension of finitary CTL* and its semantics is similar to that of hybrid CTL^*_{lp} with the exception that path quantification ranges over the *finite* paths (traces) that start at the root and visit the current node of the computation tree.

In the following, we will use the fragment of hybrid CTL^*_{lp} consisting of *well-formed* formulas, namely, formulas φ where

— each subformula $\exists\psi$ of φ has at most one free variable (namely, not bound by the downarrow binder operator);
— each subformula $\exists\psi(x)$ of φ having x as free variable occurs in φ in the context $(F^{-}x) \wedge \exists\psi(x)$.

Intuitively, the above conditions affirm that, for each state subformula $\exists\psi$, the unique free variable (if any) refers to ancestors of the current node in the computation tree.[4]

The notion of a well-formed formula of finitary hybrid CTL^{*}_{lp} is similar: the path quantifier $\exists$ is replaced by its finitary version $\exists_f$.

We first show that $\mathrm{HS}_{\mathrm{ct}}$ can be translated into the well-formed fragment of hybrid CTL^{*}_{lp} (well-formed fragment of finitary hybrid CTL^{*}_{lp}, respectively). Then, we show that this fragment is subsumed by CTL^{*} (finitary CTL^{*}, respectively).

Proposition 4.8. *Given a $\mathrm{HS}_{\mathrm{ct}}$ formula φ, one can construct in linear-time an equivalent well-formed sentence of hybrid CTL^{*}_{lp} (finitary hybrid CTL^{*}_{lp}, respectively).*

Proof. We focus on the translation from $\mathrm{HS}_{\mathrm{ct}}$ into the well-formed fragment of hybrid CTL^{*}_{lp}. The translation from $\mathrm{HS}_{\mathrm{ct}}$ into the well-formed fragment of finitary hybrid CTL^{*}_{lp} is similar, and thus omitted. Let φ be a $\mathrm{HS}_{\mathrm{ct}}$ formula. The desired hybrid CTL^{*}_{lp} sentence is the formula $\downarrow x.\mathsf{G}\,\mathsf{f}(\varphi,\mathsf{x})$, where $f(\varphi,x)$ is a mapping which is homomorphic with respect to the Boolean connectives, and over proposition letters and modalities behaves as follows:

$$
\begin{aligned}
f(p,x) &= \mathsf{G}^{-}((\mathsf{F}^{-}\mathsf{x}) \rightarrow \mathsf{p}),\\
f(\langle \mathrm{B}\rangle \psi, x) &= \mathsf{X}^{-}\mathsf{F}^{-}(\mathsf{f}(\psi,\mathsf{x}) \wedge \mathsf{F}^{-}\mathsf{x}),\\
f(\langle \overline{\mathrm{B}}\rangle \psi, x) &= \exists(\mathsf{XF}\mathsf{f}(\psi,\mathsf{x})) \wedge \mathsf{F}^{-}\mathsf{x},\\
f(\langle \mathrm{E}\rangle \psi, x) &= \downarrow y.F^{-}\Big(x \wedge \mathsf{XF} \downarrow \mathsf{x}.\mathsf{F}(\mathsf{y} \wedge \mathsf{f}(\psi,\mathsf{x}))\Big),\\
f(\langle \overline{\mathrm{E}}\rangle \psi, x) &= \downarrow y.F^{-}\Big((\mathsf{XF}\mathsf{x}) \wedge \downarrow \mathsf{x}.\mathsf{F}(\mathsf{y} \wedge \mathsf{f}(\psi,\mathsf{x}))\Big),
\end{aligned}
$$

where y is a fresh variable.

Clearly $\downarrow x.\mathsf{G}\,\mathsf{f}(\varphi,\mathsf{x})$ is well-formed. The formula $f(\varphi,x)$ intuitively states that φ holds over an interval of the current path that starts at the position (associated with the variable) x and ends at the current position. More formally, let $\mathcal{K}$ be a Kripke structure, $[h,i]$ be an interval of positions, g be a valuation assigning to the variable x the position h, and π be an initial infinite path. By a straightforward induction on the structure of φ, one can show that $\mathcal{K},\pi,g,i \models f(\varphi,x)$ if and only if $C(\mathcal{K}), C(\pi,h,i) \models_{\mathrm{st}} \varphi$, where $C(\pi,h,i)$ denotes the trace of the computation tree $C(\mathcal{K})$ starting from $\pi[0,h]$ and leading to $\pi[0,i]$. Hence, $\mathcal{K}$ is a model of $\downarrow x.\mathsf{G}\,\mathsf{f}(\varphi,\mathsf{x})$ if, for each initial trace ρ of $C(\mathcal{K})$, we have $C(\mathcal{K}),\rho \models_{\mathrm{st}} \varphi$. □

Let LTL_p be the past extension of LTL, obtained by adding the past modalities X^{-} and U^{-}. By exploiting the well-known separation theorem for LTL_p over finite and infinite words [19], which states that any LTL_p formula can be effectively converted into an equivalent Boolean combination of LTL formulas and pure past LTL_p formulas, we can prove that, under the hypothesis of well-formedness, the extensions of CTL^{*} (finitary CTL^{*}, respectively) used to encode $\mathrm{HS}_{\mathrm{ct}}$ formulas do not increase the expressive power of CTL^{*} (finitary CTL^{*}, respectively). Such a result is the fundamental step to prove, together with Proposition 4.8, that CTL^{*} subsumes $\mathrm{HS}_{\mathrm{ct}}$. In addition,

[4]The well-formedness constraint ensures that a formula captures only branching regular requirements. As an example, the formula $\exists\mathsf{F} \downarrow x.\mathsf{G}^{-}(\neg\mathsf{X}^{-}\top \rightarrow \forall\mathsf{F}(x \wedge p))$ is *not* well-formed and requires that there is a level of the computation tree such that each node in the level satisfies p. This represents a non-regular context-free branching requirement (see, e.g., [2]).

paired with Corollary 4.7, it will allow us to state the main result of the section, namely, that $\mathsf{HS}_{\mathrm{ct}}$ and finitary CTL* have the same expressiveness.

Let us now show that the well-formed fragment of hybrid CTL^*_{lp} (finitary hybrid CTL^*_{lp}, respectively) is not more expressive than CTL* (finitary CTL*, respectively). Once more, we focus on the well-formed fragment of hybrid CTL^*_{lp} omitting the similar proof for the finitary variant.

We start with some additional definitions and auxiliary results. A *pure past* LTL_p formula is an LTL_p formula which does not contain occurrences of future temporal modalities. Given two formulas φ and φ' of hybrid CTL^*_{lp}, we say that φ and φ' are *congruent* if, for every Kripke structure $\mathcal{K}$, initial infinite path π, valuation g, and current position i, $\mathcal{K}, \pi, g, i \models \varphi$ if and only if $\mathcal{K}, \pi, g, i \models \varphi'$ (note that congruence is a *stronger* requirement than equivalence).

As usual, for a formula φ of hybrid CTL^*_{lp} with one free variable x, we write $\varphi(x)$. Moreover, since the satisfaction relation depends only on the variables occurring free in the given formula, for $\varphi(x)$ we use the notation $\mathcal{K}, \pi, i \models \varphi(x \leftarrow h)$ to mean that $\mathcal{K}, \pi, g, i \models \varphi$ for any valuation g assigning h to the unique free variable x. For a formula φ of hybrid CTL^*_{lp}, let $\exists \mathit{SubF}(\varphi)$ denote the set of subformulas of φ of the form $\exists\psi$ which do not occur in the scope of the path quantifier $\exists$.

Finally, for technical reasons, we introduce the notion of *simple* hybrid CTL^*_{lp} formula.

Definition 4.9. Given a variable x, a *simple* hybrid CTL^*_{lp} formula ψ with respect to x is a hybrid CTL^*_{lp} formula satisfying the following syntactical constraints:

- x is the unique variable occurring in ψ;
- ψ does *not* contain occurrences of the binder modalities and past temporal modalities;
- $\exists \mathit{SubF}(\psi)$ consists of CTL* formulas.

Intuitively, a *simple* hybrid CTL^*_{lp} (over $\mathcal{AP}$) formula ψ with respect to x can be seen as a CTL* formula over the set of proposition letters $\mathcal{AP} \cup \{x\}$ such that x does not occur in the scope of $\exists$. The next lemma shows that ψ can be further simplified whenever it is paired with the formula $\mathsf{F}^- x$.

Lemma 4.10. *Let ψ be a* simple *hybrid CTL^*_{lp} formula with respect to x. Then, $(\mathsf{F}^- x) \wedge \psi$ is congruent to a formula of the form $(\mathsf{F}^- x) \wedge \xi$, where ξ is a Boolean combination of the atomic formula x and CTL* formulas.*

Proof. Let ψ be a *simple* hybrid CTL^*_{lp} formula with respect to x. From a syntactic point of view, ψ is not, in general, a CTL* formula due to the occurrences of the free variable x. We show that these occurrences can be separated whenever ψ is paired with $\mathsf{F}^- x$, obtaining a Boolean combination of the atomic formula x and CTL* formulas.

The base case with $\psi = x$, $\psi = p \in \mathcal{AP}$, or $\psi = \exists\psi'$ is obvious.

As for the inductive step, let ψ be a Boolean combination of simple hybrid CTL^*_{lp} formulas θ, where θ is either $p \in \mathcal{AP}$, the variable x, a CTL* formula, or a simple hybrid CTL^*_{lp} formula (with respect to x) of the forms $\mathsf{X}\theta_1$ or $\theta_1 \mathsf{U} \theta_2$. Therefore, we just need to consider the cases where $\theta = \mathsf{X}\theta_1$ or $\theta = \theta_1 \mathsf{U} \theta_2$.

Let us consider the case $\theta = \mathsf{X}\theta_1$. Since there are not past temporal modalities in θ_1, $\mathsf{X}\theta_1$ forces the free occurrence of x in ψ to be interpreted in a (strictly) future position. However, ψ is conjunct with the formula $\mathsf{F}^- x$, which turns out to be false when x is associated with a (strictly) future position. Let us denote by $\widehat{\theta}$ the CTL* formula obtained from θ by replacing each occurrence of x in ψ with $\bot$ (false). Now, when x is mapped to a (strictly) future position, $\mathsf{F}^- x$ is false, and, when x is mapped to a present/past position, $\mathsf{F}^- x$ is true, and θ and $\widehat{\theta}$ are congruent. As a consequence, it is clear that $(\mathsf{F}^- x) \wedge \theta$ is congruent to $(\mathsf{F}^- x) \wedge \widehat{\theta}$.

Let us consider the case for $\theta = \theta_1 \mathsf{U} \theta_2$. Using the same arguments of the previous case, we have that $(\mathsf{F}^- x) \wedge \theta$ is congruent to $(\mathsf{F}^- x) \wedge (\theta_2 \vee (\theta_1 \wedge \mathsf{X}(\widehat{\theta_1 \mathsf{U} \theta_2})))$. By distributivity of $\wedge$ over $\vee$, we get $((\mathsf{F}^- x) \wedge \theta_2) \vee ((\mathsf{F}^- x) \wedge \theta_1 \wedge \mathsf{X}(\widehat{\theta_1 \mathsf{U} \theta_2}))$. The thesis follows by applying the inductive hypothesis to $(\mathsf{F}^- x) \wedge \theta_2$ and to $(\mathsf{F}^- x) \wedge \theta_1$, and by factorizing $\mathsf{F}^- x$ (notice that $\widehat{\theta_1 \mathsf{U} \theta_2}$ is a CTL* formula). □

The next lemma states an important technical property of well-formed formulas, which will be exploited in Theorem 4.14 to prove that the set of sentences of the well-formed fragment of hybrid CTL$^*_{lp}$ has the same expressiveness as CTL*. Intuitively, if the hybrid features of the language do not occur in the scope of existential path quantifiers, it is possible to remove the occurrences of the binder $\downarrow$ and to suitably separate past and future modalities. The result is obtained by exploiting the equivalence of FO and LTL$_p$ over infinite words and by applying the separation theorem for LTL$_p$ over infinite words [19], that we recall here for completeness.

Theorem 4.11 (LTL$_p$ Separation over Infinite Words). *Any LTL$_p$ formula ψ can be effectively transformed into a formula*

$$\psi' = \bigvee_{i=1}^{t} (\psi_{p,i} \wedge \psi_{f,i}),$$

for some $t \geq 1$, where $\psi_{p,i}$ is a pure past LTL$_p$ formula and $\psi_{f,i}$ is an LTL formula, such that for all infinite words w over $2^{\mathcal{AP}}$ and $i \geq 0$, it holds that $w, i \models \psi$ if and only if $w, i \models \psi'$.

Lemma 4.12. *Let $(F^- x) \wedge \exists\varphi(x)$ ($\exists\varphi$, respectively) be a well-formed formula (well-formed sentence, respectively) of hybrid CTL$^*_{lp}$ such that $\exists SubF(\varphi)$ consists of CTL* formulas. Then, $(F^- x) \wedge \exists\varphi(x)$ ($\exists\varphi$, respectively) is congruent to a well-formed formula of hybrid CTL$^*_{lp}$ which is a Boolean combination of CTL* formulas and (formulas that correspond to)* pure past *LTL$_p$ formulas over the set of proposition letters $\mathcal{AP} \cup \exists SubF(\varphi) \cup \{x\}$ (., $\mathcal{AP} \cup \exists SubF(\varphi)$, respectively).*

Proof. We focus on well-formed formulas of the form $(\mathsf{F}^- x) \wedge \exists\varphi(x)$. The case of well-formed sentences of the form $\exists\varphi$ is similar, and thus omitted.

Let $\overline{\mathcal{AP}} = \mathcal{AP} \cup \exists SubF(\varphi) \cup \{x\}$. By hypothesis, $\exists SubF(\varphi)$ is a set of CTL* formulas, that is, they are devoid of any hybrid feature.

Given a Kripke structure $\mathcal{K} = (\mathcal{AP}, S, \delta, \mu, s_0)$, an initial infinite path π, and $h \geq 0$, we denote by $\pi_{\overline{\mathcal{AP}},h}$ the infinite word over $2^{\overline{\mathcal{AP}}}$, which, for every position $i \geq 0$, is defined as follows:

- $\pi_{\overline{\mathcal{AP}},h}(i) \cap \mathcal{AP} = \mu(\pi(i))$;
- $\pi_{\overline{\mathcal{AP}},h}(i) \cap \exists SubF(\varphi) = \{\psi \in \exists SubF(\varphi) \mid \mathcal{K}, \pi, i \models \psi\}$;
- $x \in \pi_{\overline{\mathcal{AP}},h}(i)$ if and only if $i = h$.

By using a fresh position variable *present* to represent the current position, the formula $\varphi(x)$ can be easily converted into an FO formula $\varphi_{\mathsf{FO}}(present)$ over $\overline{\mathcal{AP}}$ having *present* as its unique free variable, such that for all Kripke structures $\mathcal{K}$, initial infinite paths π, and positions i, h, we have

$$\mathcal{K}, \pi, i \models \varphi(x \leftarrow h) \text{ if and only if } \pi_{\overline{\mathcal{AP}},h} \models \varphi_{\mathsf{FO}}(present \leftarrow i). \tag{3}$$

(To this end, it suffices to map any proposition letter $\overline{p} \in \overline{\mathcal{AP}}$ into a unary predicate $\overline{p}$, and all the operators $\mathsf{X}, \mathsf{X}^-, \mathsf{U}, \mathsf{U}^-, \downarrow$ into FO formulas expressing their semantics.)

By the equivalence of FO and LTL$_p$ and the separation theorem for LTL$_p$over infinite words (Theorem 4.11), starting from the FO formula $\varphi_{\mathsf{FO}}(present)$, one can construct an LTL$_p$ formula φ_{LTL_p} over $\overline{\mathcal{AP}}$ of the form

$$\varphi_{\mathsf{LTL}_P} := \bigvee_{i \in I} (\varphi_{p,i} \wedge \varphi_{f,i}) \tag{4}$$

such that $\varphi_{p,i}$ is a pure past LTL_p formula, $\varphi_{f,i}$ is an LTL formula, and for all infinite words w over $2^{\overline{\mathcal{AP}}}$ and $i \geq 0$, it holds that

$$w, i \models \varphi_{\mathsf{LTL}_P} \text{ if and only if } \; w \models \varphi_{\mathsf{FO}}(\mathit{present} \leftarrow i). \tag{5}$$

The LTL_p formula φ_{LTL_P} over $\overline{\mathcal{AP}}$ corresponds to a hybrid CTL^*_{lp} formula $\varphi_{\mathsf{LTL}_P}(x)$ over $\mathcal{AP}$. (Note that the only hybrid feature is the possible occurrence of the variable x.) By definition of the infinite words $\pi_{\overline{\mathcal{AP}},h}$, one can easily show by structural induction that for all Kripke structures $\mathcal{K}$, initial infinite paths π, and positions i and h,

$$\pi_{\overline{\mathcal{AP}},h}, i \models \varphi_{\mathsf{LTL}_P} \text{ if and only if } \mathcal{K}, \pi, i \models \varphi_{\mathsf{LTL}_P}(x \leftarrow h), \tag{6}$$

the latter being a hybrid CTL^*_{lp} formula. Thus, by points (3), (5), and (6), we obtain that $\varphi(x)$ and $\varphi_{\mathsf{LTL}_P}(\mathrm{x})$ are congruent.

Since in Equation (4), for each $i \in I$, $\varphi_{p,i}$ is a pure past LTL_p formula over $\overline{\mathcal{AP}}$, $\exists\varphi_{p,i}(x)$ is trivially congruent to $\varphi_{p,i}(x)$. As a consequence, we have that $(\mathsf{F}^-x) \wedge \exists\varphi(x)$ is congruent to $(\mathsf{F}^-x) \wedge \bigvee_{i\in I}(\varphi_{p,i}(x) \wedge \exists\varphi_{f,i}(x))$, which is congruent to $\bigvee_{i\in I}(\varphi_{p,i}(x) \wedge (\mathsf{F}^-x) \wedge \exists\varphi_{f,i}(x))$, which is in turn congruent to $\bigvee_{i\in I}(\varphi_{p,i}(x) \wedge \exists((\mathsf{F}^-x) \wedge \varphi_{f,i}(x)))$.

Now, $\varphi_{f,i}(x)$ is a *simple* hybrid CTL^*_{lp} formula with respect to x, and $\exists x$ ($\exists\neg x$, respectively) is trivially congruent to x ($\neg x$, respectively). By Lemma 4.10 and some simple manipulation steps, we can prove the following sequence of equivalences:

$$\bigvee_{i\in I}(\varphi_{p,i}(x) \wedge \exists((\mathsf{F}^-x) \wedge \varphi_{f,i}(x))) = \qquad \text{(Lemma 4.10 and disjunctive normal form)}$$

$$\bigvee_{i\in I}\left(\varphi_{p,i}(x) \wedge \exists\left((\mathsf{F}^-x) \wedge \bigvee_{j\in J}(\tilde{x}_{i,j} \wedge \psi_{i,j})\right)\right) = \qquad (\mathsf{F}^-x \text{ is a pure past } \mathsf{LTL}_p \text{ formula})$$

$$\bigvee_{i\in I}\left(\varphi_{p,i}(x) \wedge (\mathsf{F}^-x) \wedge \exists\bigvee_{j\in J}(\tilde{x}_{i,j} \wedge \psi_{i,j})\right) = \qquad \text{(Distributive property of } \wedge \text{ over } \vee)$$

$$(\mathsf{F}^-x) \wedge \bigvee_{i\in I}\left(\varphi_{p,i}(x) \wedge \exists\bigvee_{j\in J}(\tilde{x}_{i,j} \wedge \psi_{i,j})\right) =$$

(Distributive property of $\exists$ over $\vee$ and $\tilde{x}_{i,j}$ is a pure past LTL_p formula)

$$(\mathsf{F}^-x) \wedge \bigvee_{i\in I}\left(\varphi_{p,i}(x) \wedge \bigvee_{j\in J}(\tilde{x}_{i,j} \wedge \exists\psi_{i,j})\right) = \qquad \text{(Distributive property of } \wedge \text{ over } \vee)$$

$$(\mathsf{F}^-x) \wedge \bigvee_{i\in I}\bigvee_{j\in J}(\varphi_{p,i}(x) \wedge \tilde{x}_{i,j} \wedge \exists\psi_{i,j})$$

where $\tilde{x}_{i,j}$ is either x, $\neg x$, or $\top$.

Hence, $(\mathsf{F}^-x) \wedge \exists\varphi(x)$ is congruent to a formula of the form $(\mathsf{F}^-x) \wedge \bigvee_{i\in I'}(\psi_{p,i}(x) \wedge \exists\psi_i)$, for some I', where $\psi_{p,i}(x)$ corresponds to a *pure past* LTL_p formula over $\overline{\mathcal{AP}}$ ($= \mathcal{AP} \cup \exists\mathit{SubF}(\varphi) \cup \{x\}$) and ψ_i is a CTL^* formula. □

The following lemma generalizes the separation result given by Lemma 4.12 to any well-formed formula of the form $(\mathsf{F}^-x) \wedge \exists\varphi(x)$, that is, to formulas where $\varphi(x)$ is unconstrained.

Lemma 4.13. *Let $(\mathsf{F}^-x) \wedge \exists\varphi(x)$ ($\exists\varphi$, respectively) be a well-formed formula (well-formed sentence, respectively) of hybrid CTL^*_{lp}. Then, there exists a finite set $\mathcal{H}$ of CTL^* formulas of the form $\exists\psi$, such that $(\mathsf{F}^-x) \wedge \exists\varphi(x)$ ($\exists\varphi$, respectively) is congruent to a well-formed formula of hybrid CTL^*_{lp} which is*

a Boolean combination of CTL formulas and (formulas that correspond to)* pure past LTL_p *formulas over the set of proposition letters* $\mathcal{AP} \cup \mathcal{H} \cup \{x\}$ *(resp.,* $\mathcal{AP} \cup \mathcal{H}$*).*

Proof. As in the case of Lemma 4.12, we focus on well-formed formulas of the form $(\mathsf{F}^{-}x) \wedge \exists\varphi(x)$ (the case of well-formed sentences of the form $\exists\varphi$ is similar).

The proof is by induction on the nesting depth of the path quantifier $\exists$ in $\varphi(x)$.

Base case: $\exists SubF(\varphi) = \emptyset$. We apply Lemma 4.12, and the result follows taking $\mathcal{H} = \emptyset$.

Inductive step: let $\exists\psi \in \exists SubF(\varphi)$. Since $(\mathsf{F}^{-}x) \wedge \exists\varphi(x)$ is well-formed, either ψ is a sentence, or ψ has a unique free variable y and $\exists\psi(y)$ occurs in $\varphi(x)$ in the context $(\mathsf{F}^{-}y) \wedge \exists\psi(y)$. Assume that the latter case holds (the former is similar). By definition of well-formed formula, y is not free in $\varphi(x)$, and $(\mathsf{F}^{-}y) \wedge \exists\psi(y)$ must occur in the scope of some occurrence of $\downarrow y$. By the inductive hypothesis, the thesis holds for $(\mathsf{F}^{-}y) \wedge \exists\psi(y)$. Hence, there exists a finite set $\mathcal{H}'$ of CTL* formulas of the form $\exists\theta$ such that $(\mathsf{F}^{-}y) \wedge \exists\psi(y)$ is congruent to a well-formed formula of hybrid CTL^{*}_{lp}, say $\xi(y)$, which is a Boolean combination of CTL* formulas and formulas that correspond to *pure past* LTL_p formulas over the set of proposition letters $\mathcal{AP} \cup \mathcal{H}' \cup \{y\}$.

By replacing each occurrence of $(\mathsf{F}^{-}y) \wedge \exists\psi(y)$ in $\varphi(x)$ with $\xi(y)$, and repeating the procedure for all the formulas in $\exists SubF(\varphi)$, we obtain a well-formed formula of hybrid CTL^{*}_{lp} of the form $(\mathsf{F}^{-}x) \wedge \exists\theta(x)$ which is congruent to $(\mathsf{F}^{-}x) \wedge \exists\varphi(x)$ (note that the congruence relation is closed under substitution) and such that $\exists SubF(\theta)$ consists of CTL* formulas. At this point, we can apply Lemma 4.12 proving the assertion. □

We can now prove that the well-formed sentences of hybrid CTL^{*}_{lp} can be expressed in CTL*.

Theorem 4.14. *The set of sentences of the well-formed fragment of hybrid* CTL^{*}_{lp} *has the same expressiveness as CTL*.*

Proof. Let φ be a well-formed sentence of hybrid CTL^{*}_{lp}. To prove the thesis, we construct a CTL* formula which is equivalent to φ.

Since φ is equivalent to $\neg\exists\neg\varphi$ and $\neg\exists\neg\varphi$ is well-formed, by applying Lemma 4.13 one can convert $\neg\exists\neg\varphi$ into a congruent hybrid CTL^{*}_{lp} formula which is a Boolean combination of CTL* formulas and formulas θ which can be seen as pure past LTL_pformulas over the set of proposition letters $\mathcal{AP} \cup \mathcal{H}$, where $\mathcal{H}$ is a set of CTL* formulas of the form $\exists\psi$.

Since the past temporal modalities in such LTL_p formulas θ refer to the initial position of the initial infinite paths, one can replace θ with an equivalent CTL* formula $f(\theta)$, where the mapping f is inductively defined as follows:

- $f(p) = p$ for all $p \in \mathcal{AP} \cup \mathcal{H}$;
- f is homomorphic with respect to the Boolean connectives;
- $f(\mathsf{X}^{-}\theta) = \bot$ and $f(\theta_1 \mathsf{U}^{-} \theta_2) = f(\theta_2)$.

The resulting CTL* formula is equivalent to $\neg\exists\neg\varphi$, as required. □

By an easy adaptation of the proof of Theorem 4.14, where one exploits the separation theorem for LTL_p over finite words [19], it is possible to characterize also the expressiveness of well-formed *finitary* hybrid CTL^{*}_{lp}.

Theorem 4.15. *The set of sentences of the well-formed fragment of finitary hybrid* CTL^{*}_{lp} *has the same expressiveness as finitary CTL*.*

Together with Proposition 4.8, Theorem 4.14 (Theorem 4.15, respectively) allows us to conclude that CTL* (finitary CTL*, respectively) subsumes $\mathsf{HS}_{\mathrm{ct}}$.

Fig. 9. The Kripke structures $\mathcal{K}_1$ and $\mathcal{K}_2$.

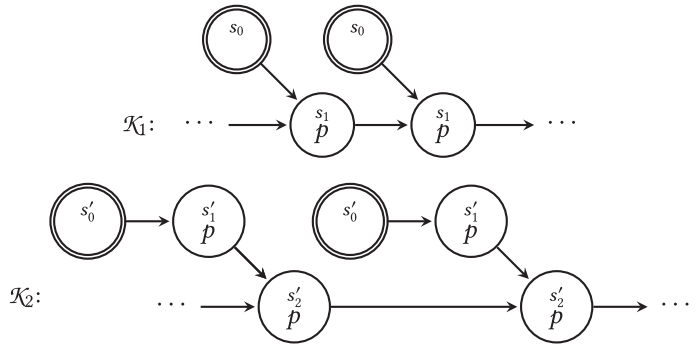


Fig. 10. Forward and backward unwinding of $\mathcal{K}_1$ and $\mathcal{K}_2$.

Finally, by exploiting Corollary 4.7, we can state the main result of the section, namely, $\mathrm{HS}_{\mathrm{ct}}$ and finitary CTL* have the same expressiveness.

Theorem 4.16. *CTL* $\geq$ HS$_{\mathrm{ct}}$. Moreover, HS$_{\mathrm{ct}}$ is as expressive as finitary CTL*.*

5 EXPRESSIVENESS COMPARISON OF $\mathrm{HS}_{\mathrm{LIN}}$, $\mathrm{HS}_{\mathrm{ST}}$, AND $\mathrm{HS}_{\mathrm{CT}}$

In this section, we compare the expressiveness of the three semantic variants of HS, namely, $\mathrm{HS}_{\mathrm{lin}}$, $\mathrm{HS}_{\mathrm{st}}$, and $\mathrm{HS}_{\mathrm{ct}}$. The resulting picture was anticipated in Figure 4. Here, we give the proofs of the depicted results.

We start showing that $\mathrm{HS}_{\mathrm{st}}$ is *not* subsumed by $\mathrm{HS}_{\mathrm{ct}}$. As a matter of fact, we show that $\mathrm{HS}_{\mathrm{st}}$ is sensitive to backward unwinding of finite Kripke structures, allowing us to sometimes discriminate finite Kripke structures with the same computation tree (these structures are always indistinguishable by $\mathrm{HS}_{\mathrm{ct}}$).

Let us consider, for instance, the two finite Kripke structures $\mathcal{K}_1$ and $\mathcal{K}_2$ of Figure 9, whose forward and backward unwinding is shown in Figure 10. Since $\mathcal{K}_1$ and $\mathcal{K}_2$ have the same computation tree, no HS formula φ under the computation-tree-based semantics can distinguish $\mathcal{K}_1$ and $\mathcal{K}_2$, that is, $\mathcal{K}_1 \models_{\mathrm{ct}} \varphi$ if and only if $\mathcal{K}_2 \models_{\mathrm{ct}} \varphi$. On the other hand, the requirement "*each state reachable from the initial one where p holds has a predecessor where p holds as well*" can be expressed, under the state-based semantics, by the HS formula $\psi := \langle \mathrm{E} \rangle (p \wedge \mathit{length}_1) \rightarrow \langle \mathrm{E} \rangle (\mathit{length}_1 \wedge \langle \overline{\mathrm{A}} \rangle (p \wedge \neg \mathit{length}_1))$. It is easy to see that $\mathcal{K}_1 \models_{\mathrm{st}} \psi$: for any initial trace ρ of $\mathcal{K}_1$, we have $\mathcal{K}_1, \rho \models_{\mathrm{st}} \langle \mathrm{E} \rangle (p \wedge \mathit{length}_1)$ iff $\rho = s_0 s_1^k$ for $k \geq 1$; the length-1 suffix s_1 is *met-by* $s_1 s_1$, and $\mathcal{K}_1, s_1 s_1 \models_{\mathrm{st}} p \wedge \neg \mathit{length}_1$.

On the contrary, in $\mathcal{K}_2$ there is an initial trace, $s'_0 s'_1$, for which $\mathcal{K}_2, s'_0 s'_1 \models_{\mathrm{st}} \langle \mathrm{E} \rangle (p \wedge \mathit{length}_1)$; however, the only traces that meet the length-1 suffix s'_1 are s'_1 itself and $s'_0 s'_1$, but neither of them model $p \wedge \neg \mathit{length}_1$. Therefore, $\mathcal{K}_2 \not\models_{\mathrm{st}} \psi$. This allows us to prove the following proposition.

Proposition 5.1. *HS$_{\mathrm{ct}}$ $\not\geq$ HS$_{\mathrm{st}}$.*

Since, as stated by Theorem 4.16, $\mathrm{HS}_{\mathrm{ct}}$ and finitary CTL* have the same expressiveness and finitary CTL* is subsumed by $\mathrm{HS}_{\mathrm{st}}$ (see Corollary 4.7), by Proposition 5.1 the next corollary follows.

Corollary 5.2. *HS$_{\mathrm{st}}$ is more expressive than HS$_{\mathrm{ct}}$.*

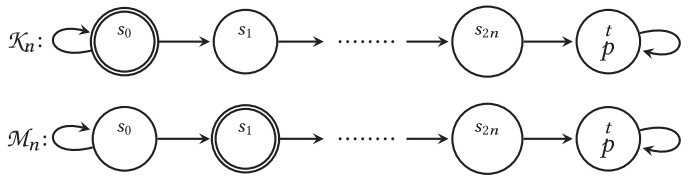


Fig. 11. The Kripke structures $\mathcal{K}_n$ and $\mathcal{M}_n$ with $n \geq 1$.

In the following, we focus on the comparison of $\mathsf{HS}_{\mathsf{lin}}$ with $\mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{ct}}$ showing that $\mathsf{HS}_{\mathsf{lin}}$ is incomparable with both $\mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{ct}}$.

The fact that $\mathsf{HS}_{\mathsf{lin}}$ does not subsume either $\mathsf{HS}_{\mathsf{st}}$ or $\mathsf{HS}_{\mathsf{ct}}$ can be easily proved as follows. Consider the CTL formula $\forall\mathsf{G}\exists\mathsf{F}p$ asserting that from each state reachable from the initial one, it is possible to reach a state where p holds. It is well-known that this formula is not LTL-definable (see [3], Theorem 6.21). Thus, by Corollary 3.4, there is no equivalent $\mathsf{HS}_{\mathsf{lin}}$ formula. On the other hand, the requirement $\forall\mathsf{G}\exists\mathsf{F}p$ can be trivially expressed under the state-based (computation-tree-based, respectively) semantics by the HS formula $\langle\overline{\mathrm{B}}\rangle\langle\mathrm{E}\rangle p$, proving the following result.

Proposition 5.3. *$\mathsf{HS}_{\mathsf{lin}} \not\geq \mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{lin}} \not\geq \mathsf{HS}_{\mathsf{ct}}$.*

To prove the converse, namely, that $\mathsf{HS}_{\mathsf{lin}}$ is not subsumed either by $\mathsf{HS}_{\mathsf{st}}$ or by $\mathsf{HS}_{\mathsf{ct}}$, we will show that the LTL formula $\mathsf{F}\,p$ (equivalent to the CTL formula $\forall\mathsf{F}\,p$) cannot be expressed in either $\mathsf{HS}_{\mathsf{ct}}$ or $\mathsf{HS}_{\mathsf{st}}$. The proof is rather involved and requires a number of definitions and intermediate results. We work it out for the state-based semantics only, because the one for the computation-tree-based semantics is very similar.

Let us start by defining two families of Kripke structures $(\mathcal{K}_n)_{n\geq 1}$ and $(\mathcal{M}_n)_{n\geq 1}$ over $\{p\}$ such that for all $n \geq 1$, the LTL formula $\mathsf{F}\,p$ distinguishes $\mathcal{K}_n$ and $\mathcal{M}_n$, and for every HS formula ψ of size at most n, ψ does *not* distinguish $\mathcal{K}_n$ and $\mathcal{M}_n$ under the state-based semantics.

For a given $n \geq 1$, the Kripke structures $\mathcal{K}_n$ and $\mathcal{M}_n$ are depicted in Figure 11. Notice that the Kripke structure $\mathcal{M}_n$ differs from $\mathcal{K}_n$ only in that its initial state is s_1 instead of s_0. Formally, $\mathcal{K}_n = (\{p\}, S_n, \delta_n, \mu_n, s_0)$, and $\mathcal{M}_n = (\{p\}, S_n, \delta_n, \mu_n, s_1)$, with $S_n = \{s_0, s_1, \ldots, s_{2n}, t\}$, $\delta_n = \{(s_0, s_0), (s_0, s_1), \ldots, (s_{2n-1}, s_{2n}), (s_{2n}, t), (t, t)\}$, $\mu(s_i) = \emptyset$ for all $0 \leq i \leq 2n$, and $\mu(t) = \{p\}$.

Now, it is immediate to see that $\mathcal{K}_n \not\models Fp$ and $\mathcal{M}_n \models Fp$.

On the contrary, we are going to prove that $\mathcal{K}_n \models_{\mathsf{st}} \psi$ if and only if $\mathcal{M}_n \models_{\mathsf{st}} \psi$ for all balanced $\mathsf{HS}_{\mathsf{st}}$ formulas ψ of length at most n with $n \geq 1$. An $\mathsf{HS}_{\mathsf{st}}$ formula ψ is *balanced* if, for each subformula $\langle\mathrm{B}\rangle\theta$ ($\langle\overline{\mathrm{B}}\rangle\theta$, respectively), θ has the form $\theta_1 \wedge \theta_2$ with $|\theta_1| = |\theta_2|$. Proving the result for balanced $\mathsf{HS}_{\mathsf{st}}$ formulas allows us to state it for any $\mathsf{HS}_{\mathsf{st}}$ formula, since it is possible to trivially convert an $\mathsf{HS}_{\mathsf{st}}$ formula ψ into a balanced one (by using conjunctions of $\top$) which is equivalent to ψ under any of the considered HS semantic variants.

To prove such a result, we need some technical definitions. Let ρ be a trace of $\mathcal{K}_n$ (note that $\mathcal{K}_n$ and $\mathcal{M}_n$ feature the same traces). By construction, ρ has the form $\rho' \cdot \rho''$, where ρ' is a (possibly empty) trace visiting only states where p does not hold, and ρ'' is a (possibly empty) trace visiting only the state t, where p holds. We say that ρ' (ρ'', respectively) is the $\emptyset$-part (p-part, respectively) of ρ. Let $N_\emptyset(\rho)$, $N_p(\rho)$, and $D_p(\rho)$ be the natural numbers defined as follows:

- $N_\emptyset(\rho) = |\rho'|$ (the length of the $\emptyset$-part of ρ);
- $N_p(\rho) = |\rho''|$ (the length of the p-part of ρ);
- $D_p(\rho) = 0$ if $N_p(\rho) > 0$ (i.e., $\mathrm{lst}(\rho) = t$); otherwise, $D_p(\rho)$ is the length of the minimal trace starting from $\mathrm{lst}(\rho)$ and leading to s_{2n}. Note that $D_p(\rho)$ is well-defined and $0 \leq D_p(\rho) \leq 2n + 1$.

By construction, the following property holds.

Proposition 5.4. *For all traces ρ and ρ' of $\mathcal{K}_n$, if $D_p(\rho) = D_p(\rho')$, then* $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$.

Now, for each $h \in [1, n]$, we introduce the notion of *h-compatibility* between traces of $\mathcal{K}_n$. Intuitively, this notion provides a sufficient condition to make two traces indistinguishable under the state-based semantics by means of balanced HS formulas having size at most h.

Definition 5.5 (h-compatibility). Let $h \in [1, n]$. Two traces ρ and ρ' of $\mathcal{K}_n$ are *h-compatible* if the following conditions hold:

— $N_p(\rho) = N_p(\rho')$;
— either $N_\emptyset(\rho) = N_\emptyset(\rho')$, or $N_\emptyset(\rho) \geq h$ and $N_\emptyset(\rho') \geq h$;
— either $D_p(\rho) = D_p(\rho')$, or $D_p(\rho) \geq h$ and $D_p(\rho') \geq h$.

We denote by $R(h)$ the binary relation over the set of traces of $\mathcal{K}_n$ such that $(\rho, \rho') \in R(h)$ if and only if ρ and ρ' are h-compatible. Notice that $R(h)$ is an equivalence relation, for all $h \in [1, n]$. Moreover, $R(h) \subseteq R(h-1)$, for all $h \in [2, n]$, that is, $R(h)$ is a refinement of $R(h-1)$.

By construction, the next property, that will be used to prove Lemma 5.9, can be easily shown.

Proposition 5.6. *For every trace ρ of $\mathcal{K}_n$ starting from s_0 (s_1, respectively), there exists a trace ρ' of $\mathcal{K}_n$ starting from s_1 (s_0, respectively) such that $(\rho, \rho') \in R(n)$.*

The following lemma lists some useful properties of the equivalence relation $R(h)$.

Lemma 5.7. *Let $h \in [2, n]$ and $(\rho, \rho') \in R(h)$. The following properties hold:*

(1) for each proper prefix σ of ρ, there exists a proper prefix σ' of ρ' such that $(\sigma, \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$;
(2) for each trace of the form $\rho \cdot \sigma$, where σ is not empty, there exists a trace of the form $\rho' \cdot \sigma'$ such that σ' is not empty and $(\rho \cdot \sigma, \rho' \cdot \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$;
(3) for each proper suffix σ of ρ, there exists a proper suffix σ' of ρ' such that $(\sigma, \sigma') \in R(h-1)$;
(4) for each trace of the form $\sigma \cdot \rho$, where σ is not empty, there exists a trace of the form $\sigma' \cdot \rho'$ such that σ' is not empty and $(\sigma \cdot \rho, \sigma' \cdot \rho') \in R(h)$.

Proof. We prove Properties 1 and 2. Properties 3 and 4 easily follow by construction and by definition of h-compatibility.

Property 1. We distinguish the following cases:

(1) $D_p(\rho) < h$ and $N_\emptyset(\rho) < h$. Since $(\rho, \rho') \in R(h)$ and $h \in [2, n]$, it holds that $D_p(\rho) = D_p(\rho')$, $N_\emptyset(\rho) = N_\emptyset(\rho')$, and $N_p(\rho) = N_p(\rho')$, and thus $\rho = \rho'$.
(2) $D_p(\rho) \geq h$. Since $(\rho, \rho') \in R(h)$, $D_p(\rho') \geq h$, $N_p(\rho') = N_p(\rho) = 0$, and either $N_\emptyset(\rho') = N_\emptyset(\rho)$, or $N_\emptyset(\rho) \geq h$ and $N_\emptyset(\rho') \geq h$. In both cases, by construction it easily follows that for each proper prefix σ of ρ, there exists a proper prefix σ' of ρ' such that $(\sigma, \sigma') \in R(h-1) \subseteq R(\lfloor \frac{h}{2} \rfloor)$.
(3) $D_p(\rho) < h$ and $N_\emptyset(\rho) \geq h$. Since $(\rho, \rho') \in R(h)$, we have that $D_p(\rho') = D_p(\rho)$ (and hence, by Proposition 5.4, $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$), $N_p(\rho') = N_p(\rho)$, and $N_\emptyset(\rho') \geq h$.
Let σ be a proper prefix of ρ. We distinguish the following three subcases:
  (1) $N_\emptyset(\sigma) < \lfloor \frac{h}{2} \rfloor$. Since $N_\emptyset(\rho) \geq h$, we have that $D_p(\sigma) \geq \lfloor \frac{h}{2} \rfloor$ and $|\sigma| = N_\emptyset(\sigma)$ (and thus $N_p(\sigma) = 0$). Since $N_\emptyset(\rho') \geq h$, by taking the proper prefix σ' of ρ' having length $N_\emptyset(\sigma)$, we obtain that $(\sigma, \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$.
  (2) $N_\emptyset(\sigma) \geq \lfloor \frac{h}{2} \rfloor$ and $D_p(\sigma) \geq \lfloor \frac{h}{2} \rfloor$. By taking the prefix σ' of ρ' of length $\lfloor \frac{h}{2} \rfloor$, we get that $(\sigma, \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$.

(3) $N_\emptyset(\sigma) \geq \lfloor \frac{h}{2} \rfloor$ and $D_p(\sigma) < \lfloor \frac{h}{2} \rfloor$. Since $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$, $N_p(\rho') = N_p(\rho)$, and $N_\emptyset(\rho') \geq h$, there exists a proper prefix σ' of ρ' such that $\mathrm{lst}(\sigma') = \mathrm{lst}(\sigma)$, $N_p(\sigma') = N_p(\sigma)$, and $N_\emptyset(\sigma') \geq \lfloor \frac{h}{2} \rfloor$. Hence, $(\sigma, \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$.

Thus, in all the cases Property 1 holds.

Property 2. Let $(\rho, \rho') \in R(h)$ and σ be a non-empty trace such that $\rho \cdot \sigma$ is a trace. We distinguish the following cases:

(1) $D_p(\rho) < h$. Since $(\rho, \rho') \in R(h)$, we have that $D_p(\rho') = D_p(\rho)$, $N_p(\rho) = N_p(\rho')$, and either $N_\emptyset(\rho') = N_\emptyset(\rho)$, or $N_\emptyset(\rho) \geq h$ and $N_\emptyset(\rho') \geq h$. Hence, $\mathrm{lst}(\rho) = \mathrm{lst}(\rho')$ and, by taking $\sigma' = \sigma$, we obtain that $(\rho \cdot \sigma, \rho' \cdot \sigma') \in R(h) \subseteq R(\lfloor \frac{h}{2} \rfloor)$.

(2) $D_p(\rho) \geq h$ and $D_p(\sigma) < \lfloor \frac{h}{2} \rfloor$. It follows that $N_\emptyset(\rho \cdot \sigma) \geq \lfloor \frac{h}{2} \rfloor$. Since $D_p(\rho') \geq h$, there exists a trace of the form $\rho' \cdot \sigma'$ such that $D_p(\rho' \cdot \sigma') = D_p(\rho \cdot \sigma)$, $N_p(\rho' \cdot \sigma') = N_p(\rho \cdot \sigma)$, and $N_\emptyset(\rho' \cdot \sigma') \geq \lfloor \frac{h}{2} \rfloor$. Hence, $(\rho \cdot \sigma, \rho' \cdot \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$.

(3) $D_p(\rho) \geq h$ and $D_p(\sigma) \geq \lfloor \frac{h}{2} \rfloor$. Thus, $D_p(\rho') \geq h$. If $N_\emptyset(\rho \cdot \sigma) < \lfloor \frac{h}{2} \rfloor$, then $N_\emptyset(\rho) = N_\emptyset(\rho')$. Therefore, there exists a trace of the form $\rho' \cdot \sigma'$ such that $N_\emptyset(\rho' \cdot \sigma') = N_\emptyset(\rho \cdot \sigma)$ and $D_p(\sigma') \geq \lfloor \frac{h}{2} \rfloor$. Otherwise, $N_\emptyset(\rho \cdot \sigma) \geq \lfloor \frac{h}{2} \rfloor$ and there exists a trace of the form $\rho' \cdot \sigma'$ such that $N_\emptyset(\rho' \cdot \sigma') \geq \lfloor \frac{h}{2} \rfloor$ and $D_p(\sigma') = \lfloor \frac{h}{2} \rfloor$. In both cases, $(\rho \cdot \sigma, \rho' \cdot \sigma') \in R(\lfloor \frac{h}{2} \rfloor)$.

Thus, Property 2 holds. □

By exploiting Lemma 5.7, we can prove the following lemma.

Lemma 5.8. *Let n be a natural number, ψ be a balanced HS_{st} formula, with $|\psi| \leq n$, and $(\rho, \rho') \in R(|\psi|)$. Then, $\mathcal{K}_n, \rho \models \psi$ if and only if $\mathcal{K}_n, \rho' \models \psi$.*

Proof. The proof is by induction on $|\psi|$. The cases for the Boolean connectives directly follow from the inductive hypothesis and the fact that $R(h) \subseteq R(k)$, for all $h, k \in [1, n]$ with $h \geq k$.

As for the other cases, we proceed as follows:

— $\psi = p$. Since $(\rho, \rho') \in R(1)$, that is, either $N_\emptyset(\rho) = N_\emptyset(\rho') = 0$ or both $N_\emptyset(\rho) \geq 1$ and $N_\emptyset(\rho') \geq 1$, ρ visits a state where p does not hold if and only if ρ' visits a state where p does not hold, which proves the thesis.

— $\psi = \langle \mathrm{B} \rangle \theta$ ($\psi = \langle \overline{\mathrm{B}} \rangle \theta$, respectively). Since ψ is balanced, θ has the form $\theta = \theta_1 \wedge \theta_2$, with $|\theta_1| = |\theta_2|$. Hence, $|\theta_1|, |\theta_2| \leq \lfloor \frac{|\psi|}{2} \rfloor$. We focus on the case $\psi = \langle \mathrm{B} \rangle \theta$. Since $R(|\psi|)$ is an equivalence relation, by symmetry it suffices to show that $\mathcal{K}_n, \rho \models \psi$ implies $\mathcal{K}_n, \rho' \models \psi$. If $\mathcal{K}_n, \rho \models \psi$, then there exists a proper prefix σ of ρ such that $\mathcal{K}_n, \sigma \models \theta_i$, for $i = 1, 2$. Since $(\rho, \rho') \in R(|\psi|)$, by property (1) of Lemma 5.7, there exists a proper prefix σ' of ρ' such that $(\sigma, \sigma') \in R(\lfloor \frac{|\psi|}{2} \rfloor)$. Since $R(\lfloor \frac{|\psi|}{2} \rfloor) \subseteq R(|\theta_i|)$, for $i = 1, 2$, by the inductive hypothesis we get that $\mathcal{K}_n, \sigma' \models \theta_i$, for $i = 1, 2$, thus proving that $\mathcal{K}_n, \rho' \models \psi$.
The case for $\psi = \langle \overline{\mathrm{B}} \rangle \theta$ can be dealt with similarly by exploiting property (2) of Lemma 5.7.

— $\psi = \langle \mathrm{E} \rangle \theta$ ($\psi = \langle \overline{\mathrm{E}} \rangle \theta$, respectively). We can proceed as in the previous case by applying property (3) of Lemma 5.7 (property (4) of Lemma 5.7, respectively) and the inductive hypothesis. □

Lemma 5.9. *For all natural numbers $n \geq 1$ and balanced HS_{st} formulas ψ, with $|\psi| \leq n$, $\mathcal{K}_n \models_{st} \psi$ if and only if $\mathcal{M}_n \models_{st} \psi$.*

Proof. First, let us assume that $\mathcal{K}_n \not\models_{st} \psi$. Then, there exists an initial trace ρ of $\mathcal{K}_n$ such that $\mathcal{K}_n, \rho \not\models_{st} \psi$. By Proposition 5.6, there exists a trace ρ' of $\mathcal{K}_n$, which is an initial trace for $\mathcal{M}_n$, such that $(\rho, \rho') \in R(|\psi|)$. By Lemma 5.8, we have that $\mathcal{K}_n, \rho' \not\models_{st} \psi$. Since for any trace σ and any HS_{st}

formula φ, we have that $\mathcal{K}_n, \sigma \models_{\mathrm{st}} \varphi$ if and only if $\mathcal{M}_n, \sigma \models_{\mathrm{st}} \varphi$ ($\mathcal{K}_n$ and $\mathcal{M}_n$ feature exactly the same set of traces with exactly the same labeling; they only differ in the initial state), we can conclude that $\mathcal{M}_n, \rho' \not\models_{\mathrm{st}} \psi$, and thus $\mathcal{M}_n \not\models_{\mathrm{st}} \psi$.

Let us now assume that $\mathcal{M}_n \not\models_{\mathrm{st}} \psi$. Then, there exists an initial trace ρ of $\mathcal{M}_n$ such that $\mathcal{M}_n, \rho \not\models_{\mathrm{st}} \psi$. As in the converse direction, we have that $\mathcal{K}_n, \rho \not\models_{\mathrm{st}} \psi$, and, by Proposition 5.6, we can easily find an initial trace ρ' of $\mathcal{K}_n$ such that $(\rho, \rho') \in R(|\psi|)$. By Lemma 5.8, we can conclude that $\mathcal{K}_n \not\models_{\mathrm{st}} \psi$. □

As an immediate consequence of Lemma 5.9 and of the fact that, for each $n \geq 1$, $\mathcal{K}_n \not\models Fp$ and $\mathcal{M}_n \models Fp$, we get the desired undefinability result.

Proposition 5.10. *The LTL formula Fp (equivalent to the CTL formula $\forall Fp$) cannot be expressed in either HS_{ct} or HS_{st}.*

The next proposition immediately follows from Corollary 3.4 and Proposition 5.10.

Proposition 5.11. *$HS_{\mathrm{st}} \not\geq HS_{\mathrm{lin}}$ and $HS_{\mathrm{ct}} \not\geq HS_{\mathrm{lin}}$.*

Putting together Propositions 5.3 and 5.11, we finally obtain the incomparability result.

Theorem 5.12. *HS_{lin} and HS_{st} are expressively incomparable, and so are HS_{lin} and HS_{ct}.*

The proved results also allow us to establish the expressiveness relations between $\mathrm{HS}_{\mathrm{st}}$, $\mathrm{HS}_{\mathrm{ct}}$, and the standard branching temporal logics CTL and CTL*.

Corollary 5.13. *The following expressiveness results hold:*

(1) HS_{st} and CTL* are expressively incomparable;
(2) HS_{st} and CTL are expressively incomparable;
(3) HS_{ct} and finitary CTL* are less expressive than CTL*;
(4) HS_{ct} and CTL are expressively incomparable.

Proof. (Item 1) By Proposition 5.10 and the fact that CTL* is not sensitive to unwinding.
(Item 2) Again, by Proposition 5.10 and the fact that CTL is not sensitive to unwinding.
(Item 3) By Theorem 4.16, $\mathrm{HS}_{\mathrm{ct}}$ is subsumed by CTL*, and $\mathrm{HS}_{\mathrm{ct}}$ and finitary CTL* have the same expressiveness. Hence, by Proposition 5.10, the result follows.
(Item 4) Thanks to Proposition 5.10, it suffices to show that there exists a $\mathrm{HS}_{\mathrm{ct}}$ formula which cannot be expressed in CTL. Let us consider the CTL* formula $\varphi := \exists\Big(((p_1 \mathsf{U} p_2) \vee (q_1 \mathsf{U} q_2)) \mathsf{U}\, r\Big)$ over the set of propositions $\{p_1, p_2, q_1, q_2, r\}$. It is shown in [17] that φ cannot be expressed in CTL. Clearly, if we replace the path quantifier $\exists$ in φ with the finitary path quantifier $\exists_f$, we obtain an equivalent formula of finitary CTL*. Thus, since $\mathrm{HS}_{\mathrm{ct}}$ and finitary CTL* have the same expressiveness (Theorem 4.16), the result follows. □

6 CONCLUSIONS AND FUTURE WORK

In the present article, we compared interval temporal logic model checking with a point-based one with respect to its expressiveness (and succinctness). To this end, we took into consideration three semantic variants of the interval temporal logic HS, namely, $\mathrm{HS}_{\mathrm{st}}$, $\mathrm{HS}_{\mathrm{ct}}$, and $\mathrm{HS}_{\mathrm{lin}}$, under the homogeneity assumption. We investigated their expressiveness and we systematically contrasted them with the point-based temporal logics LTL, CTL, finitary CTL*, and CTL*.

The resulting picture is as follows: $\mathrm{HS}_{\mathrm{lin}}$ and $\mathrm{HS}_{\mathrm{ct}}$ turn out to be as expressive as LTL and finitary CTL*, respectively. Moreover, $\mathrm{HS}_{\mathrm{lin}}$ is at least exponentially more succinct than LTL. $\mathrm{HS}_{\mathrm{st}}$ is expressively incomparable with $\mathrm{HS}_{\mathrm{lin}}$/LTL, CTL, and CTL*, but it is strictly more expressive than

HS_{ct}/finitary CTL*. We believe it possible to fill the expressiveness gap between HS_{ct} and CTL* by considering abstract interval models, induced by Kripke structures, featuring worlds also for infinite traces/intervals, and extending the semantics of HS modalities to infinite intervals. Such an extension will be investigated in future research.

It is worth noting that the decidability of the MC problem for (full) HS_{ct} and HS_{lin} immediately follows from the above results as a byproduct. We leave for future work the study of the related complexity issues, which have been systematically investigated only for HS_{st}.

MC for HS can be extended in various directions. Recently [28], a more general definition of interval labeling, that is, of the behavior of proposition letters over intervals, has been proposed, which allows one to associate a regular expression over the set of states of the Kripke structure with each proposition letter. An in-depth investigation of MC with regular expressions for HS and its fragments can be found in [5, 6], where, in particular, it is shown that MC for full HS_{st} with regular expressions is still (nonelementarily) decidable, and all the sub-fragments of $A\overline{A}B\overline{B}_{st}$ and $A\overline{A}E\overline{E}_{st}$ become complete for **PSPACE**.

Another research direction looks for possible replacements of Kripke structures by more expressive system models. On one hand, we are interested in the investigation of the MC problem for HS over *visibly pushdown systems*, that can encode recursive programs and infinite state systems. On the other, we are thinking of the possibility of devising and exploiting *inherently interval-based models* in system descriptions. Kripke structures, being based on states, are naturally oriented to the representation of the state-by-state evolution of the systems and to the characterization of their point-based properties. To express and check temporal constraints which are inherently interval-based, such as, for instance, those involving temporal aggregations, a different formalism is needed, which allows one to directly model systems on the basis of their interval behavior/properties, thus making it possible to define and benefit from a really general interval-based MC.

REFERENCES

[1] J. F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26, 11 (1983), 832–843. DOI: https://doi.org/10.1145/182.358434

[2] R. Alur, P. Cerný, and S. Zdancewic. 2006. Preserving secrecy under refinement. In *ICALP*, Lecture Notes in Computer Science, Vol. 4052. Springer, 107–118. DOI: https://doi.org/10.1007/11787006_10

[3] C. Baier and J. P. Katoen. 2008. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press.

[4] P. Blackburn and J. Seligman. 1998. What are hybrid languages? In *AiML*. CSLI Publications, 41–62.

[5] L. Bozzelli, A. Molinari, A. Montanari, and A. Peron. 2017. An in-depth investigation of interval temporal logic model checking with regular expressions. In *SEFM*, Lecture Notes in Computer Science, Vol. 10469. Springer, 104–119. DOI: https://doi.org/10.1007/978-3-319-66197-1

[6] L. Bozzelli, A. Molinari, A. Montanari, and A. Peron. 2017. On the complexity of model checking for syntactically maximal fragments of the interval temporal logic HS with regular expressions. In *GandALF*, Electronic Proceedings in Theoretical Computer Science, Vol. 256. EPTCS, 31–45. DOI: https://doi.org/10.4204/EPTCS.256.3

[7] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. 2016. Interval temporal logic model checking: The border between good and bad HS fragments. In *IJCAR* Lecture Notes in Artificial Intelligence, Vol. 9706. Springer, 389–405. DOI: https://doi.org/10.1007/978-3-319-40229-1_27

[8] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. 2016. Interval vs. point temporal logic model checking: An expressiveness comparison. In *FSTTCS*. LIPIcs, 26:1–26:14. DOI: https://doi.org/10.4230/LIPIcs.FSTTCS.2016.26

[9] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. 2016. Model checking the logic of Allen's relations meets and started-by is $\mathbf{P}^{\mathbf{NP}}$-complete. In *GandALF*. EPTCS, 76–90. DOI: https://doi.org/10.4204/EPTCS.226.6

[10] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. 2017. Satisfiability and model checking for the logic of sub-intervals under the homogeneity assumption. In *ICALP (LIPIcs)*, Vol. 80. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 120:1–120:14. DOI: 10.4230/LIPIcs.ICALP.2017.120

[11] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. 2014. The dark side of interval temporal logic: Marking the undecidability border. *Annals of Mathematics and Artificial Intelligence* 71, 1–3 (2014), 41–83. DOI: https://doi.org/10.1007/s10472-013-9376-4

[12] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. 2010. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation* 20, 1 (2010), 133–166. DOI: https://doi.org/10.1093/logcom/exn063

[13] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. 2009. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic* 161, 3 (2009), 289–304. DOI: https://doi.org/10.1016/j.apal.2009.07.003

[14] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. 2011. Optimal tableau systems for propositional neighborhood logic over all, dense, and discrete linear orders. In *TABLEAUX*, Lecture Notes in Computer Science, Vol. 6973. Springer, 73–87. DOI: https://doi.org/10.1007/978-3-642-22119-4_8

[15] G. De Giacomo and M. Y. Vardi. 2013. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI*. IJCAI/AAAI, 854–860.

[16] S. Demri, V. Goranko, and M. Lange. 2016. *Temporal Logics in Computer Science: Finite-State Systems*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9781139236119

[17] E. A. Emerson and J. Y. Halpern. 1986. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM* 33, 1 (1986), 151–178. DOI: https://doi.org/10.1145/4904.4999

[18] E. A. Emerson and C. Lei. 1985. Modalities for model checking: Branching time strikes back. In *PoPL*. Elsevier, 84–96. DOI: https://doi.org/10.1145/318593.318620

[19] D. M. Gabbay. 1987. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, Lecture Notes in Computer Science, Vol. 398. Springer, 409–448. DOI: https://doi.org/10.1007/3-540-51803-7_36

[20] J. Y. Halpern and Y. Shoham. 1991. A propositional modal logic of time intervals. *Journal of the ACM* 38, 4 (1991), 935–962. DOI: https://doi.org/10.1145/115234.115351

[21] H. Kamp. 1968. *Tense Logic and the Theory of Linear Order*. Ph.D. dissertation. Ucla.

[22] O. Kupferman, A. Pnueli, and M. Y. Vardi. 2012. Once and for all. *Journal of Computer and System Sciences* 78, 3 (2012), 981–996. DOI: https://doi.org/10.1016/j.jcss.2011.08.006

[23] F. Laroussinie and Ph. Schnoebelen. 1995. A hierarchy of temporal logics with past. *Theoretical Computer Science* 148, 2 (1995), 303–324. DOI: https://doi.org/10.1016/0304-3975(95)00035-U

[24] O. Lichtenstein and A. Pnueli. 2000. Propositional temporal logics: Decidability and completeness. *Logic Journal of the IGPL* 8, 1 (2000), 55–85. DOI: https://doi.org/10.1093/jigpal/8.1.55

[25] K. Lodaya. 2000. Sharpening the undecidability of interval temporal logic. In *ASIAN*, Lecture Notes in Computer Science, Vol. 1961. Springer, 290–298. DOI: https://doi.org/10.1007/3-540-44464-5_21

[26] A. Lomuscio and J. Michaliszyn. 2013. An epistemic Halpern-Shoham logic. In *IJCAI*. IJCAI/AAAI, 1010–1016.

[27] A. Lomuscio and J. Michaliszyn. 2014. Decidability of model checking multi-agent systems against a class of EHS specifications. In *ECAI*. IOS Press, 543–548. DOI: https://doi.org/10.3233/978-1-61499-419-0-543

[28] A. Lomuscio and J. Michaliszyn. 2016. Model checking multi-agent systems against epistemic HS specifications with regular expressions. In *KR*. AAAI Press, 298–308.

[29] J. Marcinkowski and J. Michaliszyn. 2014. The undecidability of the logic of subintervals. *Fundamenta Informaticae* 131, 2 (2014), 217–240. DOI: https://doi.org/10.3233/FI-2014-1011

[30] A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. 2016. Checking interval properties of computations. *Acta Informatica* 53, 6–8 (2016), 587–619. DOI: https://doi.org/10.1007/s00236-015-0250-1

[31] A. Molinari, A. Montanari, and A. Peron. 2015. Complexity of ITL model checking: Some well-behaved fragments of the interval logic HS. In *TIME*. IEEE Computer Society, 90–100. DOI: https://doi.org/10.1109/TIME.2015.12

[32] A. Molinari, A. Montanari, and A. Peron. 2015. A model checking procedure for interval temporal logics based on track representatives. In *CSL*. LIPIcs, 193–210. DOI: https://doi.org/10.4230/LIPIcs.CSL.2015.193

[33] A. Molinari, A. Montanari, A. Peron, and P. Sala. 2016. Model checking well-behaved fragments of HS: The (almost) final picture. In *KR*. AAAI Press, 473–483.

[34] A. Montanari, A. Murano, G. Perelli, and A. Peron. 2014. Checking interval properties of computations. In *TIME*. IEEE Computer Society, 59–68. DOI: https://doi.org/10.1109/TIME.2014.24

[35] A. Montanari, G. Puppis, and P. Sala. 2015. A decidable weakening of compass logic based on cone-shaped cardinal directions. *Logical Methods in Computer Science* 11, 4 (2015), 1–32. DOI: https://doi.org/10.2168/LMCS-11(4:7)2015

[36] B. Moszkowski. 1983. *Reasoning About Digital Circuits*. Ph.D. dissertation. Stanford University, CA.

[37] A. Pnueli. 1977. The temporal logic of programs. In *FOCS*. IEEE Computer Society, 46–57. DOI: https://doi.org/10.1109/SFCS.1977.32

[38] I. Pratt-Hartmann. 2005. Temporal prepositions and their logic. *Artificial Intelligence* 166(1–2) (2005), 1–36. DOI: https://doi.org/10.1016/j.artint.2005.04.003

[39] P. Roeper. 1980. Intervals and tenses. *Journal of Philosophical Logic* 9 (1980), 451–469. DOI: https://doi.org/10.1007/BF00262866

[40] A. P. Sistla and E. M. Clarke. 1985. The complexity of propositional linear temporal logics. *Journal of the ACM* 32, 3 (1985), 733–749. DOI: https://doi.org/10.1145/3828.3837

[41] M. Y. Vardi. 1996. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency*. Springer, 238–266. DOI: https://doi.org/10.1007/3-540-60915-6_6

[42] Y. Venema. 1990. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic* 31, 4 (1990), 529–547. DOI: https://doi.org/10.1305/ndjfl/1093635589

[43] T. Wilke. 1999. Classifying discrete temporal properties. In *STACS*, (Lecture Notes in Computer Science, Vol. 1563). Springer, 32–46. DOI: https://doi.org/10.1007/3-540-49116-3_3

Received November 2017; revised July 2018; accepted September 2018