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

Laura Bozzelli[1], Alberto Molinari[2], Angelo Montanari[3], Adriano Peron[4], and Pietro Sala[5]

1 **Technical University of Madrid (UPM), Madrid, Spain**
2 **University of Udine, Udine, Italy**
3 **University of Udine, Udine, Italy**
4 **University of Napoli "Federico II", Napoli, Italy**
5 **University of Verona, Verona, Italy**

Abstract

Model checking is a powerful method widely explored in formal verification to check the (state-transition) model of a system against desired properties of its behaviour. Classically, properties are expressed by formulas of a temporal logic, such as LTL, CTL, and CTL*. These logics are "point-wise" interpreted, as they describe how the system evolves state-by-state. On the contrary, Halpern and Shoham's interval temporal logic (HS) is "interval-wise" interpreted, thus allowing one to naturally express properties of computation stretches, spanning a sequence of states, or properties involving temporal aggregations, which are inherently "interval-based".

In this paper, we study the expressiveness of HS in model checking, in comparison with that of the standard logics LTL, CTL, and CTL*. To this end, we consider HS endowed with three semantic variants: the state-based semantics, introduced by Montanari et al., which allows branching in the past and in the future, the linear-past semantics, allowing branching only in the future, and the linear semantics, disallowing branching. These variants are compared, as for their expressiveness, among themselves and to standard temporal logics, getting a complete picture. In particular, HS with linear (resp., linear-past) semantics is proved to be equivalent to LTL (resp., finitary CTL*).

1998 ACM Subject Classification F.4.1 Mathematical Logic, D.2.4 Software/Program Verification

Keywords and phrases Interval Temporal Logics, Expressiveness, Model Checking

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.26

1 Introduction

Point-based temporal logics (PTLs) provide a fundamental framework for the specification of the behavior of reactive systems, that makes it possible to describe how the 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 modelled 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 temporal logics, such as LTL [23], 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 temporal logics, such as CTL and CTL* [8], 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 computations of a system.

36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016).
Editors: Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen; Article No. 26; pp. 26:1–26:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Interval temporal logics (ITLs) have been proposed as an alternative setting for reasoning about time [10, 22]. Unlike standard PTLs, they take intervals, rather than points, as their primitive entities. ITLs allow one to specify relevant temporal properties that involve, for instance, actions with duration, accomplishments, and temporal aggregations, which are inherently "interval-based", and thus cannot be naturally expressed by PTLs. They have been applied in various areas of computer science, including formal verification, computational linguistics, planning, and multi-agent systems [14, 22, 24]. Halpern and Shoham's modal logic of time intervals HS [10] is the most popular among 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 undecidable for all interesting (classes of) linear orders [10]; the same happens with most of its fragments [7, 13, 17].

In this paper, we focus on the *model checking problem* for HS. In order to check interval properties of computations, one needs to collect information about states into computation stretches (i.e., finite paths of the Kripke structure, *tracks* for short): each track is interpreted as an interval, whose labelling is defined on the basis of the labelling of the component states. This approach to MC has independently and simultaneously been proposed by Molinari et al. in [18] and by Lomuscio and Michaliszyn in [14, 15, 16].

The semantics proposed in [18] is *state-based*, featuring intervals/tracks which are forgetful of the history leading to the starting state of the interval itself. Since the starting state (resp., ending state) of an interval may feature several predecessors (resp., successors), this interpretation induces a branching reference in both future and past. The other relevant choice in this approach concerns the labeling of intervals: a natural principle, known as the *homogeneity assumption*, is adopted, according to which a proposition holds over an interval if and only if it holds over each component state. Under this semantics, the MC problem for full HS turns out to be decidable – it is **EXPSPACE**-hard, while the only known upper bound is non-elementary. The exact complexity of almost all the meaningful syntactic fragments of HS has been recently determined in a series of papers (e.g., [4, 6, 18, 19, 20, 21]).

The approach followed in [14, 15] is more expressive than the one in [18] since it relies on the extension of HS with knowledge modalities typical of the epistemic logics, which allow one to relate distinct paths of a Kripke structure. Additionally, the semantic assumptions differ from those of [18]: the logic is interpreted over the unwinding of the Kripke structure (*computation-tree-based* approach), and the interval labeling takes into account only the endpoints of the interval itself. A more expressive definition of interval labeling, obtained by associating each proposition with a regular expression over the set of states of the Kripke structure, was recently proposed in [16]. The decidability status of MC for full epistemic HS is currently unknown [14, 15].

In this paper, we study the expressiveness of HS, in the context of MC, in comparison with that of the standard PTLs LTL, CTL, and CTL*. The investigation is carried on enforcing the homogeneity assumption. We prove that HS endowed with the state-based semantics proposed in [18] (hereafter denoted as $\mathsf{HS}_{\mathsf{st}}$) is not comparable with LTL, CTL, and CTL*. On the one hand, the result supports the intuition that $\mathsf{HS}_{\mathsf{st}}$ gains some expressiveness by the ability to branch in the past. On the other hand, $\mathsf{HS}_{\mathsf{st}}$ does not feature the possibility to force 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 semantic variants of HS, besides the state-based semantics $\mathsf{HS}_{\mathsf{st}}$, namely, the *computation-tree-based semantics* (denoted as $\mathsf{HS}_{\mathsf{lp}}$) and the *trace-based semantics* ($\mathsf{HS}_{\mathsf{lin}}$).

The state-based and computation-tree-based approaches rely on a branching-time setting and differ in the nature of past. In the latter approach, the past is linear: each interval

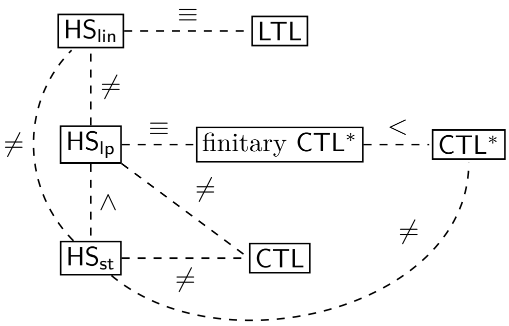


Figure 1 Overview of the expressiveness results.

may have several possible futures, but it has a unique past. Moreover, the past is assumed to be finite and cumulative (i.e., the history of the current situation increases with time, and is never forgotten). The trace-based approach relies on a linear-time setting, 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.

The variant $\mathsf{HS}_{\mathsf{lp}}$ is a natural candidate for an expressiveness comparison with the branching time logics CTL and CTL^*. The more interesting and technically involved result is the characterization of $\mathsf{HS}_{\mathsf{lp}}$, which turns out to be expressively equivalent to finitary CTL^*, i.e., the variant of CTL^* with quantification over finite paths. As for CTL, a non comparability result can be stated. Conversely, $\mathsf{HS}_{\mathsf{lin}}$ is a natural candidate for an expressiveness comparison with LTL. As a matter of fact, we prove that $\mathsf{HS}_{\mathsf{lin}}$ and LTL are equivalent (even for a small fragment of $\mathsf{HS}_{\mathsf{lin}}$). We complete the picture with a comparison of the three semantic variants $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{lp}}$, and $\mathsf{HS}_{\mathsf{lin}}$. We prove that, as expected, $\mathsf{HS}_{\mathsf{lin}}$ is not comparable with either the branching versions, $\mathsf{HS}_{\mathsf{lp}}$ and $\mathsf{HS}_{\mathsf{st}}$. The interesting result is that, on the other hand, $\mathsf{HS}_{\mathsf{lp}}$ is strictly included in $\mathsf{HS}_{\mathsf{st}}$: this supports $\mathsf{HS}_{\mathsf{st}}$, adopted in [18, 19, 20, 21, 4, 6], as a reasonable and adequate semantic choice. The complete picture of the expressiveness results is reported in Figure 1 (the symbols $\neq$, $\equiv$ and $<$ denote incomparability, equivalence, and strict expressiveness inclusion, respectively).

The paper is structured as follows. In Section 2, we introduce some preliminary notions. In Section 3 we prove the expressiveness results. In particular, in Section 3.1 we prove the equivalence between LTL and $\mathsf{HS}_{\mathsf{lin}}$; in Section 3.2 we prove the equivalence between $\mathsf{HS}_{\mathsf{lp}}$ and finitary CTL^*; finally, in Section 3.3 we compare the logics $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{lp}}$, and $\mathsf{HS}_{\mathsf{lin}}$.

2 Preliminaries

Let $(\mathbb{N}, <)$ be the set of natural numbers equipped with the standard linear ordering. For all $i, j \in \mathbb{N}$, with $i \leq j$, $[i, j]$ denotes 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 (we set $|w| = \infty$ 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 $\text{fst}(w) = w(0)$ and $\text{lst}(w) = w(n)$. $\text{Pref}(w) = \{w[0, i] \mid 0 \leq i \leq n - 1\}$ and $\text{Suff}(w) = \{w[i, n] \mid 1 \leq i \leq n\}$ are the sets of all proper prefixes and suffixes of w, respectively.

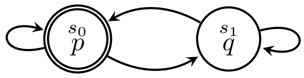


Figure 2 The Kripke structure $\mathcal{K}$.

2.1 Kripke structures and interval structures

▶ **Definition 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, $\mu : S \mapsto 2^{\mathcal{AP}}$ is a total labelling function assigning to each state s the set of propositions 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.

Figure 2 depicts the finite Kripke structure $\mathcal{K} = (\{p, q\}, \{s_0, s_1\}, \delta, \mu, s_0)$, where $\delta = \{(s_i, s_j) \mid i, j = 0, 1\}$, $\mu(s_0) = \{p\}$, and $\mu(s_1) = \{q\}$. 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 *track* (or finite path) of $\mathcal{K}$ is a non-empty prefix 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 tracks of $\mathcal{K}$ and $\mathrm{Trk}^0_{\mathcal{K}}$ be the set of initial tracks of $\mathcal{K}$. For a track ρ, $\mathrm{states}(\rho)$ denotes the set of states occurring in ρ, i.e., $\mathrm{states}(\rho) = \{\rho(0), \cdots, \rho(n)\}$, where $|\rho| = n + 1$.

▶ **Definition 2** (D-tree structure)**.** For a given 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 $\mathcal{C}(\mathcal{K})$, which is obtained by unwinding $\mathcal{K}$ from the initial state. Formally, $\mathcal{C}(\mathcal{K}) = (\mathcal{AP}, \mathrm{Trk}^0_{\mathcal{K}}, \delta', \mu', s_0)$, where the set of nodes is the set of initial tracks of $\mathcal{K}$ and for all $\rho, \rho' \in \mathrm{Trk}^0_{\mathcal{K}}$, $\mu'(\rho) = \mu(\mathrm{lst}(\rho))$ and $(\rho, \rho') \in \delta'$ iff $\rho' = \rho \cdot s$ for some $s \in S$.

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

2.2 Standard temporal logics

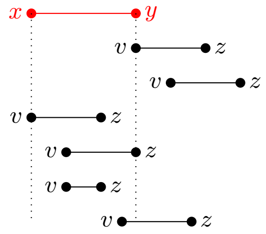
In this subsection we recall the standard propositional temporal logics CTL^*, CTL, and LTL [8, 23]. For a set of proposition letters $\mathcal{AP}$, the formulas φ of CTL^* are defined as follows:

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

where $p \in \mathcal{AP}$, X and U are the "next" and "until" temporal modalities, and $\exists$ is the existential path quantifier. We also use standard shorthands: $\forall\varphi := \neg\exists\neg\varphi$ ("universal path quantifier"), $\mathsf{F}\varphi := \top\mathsf{U}\varphi$ ("eventually") and its dual $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$ ("always"). The logic CTL is the fragment of CTL^* where each temporal modality is immediately preceded by a path quantifier, while LTL corresponds to the fragment of the formulas devoid of path quantifiers.

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



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):

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

We say that $\mathcal{K}$ 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 with the finitary path quantifier $\exists_f$. In this setting, path quantification ranges over the tracks (finite paths) starting from the current state. The satisfaction relation $\rho, i \models \varphi$, where ρ is a track 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 $\mathsf{X}\varphi$, $\rho, i \models \mathsf{X}\varphi$ iff $i + 1 < |\rho|$ and $\rho, i+1 \models \varphi$. A Kripke structure $\mathcal{K}$ is a model of a finitary CTL^* formula if for each initial track ρ of $\mathcal{K}$, it holds that $\mathcal{K}, \rho, 0 \models \varphi$.

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, who introduced the interval temporal logic HS featuring one modality for each Allen relation, but equality [10]. 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 $\{X_1, \ldots, X_n\}$, let $\mathsf{X_1 \cdots X_n}$ be the HS fragment featuring modalities for $X_1, \ldots, X_n$ only.

We assume the *non-strict semantics of* HS, which admits intervals consisting of a single point.[1] Under such an assumption, all HS modalities can be expressed in terms of modalities

[1] All the results we prove in the paper hold for the strict semantics as well.

$\langle \mathrm{B} \rangle$, $\langle \mathrm{E} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, and $\langle \overline{\mathrm{E}} \rangle$ [27], e.g., modality $\langle \mathrm{A} \rangle$ can be expressed in terms of $\langle \mathrm{E} \rangle$ and $\langle \overline{\mathrm{B}} \rangle$ as $\langle \mathrm{A} \rangle \varphi := ([E]\bot \wedge (\varphi \vee \langle \overline{\mathrm{B}} \rangle \varphi)) \vee \langle \mathrm{E} \rangle ([E]\bot \wedge (\varphi \vee \langle \overline{\mathrm{B}} \rangle \varphi))$. We also use the derived operator $\langle \mathrm{G} \rangle$ of HS (and its dual $[G]$), which allows one to select arbitrary subintervals of the given interval and is defined as: $\langle \mathrm{G} \rangle \psi := \psi \vee \langle \mathrm{B} \rangle \psi \vee \langle \mathrm{E} \rangle \psi \vee \langle \mathrm{B} \rangle \langle \mathrm{E} \rangle \psi$.

HS can be viewed as a multi-modal logic with $\langle \mathrm{B} \rangle$, $\langle \mathrm{E} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, and $\langle \overline{\mathrm{E}} \rangle$ 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 4** (Abstract interval model [18]). 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 $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ 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, and $B_{\mathbb{I}}$ and $E_{\mathbb{I}}$ as the Allen's relations B (*started-by*) and E (*finished-by*), respectively; 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 (Boolean connectives are treated as usual):

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

▶ **Definition 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 variants of HS semantics for model checking

In this section, we define the three variants of HS semantics $\mathsf{HS}_{\mathsf{st}}$ (state-based semantics), $\mathsf{HS}_{\mathsf{lp}}$ (computation-tree-based semantics), and $\mathsf{HS}_{\mathsf{lin}}$ (trace-based semantics) for model checking HS against Kripke structures. For each such variant $\mathcal{S}$, the related (finite) model checking problem is deciding whether a finite Kripke structure is a model of an HS formula under $\mathcal{S}$.

Let us start with the *state-based semantics* [18], 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 tracks of $\mathcal{K}$.

▶ **Definition 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 $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ 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 $\rho = v_1 \cdots v_n$ if and only if it holds over all the states $v_1, \ldots, v_n$ of ρ. This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval if and only if it holds over all its subintervals [25].

▶ **Definition 7** (State-based semantics). Let $\mathcal{K}$ be a Kripke structure and ψ be an HS formula. A track $\rho \in \mathrm{Trk}_{\mathcal{K}}$ satisfies ψ under the state-based semantics, 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 semantics*, denoted as $\mathcal{K} \models_{\mathsf{st}} \psi$, if for all *initial* tracks $\rho \in \mathrm{Trk}^0_{\mathcal{K}}$, it holds that $\mathcal{K}, \rho \models_{\mathsf{st}} \psi$.

We now introduce the *computation-tree-based semantics*, 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 semantic variant induces a linear reference in the past.

▶ **Definition 8** (Computation-tree-based semantics). A Kripke structure $\mathcal{K}$ *is a model of an* HS *formula* ψ *under the computation-tree-based semantics*, written $\mathcal{K} \models_{\mathsf{lp}} \psi$, if $\mathcal{C}(\mathcal{K}) \models_{\mathsf{st}} \psi$.

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

▶ **Definition 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 $\mathcal{IS}_{\mathcal{K},\pi} = ((\mathbb{N}, <), \sigma)$, where for each interval $[i, j]$, $\sigma([i, j]) = \bigcap_{h=i}^{j} \mu(\pi(h))$.

▶ **Definition 10** (Trace-based semantics). A Kripke structure $\mathcal{K}$ *is a model of an* HS *formula* ψ *under the trace-based semantics*, denoted as $\mathcal{K} \models_{\mathsf{lin}} \psi$, iff for each initial infinite path π and for each initial interval $[0, i]$, it holds that $\mathcal{IS}_{\mathcal{K},\pi}, [0, i] \models \psi$.

3 Expressiveness

In this section, we compare the expressive power of the logics $\mathsf{HS_{st}}$, $\mathsf{HS_{lp}}$, $\mathsf{HS_{lin}}$, 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. When comparing the expressive power of two logics L_1 and 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 expressiveness*), 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 *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 $L_2 \not\geq L_1$.

3.1 Equivalence between LTL and $\mathsf{HS_{lin}}$

In this section we show that $\mathsf{HS_{lin}}$ is as expressive as LTL even for small syntactical fragments of $\mathsf{HS_{lin}}$. For this purpose, we exploit the well-known equivalence between LTL and First Order Logic (FO) over infinite words. Recall that given a countable set $\{x, y, z, \ldots\}$ of (position) variables, FO formulas φ over a set of proposition symbols $\mathcal{AP} = \{p, \ldots\}$ are defined as:

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

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 $\mu(\pi(0))\mu(\pi(1))\cdots$ (for the details, see [5]). 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 [11].

▶ **Proposition 11.** *Given a* 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* $\pi \models \varphi$ *iff* $\pi, 0 \models \psi$*.*

Given a $\mathsf{HS_{lin}}$ formula ψ, we construct a FO sentence ψ_{FO} such that, for all Kripke structures $\mathcal{K}$, $\mathcal{K} \models_{\mathsf{lin}} \psi$ iff for each initial infinite path π of $\mathcal{K}$, $\pi \models \psi_{\mathsf{FO}}$. The formula ψ_{FO} is given by $\exists x((\forall z.z \geq x) \wedge \forall y.h(\psi, x, y))$, where $h(\psi, x, y)$ is a FO formula having x and y as free variables (intuitively, representing the endpoints of the current interval) and ensuring that for each infinite path π and interval $[i, j]$, $\mathcal{IS}_{\mathcal{K},\pi}, [i, j] \models \psi$ iff $(\pi, g) \models h(\psi, x, y)$ for any valuation g such that $g(x) = i$ and $g(y) = j$. The construction of $h(\psi, x, y)$ is straightforward (for the details, see the report [5]). Thus, by Proposition 11, we obtain the following result.

▶ **Theorem 12.** $\mathsf{LTL} \geq \mathsf{HS_{lin}}$.

Conversely, we show that LTL can be translated in linear-time into $\mathsf{HS_{lin}}$ (actually, the fragment AB, featuring only modalities for A and B, is expressive enough for the purpose). In the following we will make use of the B formula $length_n$, with $n \geq 1$, characterizing the intervals of length n, which is defined as follows: $length_n := (\underbrace{\langle \mathrm{B} \rangle \ldots \langle \mathrm{B} \rangle}_{n-1} \top) \ \wedge \ (\underbrace{[B] \ldots [B]}_{n} \bot)$.

▶ **Theorem 13.** *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* $\mathsf{AB_{lin}}$*.*

Proof. Let $f : \mathsf{LTL} \mapsto \mathsf{AB}$ be the mapping homomorphic w. r. to the Boolean connectives, defined as follows for each proposition p and for the temporal modalities X and U:

$$f(p) = p, \qquad f(\mathsf{X}\psi) = \langle \mathrm{A} \rangle (length_2 \wedge \langle \mathrm{A} \rangle (length_1 \wedge f(\psi))),$$
$$f(\psi_1 \mathsf{U} \psi_2) = \langle \mathrm{A} \rangle \Big(\langle \mathrm{A} \rangle (length_1 \wedge f(\psi_2)) \wedge [B](\langle \mathrm{A} \rangle (length_1 \wedge f(\psi_1)) \Big).$$

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 $\pi, i \models \psi$ iff $\mathcal{IS}_{\mathcal{K},\pi}, [i, i] \models f(\psi)$. Hence $\mathcal{K} \models \psi$ iff $\mathcal{K} \models_{\mathsf{lin}} length_1 \rightarrow f(\psi)$. ◀

▶ **Corollary 14.** $\mathsf{HS_{lin}}$ *and* LTL *have the same expressiveness.*

3.2 A characterization of $\mathsf{HS_{lp}}$

In this section we show that $\mathsf{HS_{lp}}$ is as expressive as *finitary* CTL^*. Actually, the result can be proved to hold already for the syntactical fragment ABE (which does not feature transposed modalities). In addition, we show that $\mathsf{HS_{lp}}$ is subsumed by CTL^*.

We first show that finitary CTL^* is subsumed by $\mathsf{HS_{lp}}$. The result is proved by exploiting a preliminary property stating that, when interpreted over finite words, the BE fragment of HS and LTL define the same class of finitary languages. For an LTL formula φ with proposition symbols over an alphabet Σ (in our case Σ is $2^{\mathcal{AP}}$), $L_{act}(\varphi)$ denotes the set of non-empty finite words over Σ satisfying φ under the standard action-based semantics of LTL, interpreted over finite words (see [26]). A similar notion can be given for BE formulas φ with propositional symbols in Σ (considered under the homogeneity principle). Then φ denotes a language, written $L_{act}(\varphi)$, of non-empty finite words over Σ, inductively defined as:

- $L_{act}(a) = a^+$ for each $a \in \Sigma$;
- $L_{act}(\neg\varphi) = \Sigma^+ \setminus L_{act}(\varphi)$;
- $L_{act}(\varphi_1 \wedge \varphi_2) = L_{act}(\varphi_1) \cap L_{act}(\varphi_2)$;
- $L_{act}(\langle \mathrm{B} \rangle \varphi) = \{w \in \Sigma^+ \mid \mathrm{Pref}(w) \cap L_{act}(\varphi) \neq \emptyset\}$;
- $L_{act}(\langle \mathrm{E} \rangle \varphi) = \{w \in \Sigma^+ \mid \mathrm{Suff}(w) \cap L_{act}(\varphi) \neq \emptyset\}$.

We prove that under the action-based semantics, BE formulas and LTL formulas define the same class of finitary languages. By proceeding as in Section 3.1, one can easily show that, over finite words, the class of languages defined by the fragment BE is subsumed by that defined by LTL. To prove the converse direction we exploit an algebraic condition introduced in [28], here 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.

▶ **Definition 15** (LTL-closure). A class $\mathcal{C}$ of languages of finite words over finite alphabets is LTL-*closed* iff the following conditions are satisfied, where Σ and Δ are finite alphabets, $b \in \Sigma$ and $\Gamma = \Sigma \setminus \{b\}$:

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

▶ **Theorem 16** ([28]). *Any* LTL*-closed class* $\mathcal{C}$ *of finitary languages includes the class of* LTL*-definable finitary languages.*

▶ **Theorem 17.** *Let* φ *be an* LTL *formula over a finite alphabet* Σ*. Then there exists a* BE *formula* φ_{HS} *over* Σ *such that* $L_{act}(\varphi_{\mathsf{HS}}) = L_{act}(\varphi)$*.*

Proof. It suffices to prove that the class of finitary languages definable by BE formulas is LTL-closed, and to apply Theorem 16 (the proof of LTL-closure is reported in [5]). ◀

By exploiting Theorem 17, we establish the following result.

▶ **Theorem 18.** *Let* φ *be a finitary* CTL* *formula over* $\mathcal{AP}$*. Then there is an* ABE *formula* φ_{HS} *over* $\mathcal{AP}$ *s.t. for all Kripke structures* $\mathcal{K}$ *over* $\mathcal{AP}$ *and tracks* ρ*,* $\mathcal{K}, \rho, 0 \models \varphi$ *iff* $\mathcal{K}, \rho \models_{\mathsf{st}} \varphi_{\mathsf{HS}}$*.*

Proof. The proof is by induction on the nesting depth of modality $\exists_f$ in φ. The base case (φ is a finitary LTL formula over $\mathcal{AP}$) is similar to the inductive step, thus we can focus our attention on the latter. Let H be the non-empty set of subformulas of φ of the form $\exists_f \psi$ which do not occur in the scope of the path quantifier $\exists_f$. Then φ can be seen as an LTL formula over the set of atomic propositions $\overline{\mathcal{AP}} = \mathcal{AP} \cup H$. Let $\Sigma = 2^{\overline{\mathcal{AP}}}$ and $\overline{\varphi}$ be the LTL formula over Σ obtained from φ by replacing each occurrence of $p \in \overline{\mathcal{AP}}$ 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 track 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 show by structural induction on φ that:
Claim 1: $\mathcal{K}, \rho, 0 \models \varphi$ iff $\rho_H \in L_{act}(\overline{\varphi})$.

By Theorem 17, there exists a BE formula $\overline{\varphi}_{\mathsf{HS}}$ over Σ such that $L_{act}(\overline{\varphi}) = L_{act}(\overline{\varphi}_{\mathsf{HS}})$. Moreover, 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 tracks ρ of $\mathcal{K}$, $\mathcal{K}, \rho, 0 \models \psi$ iff $\mathcal{K}, \rho \models_{\mathsf{st}} \psi_{\mathsf{HS}}$. Since ρ is arbitrary, $\mathcal{K}, \rho, i \models \exists_f \psi$ iff $\mathcal{K}, \rho[i,i], 0 \models \exists_f \psi$ iff $\mathcal{K}, \rho[i,i] \models_{\mathsf{st}} \langle \mathrm{A} \rangle\, \psi_{\mathsf{HS}}$, for each $i \geq 0$. Let φ_{HS} be the ABE formula over $\mathcal{AP}$ obtained from the BE formula $\overline{\varphi}_{\mathsf{HS}}$ by replacing each occurrence of $P \in \Sigma$ in $\overline{\varphi}_{\mathsf{HS}}$ with the formula

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

Since for all $i \geq 0$ and $\exists_f \psi \in H$, $\mathcal{K}, \rho, i \models \exists_f \psi$ iff $\mathcal{K}, \rho[i,i] \models_{\mathsf{st}} \langle \mathrm{A} \rangle\, \psi_{\mathsf{HS}}$, by a straightforward induction on the structure of $\overline{\varphi}_{\mathsf{HS}}$, for all Kripke structures $\mathcal{K}$ and tracks ρ of $\mathcal{K}$ we have

$\mathcal{K}, \rho \models_{\mathsf{st}} \varphi_{\mathsf{HS}}$ iff $\rho_H \in L_{act}(\overline{\varphi}_{\mathsf{HS}})$. Therefore, since $L_{act}(\overline{\varphi}) = L_{act}(\overline{\varphi}_{\mathsf{HS}})$, by Claim 1 $\mathcal{K}, \rho, 0 \models \varphi$ iff $\mathcal{K}, \rho \models_{\mathsf{st}} \varphi_{\mathsf{HS}}$, for arbitrary Kripke structures $\mathcal{K}$ and tracks ρ of $\mathcal{K}$. ◀

Since for the fragment ABE of HS the computation-tree-based semantics coincides with the state-based semantics, by Theorem 18 we obtain the following corollary.

▶ **Corollary 19.** *Finitary* CTL* *is subsumed by both* $\mathsf{HS}_{\mathsf{st}}$ *and* $\mathsf{HS}_{\mathsf{lp}}$.

Conversely, we show now that $\mathsf{HS}_{\mathsf{lp}}$ is subsumed by both CTL* and the finitary variant of CTL*. For this purpose, we first introduce a hybrid and linear-past extension of CTL*, called *hybrid* CTL^*_{lp}, and its finitary variant, called *finitary hybrid* CTL^*_{lp}. Hybrid logics (see [3]), besides standard modalities, make use of explicit variables and quantifiers that bind them. 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 $\mathsf{HS}_{\mathsf{lp}}$. Actually, we will show that the restricted form of 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 by 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* [3], which binds x to the current position along the given initial infinite path. We also use the standard shorthands $\mathsf{F}^-\varphi := \top\mathsf{U}^-\varphi$ ("eventually in the past") and its dual $\mathsf{G}^-\varphi := \neg\mathsf{F}^-\neg\varphi$ ("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 assigning 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 the Boolean connectives and for U and X):

$\pi, g, i \models \mathsf{X}^-\varphi \quad \Leftrightarrow i > 0$ and $\pi, g, i-1 \models \varphi$,
$\pi, g, i \models \varphi_1\mathsf{U}^-\varphi_2 \quad \Leftrightarrow$ for some $j \leq i : \pi, g, j \models \varphi_2$ and $\pi, g, k \models \varphi_1$ for all $j < k \leq i$,
$\pi, g, i \models \exists\varphi \quad \Leftrightarrow$ for some initial infinite path π' s.t. $\pi'[0,i] = \pi[0,i]$, $\pi', g, i \models \varphi$,
$\pi, g, i \models x \quad \Leftrightarrow g(x) = i$,
$\pi, g, i \models {\downarrow}x.\varphi \quad \Leftrightarrow \pi, g[x \leftarrow i], i \models \varphi$,

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 for each initial infinite path π, $\pi, g_0, 0 \models \varphi$, where g_0 assigns 0 to each variable. Note that path quantification is "memoryful", i.e., 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 and equally expressive extension of CTL* [12]. 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 (tracks) 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: (1) *each subformula* $\exists\psi$ *of* φ *has at most one free* *variable;* (2) *each subformula* $\exists\psi(x)$ *of* φ *having* x *as free variable occurs in* φ *in the context* $(\mathsf{F}^{-}x) \wedge \exists\psi(x)$. Intuitively, for each state subformula $\exists\psi$, the unique free variable (if any) refers to ancestors of the current node in the computation tree. The notion of well-formed formula of finitary hybrid CTL^*_{lp} is similar: the path quantifier $\exists$ is replaced by its finitary version $\exists_f$. 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 well-known non-regular context-free branching requirement (see, e.g., [2]).

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

▶ **Proposition 20.** *Given a* $\mathsf{HS}_{\mathsf{lp}}$ *formula* φ*, one can construct in linear-time an equivalent well-formed sentence of hybrid* CTL^*_{lp} *(resp., finitary hybrid* CTL^*_{lp}*).*

Proof. We focus on the translation from $\mathsf{HS}_{\mathsf{lp}}$ into the well-formed fragment of hybrid CTL^*_{lp}. The translation from $\mathsf{HS}_{\mathsf{lp}}$ into the well-formed fragment of finitary hybrid CTL^*_{lp} is similar. Let φ be a $\mathsf{HS}_{\mathsf{lp}}$ formula. The desired hybrid CTL^*_{lp} sentence is given by ${\downarrow}x.\mathsf{G}\, f(\varphi, x)$, where the mapping $f(\varphi, x)$ is homomorphic with respect to the Boolean connectives, and is defined for the atomic propositions and the other modalities as follows (y is a fresh variable):

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

Clearly ${\downarrow}x.\mathsf{G}\, f(\varphi, x)$ is well-formed. Moreover, 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 $\mathcal{C}(\mathcal{K}), \mathcal{C}(\pi, h, i) \models_{\mathsf{st}} \varphi$, where $\mathcal{C}(\pi, h, i)$ denotes the track of the computation tree $\mathcal{C}(\mathcal{K})$ starting from $\pi[0, h]$ and leading to $\pi[0, i]$. Hence $\mathcal{K}$ is a model of ${\downarrow}x.\mathsf{G}\, f(\varphi, x)$ if for each initial track ρ of $\mathcal{C}(\mathcal{K})$ we have $\mathcal{C}(\mathcal{K}), \rho \models_{\mathsf{st}} \varphi$. ◀

Let LTL_p be the past extension of LTL, obtained by adding the past modalities X^{-} and U^{-}. By exploiting the well-formedness requirement and the well-known separation theorem for LTL_p over finite and infinite words [9] (i.e., any LTL_p formula can be effectively converted into an equivalent Boolean combination of LTL formulas and pure past LTL_p formulas), and proceeding by induction on the nesting depth of path quantifiers, we establish the following result (the proof can be found in [5]).

▶ **Proposition 21.** *The set of well-formed sentences of hybrid* CTL^*_{lp} *(resp., finitary hybrid* CTL^*_{lp}*) has the same expressiveness as* CTL^* *(resp., finitary* CTL^**).*

By Corollary 19, and Propositions 20 and 21, we obtain the main result of Section 3.2.

▶ **Theorem 22.** $\mathsf{CTL}^* \geq \mathsf{HS}_{\mathsf{lp}}$. *Moreover,* $\mathsf{HS}_{\mathsf{lp}}$ *is as expressive as finitary* CTL^**.*

3.3 Expressiveness comparison of $\mathsf{HS}_{\mathsf{lin}}$, $\mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{lp}}$

We first show that $\mathsf{HS}_{\mathsf{st}}$ is *not* subsumed by $\mathsf{HS}_{\mathsf{lp}}$. As a matter of fact we show that $\mathsf{HS}_{\mathsf{st}}$ is sensitive to unwinding, allowing us to discriminate finite Kripke structures having

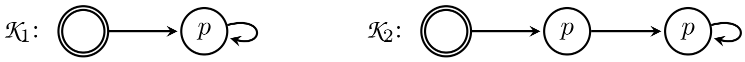


Figure 3 The Kripke structures $\mathcal{K}_1$ and $\mathcal{K}_2$.

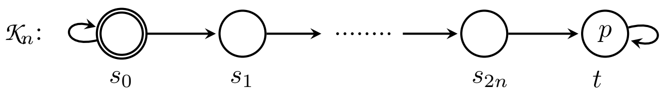


Figure 4 The Kripke structure $\mathcal{K}_n$ with $n \geq 1$.

the same computation tree (whereas they are indistinguishable by $\mathsf{HS}_{\mathsf{lp}}$). In particular, let us consider the two finite Kripke structures $\mathcal{K}_1$ and $\mathcal{K}_2$ of Figure 3. 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 $\mathcal{K}_2$, i.e., $\mathcal{K}_1 \models_{\mathsf{lp}} \varphi$ iff $\mathcal{K}_2 \models_{\mathsf{lp}} \varphi$. 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 $\psi := \langle E\rangle(p \wedge length_1) \to \langle E\rangle(length_1 \wedge \langle \overline{A}\rangle(p \wedge \neg length_1))$. Clearly $\mathcal{K}_1 \models_{\mathsf{st}} \psi$ but $\mathcal{K}_2 \not\models_{\mathsf{st}} \psi$. Hence we obtain the following result.

▶ **Proposition 23.** $\mathsf{HS}_{\mathsf{lp}} \not\geq \mathsf{HS}_{\mathsf{st}}$.

Since $\mathsf{HS}_{\mathsf{lp}}$ and finitary CTL^* have the same expressiveness (Theorem 22) and finitary CTL^* is subsumed by $\mathsf{HS}_{\mathsf{st}}$ (Corollary 19), by Proposition 23 we obtain the following result.

▶ **Corollary 24.** $\mathsf{HS}_{\mathsf{st}}$ *is more expressive than* $\mathsf{HS}_{\mathsf{lp}}$.

Let us now 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-expressible. Thus, by Corollary 14, there is no equivalent HS formula under the trace-based semantics. On the other hand, the requirement $\forall \mathsf{G} \exists \mathsf{F} p$ can be expressed under the state-based (resp., computation-tree-based) semantics by the HS formula $\langle \overline{B}\rangle\langle E\rangle p$.

▶ **Proposition 25.** $\mathsf{HS}_{\mathsf{lin}} \not\geq \mathsf{HS}_{\mathsf{st}}$ *and* $\mathsf{HS}_{\mathsf{lin}} \not\geq \mathsf{HS}_{\mathsf{lp}}$.

Next we show that $\mathsf{HS}_{\mathsf{lin}} \not\leq \mathsf{HS}_{\mathsf{st}}$ and $\mathsf{HS}_{\mathsf{lin}} \not\leq \mathsf{HS}_{\mathsf{lp}}$. To this end we establish the following.

▶ **Proposition 26.** *The* LTL *formula* $\mathsf{F} p$ *(equivalent to the* CTL *formula* $\forall \mathsf{F} p$*) cannot be expressed in either* $\mathsf{HS}_{\mathsf{lp}}$ *or* $\mathsf{HS}_{\mathsf{st}}$.

We prove Proposition 26 for the state-based semantics (for the computation-tree-based semantics the proof is similar). We exhibit 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. Hence the result follows. Fix $n \geq 1$. The Kripke structure $\mathcal{K}_n$ is given in Figure 4. The Kripke structure $\mathcal{M}_n$ is obtained from $\mathcal{K}_n$ by setting as its initial state s_1 instead of s_0. Formally, $\mathcal{K}_n = (\{p\}, S_n, \delta_n, \mu_n, s_0)$ and $\mathcal{M}_n = (\{p\}, S_n, \delta_n, \mu_n, s_1)$, where $S_n = \{s_0, s_1, \ldots, s_{2n}, t\}$, $\delta_n = \{(s_0, s_0), (s_0, s_1), \ldots, (s_{2n-1}, s_{2n}), (s_{2n}, t), (t, t)\}$, $\mu(s_i) = \emptyset$ for all $0 \leq i \leq 2n$, and $\mu(t) = \{p\}$. Clearly $\mathcal{K}_n \not\models \mathsf{F}p$ and $\mathcal{M}_n \models \mathsf{F}p$.

We say that a HS formula ψ is *balanced* if, for each subformula $\langle \mathrm{B}\rangle\, \theta$ (resp., $\langle \overline{\mathrm{B}}\rangle\, \theta$), θ is of the form $\theta_1 \wedge \theta_2$ with $|\theta_1| = |\theta_2|$. By using conjunctions of $\top$, one can trivially convert a HS formula ψ into a balanced HS formula which is equivalent to ψ under any of the considered HS semantic variants. Lemma 27 is proved in [5]: by such a lemma and the fact that, for each $n \geq 1$, $\mathcal{K}_n \not\models \mathsf{F}p$ and $\mathcal{M}_n \models \mathsf{F}p$, we get a proof of Proposition 26.

▶ **Lemma 27.** *For all $n \geq 1$ and balanced* HS *formulas ψ s.t. $|\psi| \leq n$, $\mathcal{K}_n \models_{\mathsf{st}} \psi$ iff $\mathcal{M}_n \models_{\mathsf{st}} \psi$.*

By Propositions 25–26, we obtain the following result.

▶ **Corollary 28.** $\mathsf{HS}_{\mathsf{lin}}$ *and* $\mathsf{HS}_{\mathsf{st}}$ *(resp.,* $\mathsf{HS}_{\mathsf{lp}}$*) are expressively incomparable.*

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

▶ **Corollary 29.**

1. $\mathsf{HS}_{\mathsf{st}}$ *and* CTL* *(resp.,* CTL*) are expressively incomparable;*
2. $\mathsf{HS}_{\mathsf{lp}}$ *and finitary* CTL* *are less expressive than* CTL**;*
3. $\mathsf{HS}_{\mathsf{lp}}$ *and* CTL *are expressively incomparable.*

Proof. (Point 1) By Proposition 26 and the fact that CTL* is not sensitive to unwinding.
(Point 2) By Theorem 22, $\mathsf{HS}_{\mathsf{lp}}$ is subsumed by CTL*, and $\mathsf{HS}_{\mathsf{lp}}$ and finitary CTL* have the same expressiveness. Hence, by Proposition 26, the result follows.
(Point 3) By Proposition 26, it suffices to show that there exists a $\mathsf{HS}_{\mathsf{lp}}$ 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 [8] 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 $\mathsf{HS}_{\mathsf{lp}}$ and finitary CTL* have the same expressiveness (Theorem 22), the result follows. ◀

4 Conclusions and future work

In this paper, we have studied three semantic variants, namely, $\mathsf{HS}_{\mathsf{st}}$, $\mathsf{HS}_{\mathsf{lp}}$, and $\mathsf{HS}_{\mathsf{lin}}$, of the interval temporal logic HS, comparing their expressiveness to that of the standard temporal logics LTL, CTL, finitary CTL*, and CTL*. The reported results imply the decidability of the model checking problem for $\mathsf{HS}_{\mathsf{lp}}$ and $\mathsf{HS}_{\mathsf{lin}}$; the related complexity issues will be studied in the future work. Moreover, we shall investigate how the expressiveness changes when the homogeneity assumption is relaxed.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.
2. R. Alur, P. Cerný, and S. Zdancewic. Preserving secrecy under refinement. In *ICALP*, LNCS 4052, pages 107–118, 2006. doi:10.1007/11787006_10.
3. P. Blackburn and J. Seligman. What are hybrid languages? In *AiML*, pages 41–62, 1998.
4. L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Interval Temporal Logic Model Checking: the Border Between Good and Bad HS Fragments. In *IJCAR*, LNAI 9706, pages 389–405, 2016. doi:10.1007/978-3-319-40229-1_27.
5. L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Interval vs. point temporal logic model checking: an expressiveness comparison. Technical report, Univ. of Udine, Italy, 2016. URL: http://www.uniud.it/it/ateneo-uniud/ateneo-uniud-organizzazione/dipartimenti/dima/ricerca/pubblicazioni/preprints/3.2016.
6. L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Model Checking the Logic of Allen's Relations Meets and Started-by is $\mathbf{P}^{\mathbf{NP}}$-Complete. In *GandALF*, pages 76–90, 2016. doi:10.4204/EPTCS.226.6.

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

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

9 D. M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, LNCS 398, pages 409–448, 1987. doi:10.1007/3-540-51803-7_36.

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

11 H. Kamp. *Tense Logic and the Theory of Linear Order.* PhD thesis, Ucla, 1968.

12 O. Kupferman, A. Pnueli, and M. Y. Vardi. Once and for all. *J. Comput. Syst. Sci.*, 78(3):981–996, 2012. doi:10.1016/j.jcss.2011.08.006.

13 K. Lodaya. Sharpening the undecidability of interval temporal logic. In *ASIAN*, LNCS 1961, pages 290–298, 2000. doi:10.1007/3-540-44464-5_21.

14 A. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *IJCAI*, pages 1010–1016, 2013.

15 A. Lomuscio and J. Michaliszyn. Decidability of model checking multi-agent systems against a class of EHS specifications. In *ECAI*, pages 543–548, 2014. doi:10.3233/978-1-61499-419-0-543.

16 A. Lomuscio and J. Michaliszyn. Model checking multi-agent systems against epistemic HS specifications with regular expressions. In *KR*, pages 298–308, 2016.

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

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

19 A. Molinari, A. Montanari, and A. Peron. Complexity of ITL model checking: some well-behaved fragments of the interval logic HS. In *TIME*, pages 90–100, 2015. doi:10.1109/TIME.2015.12.

20 A. Molinari, A. Montanari, and A. Peron. A model checking procedure for interval temporal logics based on track representatives. In *CSL*, pages 193–210, 2015. doi:10.4230/LIPIcs.CSL.2015.193.

21 A. Molinari, A. Montanari, A. Peron, and P. Sala. Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture. In *KR*, pages 473–483, 2016.

22 B. Moszkowski. *Reasoning About Digital Circuits.* PhD thesis, Stanford University, CA, USA, 1983.

23 A. Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977. doi:10.1109/SFCS.1977.32.

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

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

26 M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for concurrency*, pages 238–266. Springer, 1996. doi:10.1007/3-540-60915-6_6.

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

28 T. Wilke. Classifying discrete temporal properties. In *STACS*, LNCS 1563, pages 32–46, 1999. doi:10.1007/3-540-49116-3_3.