Expressiveness and Decidability of Temporal Logics for Asynchronous Hyperproperties

Laura Bozzelli
University of Napoli "Federico II", Italy

Adriano Peron
University of Napoli "Federico II", Italy

César Sánchez
IMDEA Software Institute, Madrid, Spain

Abstract

Hyperproperties are properties of systems that relate different executions traces, with many applications from security to symmetry, consistency models of concurrency, etc. In recent years, different linear-time logics for specifying *asynchronous* hyperproperties have been investigated. Though model checking of these logics is undecidable, useful decidable fragments have been identified with applications e.g. for asynchronous security analysis. In this paper, we address expressiveness and decidability issues of temporal logics for asynchronous hyperproperties. We compare the expressiveness of these logics together with the extension S1S[E] of S1S with the equal-level predicate by obtaining an almost complete expressiveness picture. We also study the expressive power of these logics when interpreted on singleton sets of traces. We show that for two asynchronous extensions of HyperLTL, checking the existence of a singleton model is already undecidable, and for one of them, namely Context HyperLTL ($\mathsf{HyperLTL}_C$), we establish a characterization of the singleton models in terms of the extension of standard FO[<] over traces with addition. This last result generalizes the well-known equivalence between FO[<] and LTL. Finally, we identify new boundaries on the decidability of model checking $\mathsf{HyperLTL}_C$.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Hyperproperties, Asynchronous hyperproperties, Temporal logics for hyperproperties, Expressiveness, Decidability, Model checking

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.27

Related Version *Full Version*: https://arxiv.org/abs/2207.02956 [7]

Funding *César Sánchez*: funded in part by the Madrid Regional Government under project "S2018/TCS-4339 (BLOQUES-CM)" and by a research grant from Nomadic Labs and the Tezos Foundation.

1 Introduction

Hyperproperties. In the last decade, a novel specification paradigm has been introduced that generalizes traditional trace properties by properties of sets of traces, the so called *hyperproperties* [9]. Hyperproperties relate execution traces of a reactive system and are useful in many settings. In the area of information flow control, hyperproperties can formalize security policies (like noninterference [18, 26] and observational determinism [32]) which compare observations made by an external low-security agent along traces resulting from different values of not directly observable inputs. These security requirements go, in general, beyond regular properties and cannot be expressed in classical regular temporal logics such as LTL [27], CTL, and CTL* [12]. Hyperproperties also have applications in other settings, such as the symmetric access to critical resources in distributed protocols [15], consistency models in concurrent computing [4], and distributed synthesis [14].

33rd International Conference on Concurrency Theory (CONCUR 2022).
Editors: Bartek Klin, Sławomir Lasota, and Anca Muscholl; Article No. 27; pp. 27:1–27:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In the context of model checking of finite-state reactive systems, many temporal logics for hyperproperties have been proposed [11, 8, 5, 28, 13, 10, 21] for which model checking is decidable, including HyperLTL [8], HyperCTL* [8], HyperQPTL [28, 10], and HyperPDL-Δ [21] which extend LTL, CTL*, QPTL [29], and PDL [17], respectively, by explicit first-order quantification over traces and trace variables to refer to multiple traces at the same time. The semantics of all these logics is *synchronous* and the temporal modalities are evaluated by a lockstepwise traversal of all the traces assigned to the quantified trace variables.

A different approach for the formalization of synchronous hyper logics is based on hyper variants of monadic second-order logic over traces or trees [10]. For the linear-time setting, we recall the logic S1S[E] [10] (and its first-order fragment FO[<,E] [16]) which syntactically extends monadic second-order logic of one successor S1S with the *equal-level predicate E*, which relate the same time points on different traces. Another class of hyperlogics is obtained by adopting a *team semantics* for standard temporal logics, in particular, LTL [24, 25, 31, 19].

Asynchronous extensions of Hyper logics. Many application domains require asynchronous properties that relate traces at distinct time points which can be arbitrarily far from each other. For example, asynchronous specifications are needed to reason about a multithreaded environment in which threads are not scheduled in lockstep, and traces associated with distinct threads progress at different speed. Asynchronous hyperproperties are also useful in information-flow security where an observer is not time-sensitive, so the observer cannot distinguish consecutive time points along an execution having the same observation. This again requires asynchronously matching sequences of observations along distinct execution traces. A first systematic study of asynchronous hyperproperties is done in [22], where two powerful and expressively equivalent linear-time asynchronous formalisms are introduced: the temporal fixpoint calculus H_μ and an automata-theoretic formalism where the quantifier-free part of a specification is expressed by the class of parity multi-tape Alternating Asynchronous Word Automata (AAWA) [22]. While the expressive power of the quantifier-part of HyperLTL is just that of LTL over tuples of traces of fixed arity (*multi-traces*), AAWA allow to specify very expressive non-regular multi-trace properties. Model checking against H_μ or its AAWA-based counterpart is undecidable even for the quantifier alternation-free fragment. In [22], two decidable subclasses of parity AAWA are identified which express only multi-trace ω-regular properties and lead to two H_μ fragments with decidable model checking. More recently, other temporal logics [2, 6] which syntactically extend HyperLTL have been introduced for expressing asynchronous hyperproperties. *Asynchronous HyperLTL* (A-HyperLTL) [2], useful for asynchronous security analysis, models asynchronicity by means of an additional quantification layer over the so called *trajectories*. Intuitively, a trajectory controls the relative speed at which traces progress by choosing at each instant which traces move and which traces stutter. The general logic also has an undecidable model-checking problem, but [2] identifies practical decidable fragments, and reports an empirical evaluation. *Stuttering HyperLTL* (HyperLTL$_S$) and *Context HyperLTL* (HyperLTL$_C$) are introduced in [6] as more expressive asynchronous extensions of HyperLTL. HyperLTL$_S$ uses relativized versions of the temporal modalities with respect to finite sets Γ of LTL formulas. Intuitively, these modalities are evaluated by a lockstepwise traversal of the sub-traces of the given traces which are obtained by removing “redundant” positions with respect to the pointwise evaluation of the LTL formulas in Γ. HyperLTL$_C$ extends HyperLTL by unary modalities $\langle C \rangle$ parameterized by a non-empty subset C of trace variables – called the *context* – which restrict the evaluation of the temporal modalities to the traces associated with the variables in C. Both HyperLTL$_S$ and HyperLTL$_C$ are subsumed by H_μ and still have an undecidable model-checking problem, and fragments of the two logics with a decidable model-checking have been investigated [6].

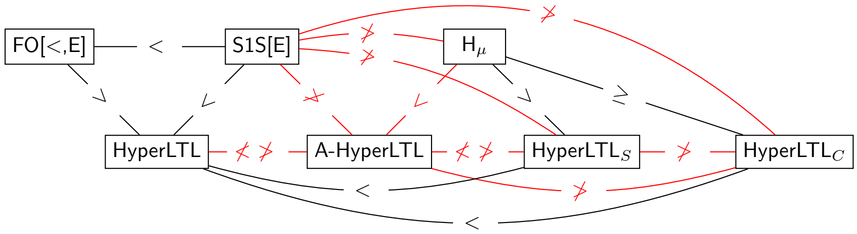

Figure 1 Expressiveness comparisons between linear-time hyper logics.

Our contribution. In this paper, we study expressiveness and decidability of asynchronous extensions of HyperLTL [22, 2, 6]. Our main goal is to compare the expressive power of these logics together with the known logics for linear-time hyperproperties based on the equal-level predicate whose most powerful representative is S1S[E]. The first-order fragment FO[<,E] of S1S[E] is already strictly more expressive than HyperLTL [16] and, unlike S1S[E], its model-checking problem is decidable [10]. We obtain an almost complete expressiveness picture, summarized in Figure 1, where novel results are annotated in red. In particular, for A-HyperLTL, we show that although HyperLTL and A-HyperLTL are expressively incomparable, HyperLTL can be embedded into A-HyperLTL using a natural encoding. We also establish that A-HyperLTL is strictly less expressive than H$_\mu$ and its AAWA counterpart. For the relative expressiveness of A-HyperLTL, HyperLTL$_S$, and HyperLTL$_C$, we prove that A-HyperLTL and HyperLTL$_S$ are expressively incomparable, and that HyperLTL$_C$ is not subsumed by A-HyperLTL or by HyperLTL$_S$. The question of whether A-HyperLTL and HyperLTL$_S$ are subsumed or not by HyperLTL$_C$ remains open. Additionally, we show that each of these logics is not subsumed by S1S[E]. This last result solves a recent open question [22, 2].

Since hyperproperties are a generalization of trace properties, we also investigate the expressive power of the considered asynchronous extensions of HyperLTL when interpreted on singleton sets of traces. For HyperLTL and its more expressive extension HyperLTL$_S$, singleton models are just the ones whose traces are LTL-definable and checking the existence of such a model (*single-trace satisfiability*) is decidable and PSPACE-complete. On the other hand, we show that for both A-HyperLTL and HyperLTL$_C$, single-trace satisfiability is highly undecidable being Σ_1^1-hard. Moreover, for HyperLTL$_C$ extended with past temporal modalities, we provide a nice characterization of the singleton models which generalizes the well-known equivalence of LTL and first-order logic FO[<] over traces established by Kamp's theorem. We show that over singleton models, HyperLTL$_C$ with past corresponds to the extension FO[<, +] of FO[<] with addition over variables.

Finally, we investigate the decidability frontier for model-checking HyperLTL$_C$ by enforcing the undecidability result of [6] and by identifying a maximal fragment of HyperLTL$_C$ for which model checking is decidable. This fragment subsumes HyperLTL and can be translated into FO[<,E]. Due to lack of space, many proofs are omitted and included in the longer version of this paper [7].

2 Preliminaries

Let $\mathbb{N}$ be the set of natural numbers. For all $i, j \in \mathbb{N}$, $[i, j]$ denotes the set of natural numbers h such that $i \leq h \leq j$. Given a word w over some alphabet Σ, $|w|$ is the length of w ($|w| = \infty$ if w is infinite). For each $0 \leq i < |w|$, $w(i)$ is the $(i+1)^{th}$ symbol of w, and w^i is the suffix of w from position i, i.e., the word $w(i)w(i+1)\ldots$.

We fix a finite set AP of atomic propositions. A *trace* is an infinite word over 2^{AP}. A *pointed trace* is a pair (π, i) consisting of a trace π and a position $i \in \mathbb{N}$. Two traces π and π' are *stuttering equivalent* if there are two infinite sequences of positions $0 = i_0 < i_1 \ldots$ and $0 = i'_0 < i'_1 \ldots$ s.t. for all $k \geq 0$ and for all $\ell \in [i_k, i_{k+1} - 1]$ and $\ell' \in [i'_k, i'_{k+1} - 1]$, $\pi(\ell) = \pi'(\ell')$. The trace π' is a *stuttering expansion* of the trace π if there is an infinite sequence of positions $0 = i_0 < i_1 \ldots$ such that for all $k \geq 0$ and for all $\ell \in [i_k, i_{k+1} - 1]$, $\pi'(\ell) = \pi(k)$.

Kripke structures. A *Kripke structure* (*over AP*) is a tuple $\mathcal{K} = \langle S, S_0, E, V\rangle$, where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, $E \subseteq S \times S$ is a transition relation and $V : S \to 2^{\mathsf{AP}}$ is an *AP-valuation* of the set of states. A *path* of $\mathcal{K}$ is an infinite sequence of states $t_0, t_1, \ldots$ such that $t_0 \in S_0$ and $(t_i, t_{i+1}) \in E$ for all $i \geq 0$. The Kripke structure $\mathcal{K}$ induces the set $\mathcal{L}(\mathcal{K})$ of traces of the form $V(t_0), V(t_1), \ldots$ such that $t_0, t_1, \ldots$ is a path of $\mathcal{K}$.

Relative Expressiveness. In Sections 3-4, we compare the expressiveness of various logics for linear-time hyperproperties. Let $\mathcal{M}$ be a set of models (in our case, a model is a set of traces), and L and L' be two logical languages interpreted over models in $\mathcal{M}$. Given two formulas $\varphi \in L$ and $\varphi' \in L'$, we say that φ and φ' are *equivalent* if for each model $M \in \mathcal{M}$, M satisfies φ iff M satisfies φ'. The language L *is subsumed by* L', denoted $L \leq L'$, if each formula in L has an equivalent formula in L'. The language L *is strictly less expressive than* L (written $L < L'$) if $L \leq L'$ and there is a L'-formula which has no equivalent in L. Finally, two logics L and L' *are expressively incomparable* if both $L \not\leq L'$ and $L' \not\leq L$.

Linear-time hyper specifications. We consider an abstract notion of linear-time hyper specifications which are interpreted over sets of traces. We fix a finite set VAR of trace variables. A *pointed-trace assignment* Π is a partial mapping over VAR assigning to each trace variable x in its domain $\mathit{Dom}(\Pi)$ a pointed trace. The assignment Π is *initial* if for each $x \in \mathit{Dom}(\Pi)$, $\Pi(x)$ is of the form $(\pi, 0)$ for some trace π. For a variable $x \in \mathsf{VAR}$ and a pointed trace (π, i), we denote by $\Pi[x \mapsto (\pi, i)]$ the pointed-trace assignment having domain $\mathit{Dom}(\Pi) \cup \{x\}$ defined as: $\Pi[x \mapsto (\pi, i)](x) = (\pi, i)$ and $\Pi[x \mapsto (\pi, i)](y) = \Pi(y)$ if $y \neq x$.

A *multi-trace specification* $\mathcal{S}(x_1, \ldots, x_n)$ is a specification (in some formalism) parameterized by a subset $\{x_1, \ldots, x_n\}$ of VAR whose semantics is given by a set Υ of *initial* pointed-trace assignments with domain $\{x_1, \ldots, x_n\}$: we write $\Pi \models S(x_1, \ldots, x_n)$ for the trace assignments Π in Υ. Given a class $\mathcal{C}$ of multi-trace specifications, *linear-time hyper expressions* ξ over $\mathcal{C}$ are defined as: $\xi \overset{\text{def}}{=} \exists x.\xi \mid \forall x.\xi \mid S(x_1, \ldots, x_n)$, where $x, x_1, \ldots, x_n \in \mathsf{VAR}$, $S(x_1, \ldots, x_n) \in \mathcal{C}$, and $\exists x$ (resp., $\forall x$) is the *hyper* existential (resp., universal) trace quantifier for variable x. An expression ξ is a *sentence* if every variable x_i in the multi-trace specification $S(x_1, \ldots, x_n)$ of ξ is *not free* (i.e., x_i is in the scope of a quantifier for variable x_i). The *quantifier alternation depth* of ξ is the number of switches between $\exists$ and $\forall$ quantifiers in the quantifier prefix of ξ. For a set $\mathcal{L}$ of traces and an initial pointed-trace assignment Π such that $\mathit{Dom}(\Pi)$ contains the free variables of ξ and the traces referenced by Π are in $\mathcal{L}$, the satisfaction relation $(\mathcal{L}, \Pi) \models \xi$ is inductively defined as follows:

$$\begin{array}{ll}
(\mathcal{L}, \Pi) \models \exists x.\xi & \Leftrightarrow \text{ for some trace } \pi \in \mathcal{L} : (\mathcal{L}, \Pi[x \mapsto (\pi, 0)]) \models \xi \\
(\mathcal{L}, \Pi) \models \forall x.\xi & \Leftrightarrow \text{ for each trace } \pi \in \mathcal{L} : (\mathcal{L}, \Pi[x \mapsto (\pi, 0)]) \models \xi \\
(\mathcal{L}, \Pi) \models S(x_1, \ldots, x_n) & \Leftrightarrow \Pi \models S(x_1, \ldots, x_n)
\end{array}$$

For a sentence ξ, we write $\mathcal{L} \models \xi$ to mean that $(\mathcal{L}, \Pi_\emptyset) \models \xi$, where $\Pi_\emptyset$ is the empty assignment. If $\mathcal{L} \models \xi$ we say that $\mathcal{L}$ is a *model* of ξ. If, additionally, $\mathcal{L}$ is a singleton we call it a *single-trace model*. By restricting our attention to the single-trace models, a linear-time hyper sentence ξ denotes a trace property consisting of the traces π such that $\{\pi\} \models \xi$. For a class $\mathcal{C}$ of multi-trace specifications, we consider the following decision problems:

- the *satisfiability* (resp., *single-trace satisfiability*) problem is checking for a linear-time hyper sentence ξ over $\mathcal{C}$, whether ξ has a model (resp., a single-trace model), and
- the model checking problem is checking for a Kripke structure $\mathcal{K}$ and a linear-time hyper sentence ξ over $\mathcal{C}$, whether $\mathcal{L}(\mathcal{K}) \models \xi$.

For instance, HyperLTL formulas are linear-time hyper sentences over the class of multi-trace specifications, called *HyperLTL quantifier-free formulas*, obtained by standard LTL formulas [27] by replacing atomic propositions p with relativized versions $p[x]$, where $x \in \mathsf{VAR}$. Intuitively, $p[x]$ asserts that p holds at the current position of the trace assigned to x. Given an HyperLTL quantifier-free formula $\psi(x_1, \ldots, x_n)$, an initial pointed trace assignment Π such that $Dom(\Pi) \supseteq \{x_1, \ldots, x_n\}$, and a position $i \geq 0$, the satisfaction relation $(\Pi, i) \models \psi$ is defined as a natural extension of the satisfaction relation $(\pi, i) \models \theta$ for LTL formulas θ and traces π. In particular,

- $(\Pi, i) \models p[x_k]$ if $\Pi(x_k) = (\pi, 0)$ and $p \in \pi(i)$,
- $(\Pi, i) \models \mathsf{X}\psi$ if $(\Pi, i+1) \models \psi$, and
- $(\Pi, i) \models \psi_1 \mathsf{U} \psi_2$ if there is $j \geq i$ such that $(\Pi, j) \models \psi_2$ and $(\Pi, k) \models \psi_1$ for all $k \in [i, j-1]$.

Asynchronous Word Automata and the Fixpoint Calculus H_μ. We shortly recall the framework of parity alternating asynchronous word automata (parity AAWA) [22], a class of finite-state automata for the asynchronous traversal of multiple infinite words. Intuitively, given $n \geq 1$, an AAWA with n tapes (nAAWA) has access to n infinite words over the input alphabet Σ and at each step, activates multiple copies where for each of them, there is exactly one input word whose current input symbol is consumed (i.e., the reading head of such word moves one position to the right). In particular, the target of a move of $\mathcal{A}$ is encoded by a pair (q, i), where q indicates the target state while the direction $i \in [1, n]$ indicates on which input word to progress. Details on the syntax and semantics of AAWA are given in [22, 7]. We denote by *Hyper AAWA* the class of linear-time hyper sentences over the multi-trace specifications given by parity AAWA. We also consider the fixpoint calculus H_μ introduced in [22] that provides a logical characterization of Hyper AAWA.

3 Advances in Asynchronous Extensions of HyperLTL

In this section, we investigate expressiveness and decidability issues on known asynchronous extensions of HyperLTL, namely, *Asynchronous HyperLTL* [2], *Stuttering HyperLTL* [6], and *Context HyperLTL* [6].

3.1 Results for Asynchronous HyperLTL (A-HyperLTL)

We first recall A-HyperLTL [2], a syntactical extension of HyperLTL which allows to express pure asynchronous hyperproperties. Then, we show that although A-HyperLTL does not subsume HyperLTL, HyperLTL can be embedded into A-HyperLTL by means of an additional proposition. Second, we establish that A-HyperLTL is subsumed by Hyper AAWA and the fixpoint calculus H_μ. Finally, we show that unlike HyperLTL, single-trace satisfiability of A-HyperLTL is undecidable.

The logic A-HyperLTL models the asynchronous passage of time between computation traces using the notion of a trajectory. Given a non-empty subset $V \subseteq \mathsf{VAR}$, a *trajectory over* V is an infinite sequence t of non-empty subsets of V. Intuitively, the positions $i \geq 0$ along t model the global time flow and for each position $i \geq 0$, $t(i)$ determines the trace variables in V whose associated traces make progress at time i. The trajectory t is *fair* if for each $x \in V$, there are infinitely many positions i such that $x \in t(i)$.

A-HyperLTL formulas are linear-time hyper sentences over multi-trace specifications ψ, called *A-HyperLTL quantifier-free formulas*, where ψ is of the form $\mathsf{E}\theta$ or $\mathsf{A}\theta$ and θ is a HyperLTL quantifier-free formula: E is the existential trajectory modality and A is the universal trajectory modality. Given a pointed trace assignment Π and a trajectory t over $Dom(\Pi)$, the *successor of* (Π, t), denoted by $(\Pi, t) + 1$, is defined as (Π', t'), where: (1) t' is the trajectory t^1 (the suffix of t from position 1), and (2) $Dom(\Pi') = Dom(\Pi)$ and for each $x \in Dom(\Pi)$ with $\Pi(x) = (\pi, i)$, $\Pi'(x) = (\pi, i + 1)$ if $x \in t(0)$, and $\Pi'(x) = \Pi(x)$ otherwise.

For each $k \geq 1$, we write $(\Pi, t)+k$ for denoting the pair (Π'', t'') obtained by k-applications of the successor function starting from (Π, t). Given a HyperLTL quantifier-free formula θ such that $Dom(\Pi)$ contains the variables occurring in θ, the satisfaction relations $\Pi \models \mathsf{E}\theta$, $\Pi \models \mathsf{A}\theta$, and $(\Pi, t) \models \theta$ are defined as follows (we omit the semantics of the Boolean connectives):

$$\begin{array}{rcll}
\Pi & \models & \mathsf{E}\theta & \Leftrightarrow \text{ for some fair trajectory } t \text{ over } Dom(\Pi),\ (\Pi, t) \models \theta \\
\Pi & \models & \mathsf{A}\theta & \Leftrightarrow \text{ for all fair trajectories } t \text{ over } Dom(\Pi),\ (\Pi, t) \models \theta \\
(\Pi, t) & \models & p[x] & \Leftrightarrow \Pi(x) = (\pi, i) \text{ and } p \in \pi(i) \\
(\Pi, t) & \models & \mathsf{X}\theta & \Leftrightarrow (\Pi, t) + 1 \models \theta \\
(\Pi, t) & \models & \theta_1 \mathsf{U} \theta_2 & \Leftrightarrow \text{for some } i \geq 0 : (\Pi, t) + i \models \theta_2 \text{ and } (\Pi, t) + k \models \theta_1 \text{ for all } 0 \leq k < i
\end{array}$$

We also exploit an alternative characterization of the semantics of quantifier-free A-HyperLTL formulas which easily follows from the definition of trajectories.

▶ **Proposition 1.** *Let θ be a quantifier-free HyperLTL formula over trace variables $x_1, \ldots, x_k$, and let $\pi_1, \ldots, \pi_k$ be k traces. Then:*

- $\{x_1 \leftarrow (\pi_1, 0), \ldots, x_k \leftarrow (\pi_k, 0)\} \models \mathsf{E}\theta$ *iff for all $i \in [1, k]$, there is a stuttering expansion π'_i of π_i such that $\{x_1 \leftarrow (\pi'_1, 0), \ldots, x_k \leftarrow (\pi'_k, 0)\} \models \theta$.*
- $\{x_1 \leftarrow (\pi_1, 0), \ldots, x_k \leftarrow (\pi_k, 0)\} \models \mathsf{A}\theta$ *iff for all $i \in [1, k]$ and for all stuttering expansions π'_i of π_i, it holds that $\{x_1 \leftarrow (\pi'_1, 0), \ldots, x_k \leftarrow (\pi'_k, 0)\} \models \theta$.*

A-HyperLTL versus HyperLTL. We now show that, unlike other temporal logics for asynchronous hyperproperties (see Sections 3.2 and 3.3), A-HyperLTL does not subsume HyperLTL. Given an atomic proposition p, we consider the following linear-time hyperproperty.

***p*-synchronicity:** a set $\mathcal{L}$ of traces satisfies the p-synchronicity hyperproperty if for all traces $\pi, \pi' \in \mathcal{L}$ and positions $i \geq 0$, $p \in \pi(i)$ iff $p \in \pi'(i)$.

This hyperproperty can be expressed in HyperLTL as follows: $\forall x_1. \forall x_2. \mathsf{G}(p[x_1] \leftrightarrow p[x_2])$. However, it cannot be expressed in A-HyperLTL (for details, see [7]).

▶ **Theorem 2.** *A-HyperLTL cannot express p-synchronicity. Hence, A-HyperLTL does not subsume HyperLTL.*

Though A-HyperLTL does not subsume HyperLTL, we can embed HyperLTL into A-HyperLTL by using an additional proposition $\# \notin \mathsf{AP}$ as follows. We can ensure that along a trajectory, traces progress at each global instant by requiring that proposition # holds exactly at the even positions. Formally, given a trace π over AP, we denote by $enc_{\#}(\pi)$ the trace over $\mathsf{AP} \cup \{\#\}$ defined as: $enc_{\#}(\pi)(2i) = \pi(2i) \cup \{\#\}$ and $enc_{\#}(\pi)(2i + 1) = \pi(2i + 1)$ for all $i \geq 0$. We extend the encoding $enc_{\#}$ to sets of traces $\mathcal{L}$ and assignments Π over AP in the obvious way. For each $x \in \mathsf{VAR}$, let $\theta_{\#}(x)$ be the following one-variable quantifier-free HyperLTL formula: $\#[x] \wedge \mathsf{G}(\#[x] \leftrightarrow \neg\mathsf{X}\#[x])$. It is easy to see that for a trace ρ over $\mathsf{AP} \cup \{\#\}$, a stuttering expansion ρ' of ρ satisfies $\theta_{\#}(x)$ with x bound to ρ' iff $\rho' = \rho$ and ρ is the #-encoding of some trace over AP. It follows that satisfiability of an HyperLTL formula φ can be reduced in linear-time to the satisfiability of the A-HyperLTL formula $\varphi_\#$ obtained from φ by replacing the quantifier-free part $\psi(x_1,\ldots,x_k)$ of φ with the quantifier-free A-HyperLTL formula $\mathsf{E}.\,(\psi(x_1,\ldots,x_k) \wedge \bigwedge_{i\in[1,k]} \theta_\#(x_i))$. For model checking, given a Kripke structure $\mathcal{K} = \langle S, S_0, E, V\rangle$, we construct in linear-time a Kripke structure $\mathcal{K}_\#$ over $\mathsf{AP} \cup \{\#\}$ such that $\mathcal{L}(\mathcal{K}_\#) = enc_\#(\mathcal{L}(\mathcal{K}))$. Formally, $\mathcal{K}_\# = \langle S \times \{0,1\},\, S_0 \times \{1\}, E', V'\rangle$ where $E' = \{((s,b),(s',1-b)) \mid (s,s') \in E \text{ and } b = 0,1\}$, $V'((s,1)) = V(s) \cup \{\#\}$ and $V'((s,0)) = V(s)$ for all $s \in S$. Thus, we obtain the following result.

▶ **Theorem 3.** *Satisfiability (resp., model checking) of HyperLTL can be reduced in linear-time to satisfiability (resp., model checking) of A-HyperLTL.*

A-HyperLTL versus Hyper AAWA and H_μ. We show that A-HyperLTL is subsumed by Hyper AAWA and H_μ. To this purpose, we exhibit an exponential-time translation of quantifier-free A-HyperLTL formulas into equivalent parity AAWA.

▶ **Theorem 4.** *Given an A-HyperLTL quantifier-free formula ψ with trace variables $x_1,\ldots,x_n$, one can build in singly exponential time a parity nAAWA $\mathcal{A}_\psi$ over 2^{AP} accepting the set of n-tuples $(\pi_1,\ldots,\pi_n)$ of traces such that $(\{x_1 \leftarrow (\pi_1,0),\ldots,x_n \leftarrow (\pi_n,0)\}) \models \psi$.*

Proof sketch. We first assume that ψ is of the form $\mathsf{E}\theta$ for some HyperLTL quantifier-free formula θ. By an adaptation of the standard automata theoretic approach for LTL [30], we construct a *nondeterministic* nAAWA (nNAWA) $\mathcal{A}_{\mathsf{E}\theta}$ equipped with standard generalized Büchi acceptance conditions which accepts a n-tuple $(\pi_1,\ldots,\pi_n)$ of traces iff there is a fair trajectory t such that $(\{x_1 \leftarrow (\pi_1,0),\ldots,x_n \leftarrow (\pi_n,0)\}), t \models \theta$. By standard arguments, a generalized Büchi nNAWA can be converted in quadratic time into an equivalent parity nNAWA. The behaviour of the automaton is subdivided into phases where each phase intuitively corresponds to a global timestamp. During a phase, $\mathcal{A}_{\mathsf{E}\theta}$ keeps tracks in its state of the guessed set of subformulas of θ that hold at the current global instant and guesses which traces progress at the next global instant by moving along a non-empty guessed set of directions in $\{1,\ldots,n\}$ in turns. In particular, after a movement along direction i, the automaton keeps track in its state of the previous chosen direction i and *either* moves to the next phase, *or* remains in the current phase by choosing a direction $j > i$. The transition function in moving from the end of a phase to the beginning of the next phase captures the semantics of the next modalities and the "local" fixpoint characterization of the until modalities. Moreover, the generalized Büchi acceptance condition is used for ensuring the fulfillment of the liveness requirements θ_2 in the until sub-formulas $\theta_1 \mathsf{U} \theta_2$, and for guaranteeing that the guessed trajectory is fair (i.e., for each direction $i \in [1,n]$, the automaton moves along i infinitely often). Details of the construction are given in [7].

Now, let us consider a quantifier-free A-HyperLTL formula of the form $\mathsf{A}\theta$ with trace variables $x_1,\ldots,x_n$, and let $\mathcal{A}_{\mathsf{E}\neg\theta}$ be the parity nAAWA associated with the formula $\mathsf{E}\neg\theta$. By [22], one can construct in linear-time (in the size of $\mathcal{A}_{\mathsf{E}\neg\theta}$), a parity nAAWA $\mathcal{A}_{\mathsf{A}\theta}$ accepting the complement of the language of n-tuples of traces accepted by $\mathcal{A}_{\mathsf{E}\neg\theta}$. ◀

Thus, being H_μ and Hyper AAWA expressively equivalent, we obtain the following result.

▶ **Corollary 5.** *Hyper AAWA subsumes A-HyperLTL. H_μ also subsumes A-HyperLTL.*

Undecidability of single-trace satisfiability for A-HyperLTL. It is easy to see that for HyperLTL, single-trace satisfiability corresponds to LTL satisfiability (hence, it is PSPACE-complete). We show now that for A-HyperLTL, the problem is highly undecidable being at least Σ_1^1-hard. The crucial observation is that we can enforce alignment requirements on distinct stuttering expansions of the same trace which allow to encode recurrent computations of *non-deterministic* 2*-counter machines* [23]. Recall that such a machine is a tuple $M = \langle Q, \Delta, \delta_{init}, \delta_{rec} \rangle$, where Q is a finite set of (control) locations, $\Delta \subseteq Q \times L \times Q$ is a transition relation over the instruction set $L = \{\mathsf{inc}, \mathsf{dec}, \mathsf{if\_zero}\} \times \{1, 2\}$, and $\delta_{init} \in \Delta$ and $\delta_{rec} \in \Delta$ are two designated transitions, the initial and the recurrent one.

An M-configuration is a pair (δ, ν) consisting of a transition $\delta \in \Delta$ and a counter valuation $\nu : \{1, 2\} \to \mathbb{N}$. A computation of M is an *infinite* sequence of configurations of the form $((q_0, (op_0, c_0), q_1), \nu_0), ((q_1, (op_1, c_1), q_2), \nu_1), \ldots$ such that for each $i \geq 0$:

- $\nu_{i+1}(3 - c_i) = \nu_i(3 - c_i)$;
- $\nu_{i+1}(c_i) = \nu_i(c_i) + 1$ if $op_i = \mathsf{inc}$, and $\nu_{i+1}(c_i) = \nu_i(c_i) - 1$ if $op_i = \mathsf{dec}$;
- $\nu_{i+1}(c_i) = \nu_i(c_i) = 0$ if $op_i = \mathsf{if\_zero}$.

The recurrence problem is to decide whether for a given machine M, there is a computation starting at the initial configuration (δ_{init}, ν_0), where $\nu_0(c) = 0$ for each $c \in \{1, 2\}$, which visits δ_{rec} infinitely often. This problem is known to be Σ_1^1-complete [23].

▶ **Theorem 6.** *The single-trace satisfiability problem of* A-HyperLTL *is* Σ_1^1*-hard.*

Proof sketch. Let $M = \langle Q, \Delta, \delta_{init}, \delta_{rec} \rangle$ be a counter machine. We construct a two-variable A-HyperLTL formula φ_M such that M is a positive instance of the recurrence problem if and only if φ_M has a single-trace model. The set of atomic propositions is $\mathsf{AP} \overset{\text{def}}{=} \Delta \cup \{c_1, c_2, \#, \mathit{beg}, \mathit{pad}\}$. Intuitively, propositions c_1 and c_2 are used to encode the values of the two counters in M and # is used to ensure that the values of the counters are not modified in the stuttering expansions of a trace encoding a computation of M. Proposition *beg* marks the beginning of a configuration code, and proposition *pad* is exploited for encoding a padding word at the end of a configuration code: formula φ_M will ensure that only these words can be "expanded" in the stuttering expansions of a trace. Formally, an M-configuration (δ, ν) is encoded by the finite words over 2^{AP} (called *segments*) of the form $\{\mathit{beg}, \delta\} P_1 \ldots P_m \{\mathit{pad}\}^k$, where $k \geq 1$, $m = \max(\nu(1), \nu(2))$, and for all $i \in [1, m]$,

- $\emptyset \neq P_i \subseteq \{\#, c_1, c_2\}$,
- $\# \in P_i$ iff i is odd, and
- for all $\ell \in \{1, 2\}$, $c_\ell \in P_i$ iff $i \leq \nu(\ell)$.

A computation ρ of M is then encoded by the traces obtained by concatenating the codes of the configurations along ρ starting from the first one. The A-HyperLTL formula φ_M is given by $\exists x_1 \exists x_2. \mathsf{E}\, \psi$, where the quantifier-free HyperLTL formula ψ guarantees that for the two stuttering expansions π_1 and π_2 of the given trace π, the following holds:

- both π_1 and π_2 are infinite concatenations of segments;
- the first segment of π_1 encodes the initial configuration (δ_{init}, ν_0) of M and the second segment of π_1 encodes a configuration which is a successor of (δ_{init}, ν_0) in M;
- δ_{rec} occurs infinitely often along π_1;
- for each $i \geq 2$, the $(i+1)^{th}$ segment s_2 of π_2 starts at the same position as the i^{th} segment s_1 of π_1. Moreover, s_1 and s_2 have the same length and the configuration encoded by s_2 is a successor in M of the configuration encoded by s_1.

Now, since π_1 and π_2 are stuttering expansions of the same trace π, the alternation requirement for proposition # in the encoding of an M-configurations ensures that π_1 and π_2 encode the same infinite sequence of M-configurations. Hence, φ_M has a single-trace model if and only if M is a positive instance of the recurrence problem. Details appear in [7]. ◀

3.2 Results for Stuttering HyperLTL (HyperLTL$_S$)

Stuttering HyperLTL (HyperLTL$_S$) [6] is an asynchronous extension of HyperLTL obtained by using *stutter-relativized* versions of the temporal modalities w.r.t. finite sets Γ of LTL formulas. In this section, we show that A-HyperLTL and HyperLTL$_S$ are expressively incomparable.

In HyperLTL$_S$, the notion of successor of a position i along a trace π is relativized using a finite set Γ of LTL formulas. If in the interval $[i, \infty[$, the truth value of each formula in Γ does not change along π (i.e., for each $j \geq i$ and for each $\theta \in \Gamma$, $(\pi, i) \models \theta$ iff $(\pi, j) \models \theta$), then the Γ-*successor of* i *in* π coincides with the local successor $i + 1$. Otherwise, the Γ-*successor of* i *in* π is the smallest position $j > i$ such that the truth value of some formula θ in Γ changes in moving from i to j (i.e., for some $\theta \in \Gamma$, $(\pi, i) \models \theta$ iff $(\pi, j) \not\models \theta$). The Γ-successor induces a trace, called Γ-*stutter trace of* π and denoted by $\mathit{stfr}_\Gamma(\pi)$, obtained from π by repeatedly applying the Γ-successor starting from position 0, i.e. $\mathit{stfr}_\Gamma(\pi) \overset{\text{def}}{=} \pi(i_0)\pi(i_1)\ldots$, where $i_0 = 0$ and i_{k+1} is the Γ-successor of i_k in π for all $k \geq 0$. Note that $\mathit{stfr}_\Gamma(\pi) = \pi$ if $\Gamma = \emptyset$. Given a pointed-trace assignment Π, the Γ-*successor* $\mathit{succ}_\Gamma(\Pi)$ *of* Π is the pointed trace-assignment with domain $\mathit{Dom}(\Pi)$ defined as follows for each $x \in \mathit{Dom}(\Pi)$: if $\Pi(x) = (\pi, i)$, then $\mathit{succ}_\Gamma(\Pi)(x) = (\pi, \ell)$ where ℓ is the Γ-successor of i in π. For each $j \in \mathbb{N}$, we use succ^j_Γ for the function obtained by j applications of the function succ_Γ.

HyperLTL$_S$ formulas are linear-time hyper sentences over multi-trace specifications ψ, called *HyperLTL$_S$ quantifier-free formulas*, where the syntax of ψ is as follows:

$$\psi ::= \top \mid p[x] \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}_\Gamma\psi \mid \psi \mathsf{U}_\Gamma \psi$$

where $p \in \mathsf{AP}$, $x \in \mathsf{VAR}$, Γ is a finite set of LTL formulas over AP, and X_Γ and U_Γ are the stutter-relativized versions of the LTL temporal modalities. Informally, the relativized temporal modalities X_Γ and U_Γ are evaluated by a lockstepwise traversal of the Γ-stutter traces associated with the currently quantified traces. Standard HyperLTL corresponds to the fragment of HyperLTL$_S$ where the subscript of each temporal modality is the empty set $\emptyset$.

Given a HyperLTL$_S$ quantifier-free formula ψ and a pointed trace assignment Π such that $\mathit{Dom}(\Pi)$ contains the trace variables occurring in ψ, the satisfaction relation $\Pi \models \psi$ is inductively defined as follows (we omit the semantics of the Boolean connectives):

$$\begin{array}{ll}
\Pi \models p[x] & \Leftrightarrow \Pi(x) = (\pi, i) \text{ and } p \in \pi(i) \\
\Pi \models \mathsf{X}_\Gamma\psi & \Leftrightarrow \mathit{succ}_\Gamma(\Pi) \models \psi \\
\Pi \models \psi_1 \mathsf{U}_\Gamma \psi_2 & \Leftrightarrow \text{for some } i \geq 0 : \ \mathit{succ}^i_\Gamma(\Pi) \models \psi_2 \text{ and } \mathit{succ}^k_\Gamma(\Pi) \models \psi_1 \text{ for all } 0 \leq k < i
\end{array}$$

Stuttering LTL formulas, corresponding to one-variable HyperLTL$_S$ quantifier-free formulas, can be translated in polynomial time into equivalent LTL formulas (see [6]). Thus, since LTL satisfiability is PSPACE-complete, the following result holds.

▶ **Proposition 7.** *The trace properties definable by HyperLTL$_S$ formulas are LTL definable, and single-trace satisfiability of HyperLTL$_S$ is* PSPACE*-complete.*

HyperLTL$_S$ versus A-HyperLTL. We show that HyperLTL$_S$ and A-HyperLTL are expressively incomparable even over singleton sets of atomic propositions. By Theorem 2, unlike HyperLTL$_S$, A-HyperLTL does not subsume HyperLTL even when $|\mathsf{AP}| = 1$. Hence, HyperLTL$_S$ is not subsumed by A-HyperLTL even when $|\mathsf{AP}| = 1$. We show now that the converse holds as well. Intuitively, A-HyperLTL can encode counting mechanisms which cannot be expressed in HyperLTL$_S$. Let $\mathsf{AP} = \{p\}$. We exhibit two families $\{\mathcal{L}_n\}_{n\geq 1}$ and $\{\mathcal{L}'_n\}_{n\geq 1}$ of trace sets and an A-HyperLTL formula φ_A such that

- φ_A can distinguish the traces set $\mathcal{L}_n$ and $\mathcal{L}'_n$ for each $n \geq 1$, but
- for each HyperLTL$_S$ formula ψ, there is n such that ψ does not distinguish $\mathcal{L}_n$ and $\mathcal{L}'_n$.

For each $n \geq 1$, let π_n, ρ_n, and ρ'_n be the traces defined as:

$$\pi_n \overset{\text{def}}{=} (\emptyset \cdot p)^n \cdot \emptyset^\omega \qquad \rho_n \overset{\text{def}}{=} (\emptyset \cdot p)^{2n} \cdot \emptyset^\omega \qquad \rho'_n \overset{\text{def}}{=} (\emptyset \cdot p)^{2n+1} \cdot \emptyset^\omega$$

For each $n \geq 1$, define $\mathcal{L}_n \overset{\text{def}}{=} \{\pi_n, \rho_n\}$ and $\mathcal{L}'_n \overset{\text{def}}{=} \{\pi_n, \rho'_n\}$. Let $\psi_1(x)$ and $\psi_2(x)$ be two one-variable quantifier-free HyperLTL formulas capturing the following requirements:

- $\psi_1(x)$ captures traces of the form $(\emptyset \cdot p)^k \cdot \emptyset^\omega$ for some $k \geq 1$,
- $\psi_2(x)$ captures traces of the form $(\emptyset^2 \cdot p^2)^k \cdot \emptyset^\omega$ for some $k \geq 1$.

Let $\psi(x, y)$ be the two-variable quantifier-free HyperLTL formula defined as follows:

$$\psi(x, y) \overset{\text{def}}{=} \mathsf{F}(p[x] \wedge p[y] \wedge \mathsf{XG}(\neg p[x] \wedge \neg p[y]))$$

Intuitively, if x is bound to a trace ν_1 satisfying ψ_1 and y is bound to a trace ν_2 satisfying ψ_2, then $\psi(x, y)$ holds iff ν_1 is of the form $(\emptyset \cdot p)^{2k}$ and ν_2 is of the form $(\emptyset^2 \cdot p^2)^k$ for some $k \geq 1$. The A-HyperLTL formula φ_A is then defined as follows:

$$\varphi_A \overset{\text{def}}{=} \forall x_1.\forall x_2.\mathsf{E}\left([\psi(x_1, x_2) \wedge \psi_1(x_1) \wedge \psi_1(x_2)] \vee [\psi(x_1, x_2) \wedge \bigvee_{i \in \{1,2\}} (\psi_1(x_i) \wedge \psi_2(x_{3-i}))]\right)$$

Let π a trace of the form $\pi = (\emptyset \cdot p)^k \cdot \emptyset^\omega$ for some $k \geq 1$. We observe that the unique stuttering expansion ν_1 of π such that ν_1 satisfies $\psi_1(x)$ is π itself. Similarly, there is a unique stuttering expansion ν_2 of π such that ν_2 satisfies $\psi_2(x)$, and such a trace ν_2 is given by $(\emptyset^2 \cdot p^2)^k \cdot \emptyset^\omega$. Fix $n \geq 1$. Let us consider the trace set $\mathcal{L}_n = \{\pi_n, \rho_n\}$. By construction, if both variables x_1 and x_2 in the definition of φ_A are bound to the same trace π in $\mathcal{L}_n$, then the first disjunct in the definition of φ_A is fulfilled by taking π itself as an expansion of π. On the other hand, assume that variable x_1 (resp., x_2) is bound to trace π_n and variable x_2 (resp., x_1) is bound to trace ρ_n. In this case, by taking as stuttering expansion of π_n the trace $(\emptyset^2 \cdot p^2)^n \cdot \emptyset^\omega$ and as stuttering expansion of $\rho_n = (\emptyset \cdot p)^{2n} \cdot \emptyset^\omega$ the trace ρ_n itself, the second disjunct in the definition of φ_A is fulfilled. Hence, $\mathcal{L}_n$ is a model of φ_A.

Now, we show that $\mathcal{L}'_n = \{\pi_n, \rho'_n\}$ does not satisfy φ_A. Let us consider the mapping assigning to variable x_1 the trace π_n and to variable x_2 the trace ρ'_n. With this mapping, the quantifier-free part of φ_A cannot be fulfilled. This because the unique stuttering expansion of $\pi_n = (\emptyset \cdot p)^n \cdot \emptyset^\omega$ (resp., $\rho'_n = (\emptyset \cdot p)^{2n+1} \cdot \emptyset^\omega$) satisfying ψ_1 is π_n (resp., ρ'_n) itself. Moreover, the unique stuttering expansion of π_n (resp., ρ'_n) satisfying ψ_2 is $(\emptyset^2 \cdot p^2)^n \cdot \emptyset^\omega$ (resp., $(\emptyset^2 \cdot p^2)^{2n+1} \cdot \emptyset^\omega$). Hence, for all $n \geq 1$, $\mathcal{L}_n \models \varphi_A$ and $\mathcal{L}'_n \not\models \varphi_A$. On the other hand, one can show that the following holds (for details, see [7]).

▶ **Proposition 8.** *For each HyperLTL$_S$ formula ψ, there is $n \geq 1$ s.t. $\mathcal{L}_n \models \psi$ iff $\mathcal{L}'_n \models \psi$.*

Thus, since A-HyperLTL does not subsume HyperLTL$_S$, we obtain the following result.

▶ **Corollary 9.** *A-HyperLTL and HyperLTL$_S$ are expressively incomparable.*

3.3 Results for Context HyperLTL (HyperLTL$_C$)

Context HyperLTL (HyperLTL$_C$) [6] extends HyperLTL by unary modalities $\langle C \rangle$ parameterized by a non-empty subset C of trace variables – called the *context* – which restrict the evaluation of the temporal modalities to the traces associated with the variables in C. We show that HyperLTL$_C$ is not subsumed by A-HyperLTL or HyperLTL$_S$, and single-trace satisfiability of HyperLTL$_C$ is undecidable. Moreover, we provide a characterization of the *finite trace properties* denoted by HyperLTL$_C$ formulas in terms of the extension FO$_f$[<, +] of standard first-order logic FO$_f$[<] over finite words with *addition.* We also establish that the variant of

$\mathsf{FO}_f[<,+]$ over infinite words characterizes the trace properties expressible in the extension of $\mathsf{HyperLTL}_C$ with past temporal modalities. Finally, we identify a fragment of $\mathsf{HyperLTL}_C$ which subsumes $\mathsf{HyperLTL}$ and for which model checking is decidable.

$\mathsf{HyperLTL}_C$ formulas are linear-time hyper sentences over multi-trace specifications ψ, called *$\mathsf{HyperLTL}_C$ quantifier-free formulas*, where the syntax of ψ is as follows:

$$\psi ::= \top \mid p[x] \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid \langle C \rangle \psi$$

where $p \in \mathsf{AP}$, $x \in \mathsf{VAR}$, and $\langle C \rangle$ is the context modality with $\emptyset \neq C \subseteq \mathsf{VAR}$. A context C is *global for a formula* φ if C contains all the trace variables occurring in φ. Let Π be a pointed-trace assignment. Given a context C and an offset $i \geq 0$, we denote by $\Pi +_C i$ the pointed-trace assignment with domain $\mathit{Dom}(\Pi)$ defined as follows. For each $x \in \mathit{Dom}(\Pi)$, where $\Pi(x) = (\pi, h)$: $[\Pi +_C i](x) = (\pi, h + i)$ if $x \in C$, and $[\Pi +_C i](x) = \Pi(x)$ otherwise. Intuitively, the positions of the pointed traces associated with the variables in C advance of the offset i, while the positions of the other pointed traces remain unchanged. Let ψ be a $\mathsf{HyperLTL}_C$ quantifier-free formula such that $\mathit{Dom}(\Pi)$ contains the variables occurring in ψ. The satisfaction relation $(\Pi, C) \models \psi$ is defined as follows (we omit the semantics of the Boolean connectives):

$$\begin{array}{ll}
(\Pi, C) \models p[x] & \Leftrightarrow \Pi(x) = (\pi, i) \text{ and } p \in \pi(i) \\
(\Pi, C) \models \mathsf{X}\psi & \Leftrightarrow (\Pi +_C 1, C) \models \psi \\
(\Pi, C) \models \psi_1 \mathsf{U} \psi_2 & \Leftrightarrow \text{for some } i \geq 0 : \ (\Pi +_C i, C) \models \psi_2 \text{ and } (\Pi +_C k, C) \models \psi_1 \text{ for all } k < i \\
(\Pi, C) \models \langle C' \rangle \psi & \Leftrightarrow (\Pi, C') \models \psi
\end{array}$$

We write $\Pi \models \psi$ to mean that $(\Pi, \mathsf{VAR}) \models \psi$.

$\mathsf{HyperLTL}_C$ versus A-HyperLTL and $\mathsf{HyperLTL}_S$. We show that $\mathsf{HyperLTL}_C$ is able to capture powerful non-regular trace properties which cannot be expressed in A-HyperLTL or in $\mathsf{HyperLTL}_S$. In particular, we consider the following trace property over $\mathsf{AP} = \{p\}$:

Suffix Property: a trace π satisfies the suffix property if π has a proper suffix π^i for some $i > 0$ such that $\pi^i = \pi$.

This property can be expressed in $\mathsf{HyperLTL}_C$ by the following formula

$$\varphi_{\mathit{suff}} \stackrel{\text{def}}{=} \forall x_1. \forall x_2. \bigwedge_{p \in \mathsf{AP}} \mathsf{G}(p[x_1] \leftrightarrow p[x_2]) \wedge \{x_2\}\mathsf{F}\mathsf{X}\{x_1, x_2\} \bigwedge_{p \in \mathsf{AP}} \mathsf{G}(p[x_1] \leftrightarrow p[x_2])$$

We show that no A-HyperLTL and no $\mathsf{HyperLTL}_S$ formula is equivalent to φ_{suff}.

▶ **Theorem 10.** *A-HyperLTL and $\mathsf{HyperLTL}_S$ cannot express the suffix property. Hence, $\mathsf{HyperLTL}_C$ is not subsumed by A-HyperLTL or by $\mathsf{HyperLTL}_S$.*

Proof sketch. By Proposition 7, the set of single-trace models of a $\mathsf{HyperLTL}_S$ formula is regular. Thus, since the suffix trace property is not regular, the result for $\mathsf{HyperLTL}_S$ follows.

Consider now A-HyperLTL. For each $n \geq 1$, let $\pi_n \stackrel{\text{def}}{=} (p^n \cdot \emptyset)^\omega$ and $\pi'_n \stackrel{\text{def}}{=} p^{n+1} \cdot \emptyset \cdot (p^n \cdot \emptyset)^\omega$. By construction π_n satisfies the suffix property but π'_n not. Hence, for each $n \geq 1$, the $\mathsf{HyperLTL}_C$ formula φ_{suff} distinguishes the singleton sets $\{\pi_n\}$ and $\{\pi'_n\}$. On the other hand, we can show the following result, hence, Theorem 10 directly follows (a proof of the following claim is given in [7]).

▷ Claim. For each A-HyperLTL formula ψ, there is $n \geq 1$ such that $\{\pi_n\} \models \psi$ *iff* $\{\pi'_n\} \models \psi$. ◀

Single-trace satisfiability and characterization of $\mathsf{HyperLTL}_C$ finite-trace properties. Like A-HyperLTL, and unlike HyperLTL and $\mathsf{HyperLTL}_S$, single-trace satisfiability of $\mathsf{HyperLTL}_C$ turns out to be undecidable. In particular, by a straightforward adaptation of the undecidability proof in [6] for model checking $\mathsf{HyperLTL}_C$, one can reduce the recurrence problem in Minsky counter machines [23] to single-trace satisfiability of $\mathsf{HyperLTL}_C$.

▶ **Theorem 11.** *The single-trace satisfiability problem for A-HyperLTL is Σ_1^1-hard.*

A finite trace (over AP) is a finite non-empty word over 2^{AP}. By adding a fresh proposition $\# \notin \mathsf{AP}$, a finite trace can be encoded by the trace $enc(w)$ over $\mathsf{AP} \cup \{\#\}$ given by $w \cdot \{\#\}^\omega$. Given a $\mathsf{HyperLTL}_C$ formula φ over $\mathsf{AP} \cup \{\#\}$, the *finite-trace property denoted by* φ is the language $\mathcal{L}_f(\varphi)$ of finite traces w over AP such that the single-trace model $\{enc(w)\}$ satisfies φ. We provide now a characterization of the finite-trace properties denoted by $\mathsf{HyperLTL}_C$ formulas over $\mathsf{AP} \cup \{\#\}$ in terms of the extension $\mathsf{FO}_f[<,+]$ of standard first-order logic $\mathsf{FO}_f[<]$ over finite words on 2^{AP} with addition. Formally, $\mathsf{FO}_f[<,+]$ is a first-order logic with equality over the signature $\{<,+\} \cup \{P_a \mid a \in \mathsf{AP}\}$, where the atomic formulas ψ have the following syntax with x, y and z being first-order variables: $\psi \overset{\text{def}}{=} x = y \mid x < y \mid z = y + x \mid P_a(x)$. A $\mathsf{FO}_f[<,+]$ sentence (i.e., a $\mathsf{FO}_f[<,+]$ formula with no free variables) is interpreted over finite traces w, where: (i) variables ranges over the set $\{0, \ldots, |w| - 1\}$ of positions of w, (ii) the binary predicate $<$ is the natural ordering on $\{0, \ldots, |w| - 1\}$, and (iii) the predicates $z = y + x$ and $P_a(x)$ are interpreted in the obvious way. We establish the following result.

▶ **Theorem 12.** *Given a $\mathsf{FO}_f[<,+]$ sentence φ over AP, one can construct in polynomial time a $\mathsf{HyperLTL}_C$ formula ψ over $\mathsf{AP} \cup \{\#\}$ such that $\mathcal{L}_f(\psi)$ is the set of models of φ. Vice versa, given a $\mathsf{HyperLTL}_C$ formula ψ over $\mathsf{AP} \cup \{\#\}$, one can construct in single exponential time a $\mathsf{FO}_f[<,+]$ sentence φ whose set of models is $\mathcal{L}_f(\psi)$.*

Intuitively, when a $\mathsf{HyperLTL}_C$ formula ψ over $\mathsf{AP} \cup \{\#\}$ is interpreted over singleton models $\{enc(w)\}$ for a given finite trace w, the trace variables in the quantifier-free part of ψ and the temporal modalities evaluated in different contexts can simulate quantification over positions in w and the atomic formulas of $\mathsf{FO}_f[<,+]$. In particular, the addition predicate $z = x + y$ can be simulated by requiring that two segments of w whose endpoints are referenced by trace variables have the same length: this is done by shifting with the eventually modality in a suitable context the left segment of a non-negative offset, and by checking that the endpoints of the resulting segments coincide. Note that for trace variables x and y which refer to positions i and j of w, one can require that i and j coincide by the $\mathsf{HyperLTL}_C$ formula $\{x, y\}\mathsf{F}(\neg\#[x] \wedge \neg\#[y] \wedge \mathsf{X}(\#[x] \wedge \#[y]))$. Similarly, if we consider the extension of $\mathsf{HyperLTL}_C$ with the past counterparts of the temporal modalities, then the trace properties denoted by past $\mathsf{HyperLTL}_C$ formulas correspond to the ones denoted by sentences in the variant $\mathsf{FO}[<,+]$ of $\mathsf{FO}_f[<,+]$ over infinite words on 2^{AP} (traces). For arbitrary traces, past temporal modalities are crucial for enforcing that two variables refer to the same position (for details see [7]).

▶ **Theorem 13.** *Past $\mathsf{HyperLTL}_C$ and $\mathsf{FO}[<,+]$ capture the same class of trace properties.*

Results about model checking $\mathsf{HyperLTL}_C$. Model checking $\mathsf{HyperLTL}_C$ is known to be undecidable even for formulas where the unique temporal modality occurring in the scope of a non-global context is F. For the fragment where the unique temporal modality occurring in a non-global context is X, then the problem is decidable. This fragment has the same expressiveness as HyperLTL but it is exponentially more succinct than HyperLTL [6]. We provide now new insights on model checking $\mathsf{HyperLTL}_C$. On the negative side, we show that model checking is undecidable even for the fragment $\mathcal{U}$ consisting of two-variable quantifier alternation-free formulas of the form $\exists x_1.\exists x_2.\, \psi_0 \wedge \{x_2\}\mathsf{F}\{x_1, x_2\}\psi$, where ψ_0 and ψ are quantifier-free HyperLTL formulas. A proof of Theorem 14 appears in [7].

▶ **Theorem 14.** *Model-checking against the fragment $\mathcal{U}$ of $\mathsf{HyperLTL}_C$ is Σ_0^1-hard.*

By Theorem 14, $\mathsf{HyperLTL}_C$ model checking becomes undecidable whenever in a formula a non-singleton context C occurs within a distinct context $C' \neq C$. Thus, we consider the fragment of $\mathsf{HyperLTL}_C$, called *simple $\mathsf{HyperLTL}_C$*, where each context C which occurs in the scope of a distinct context $C' \neq C$ is a singleton. Note that simple $\mathsf{HyperLTL}_C$ subsumes HyperLTL.

▶ **Theorem 15.** *The model checking problem of simple $\mathsf{HyperLTL}_C$ is decidable.*

Theorem 15 is proven by a polynomial time translation of simple $\mathsf{HyperLTL}_C$ formulas into equivalent sentences of *first-order logic FO[<,E] with the equal-level predicate E* (see [16]) whose model checking problem is known to be decidable [10]. This logic is interpreted over sets $\mathcal{L}$ of traces, and first-order variables refer to pointed traces over $\mathcal{L}$. In simple $\mathsf{HyperLTL}_C$, the evaluation of temporal modalities is subdivided in two phases. In the first phase, modalities are evaluated by a synchronous traversal of the traces bound to the variables in a non-singleton context. In the second phase, the temporal modalities are evaluated along a single trace and singleton contexts allows to switch from a trace to another one by enforcing a weak form of mutual temporal relation. This behaviour can be encoded in FO[<,E] (for details, see [7]).

3.4 H_μ versus A-HyperLTL, $\mathsf{HyperLTL}_S$, and $\mathsf{HyperLTL}_C$

Both $\mathsf{HyperLTL}_S$ and $\mathsf{HyperLTL}_C$ are subsumed by H_μ and Hyper AAWA [6]. In particular, quantifier-free formulas of $\mathsf{HyperLTL}_S$ and $\mathsf{HyperLTL}_C$ can be translated in polynomial time into equivalent Büchi AAWA. Corollary 5 shows that A-HyperLTL is subsumed by H_μ as well. Thus, since A-HyperLTL and $\mathsf{HyperLTL}_S$ are expressively incomparable (by Corollary 9), there is an H_μ formula which cannot be expressed in A-HyperLTL (resp., $\mathsf{HyperLTL}_S$). Therefore, we obtain the following corollary.

▶ **Corollary 16.** *H_μ is strictly more expressive than A-HyperLTL and $\mathsf{HyperLTL}_S$, and subsumes $\mathsf{HyperLTL}_C$.*

4 Asynchronous vs Synchronous Extensions of HyperLTL

We compare now the expressiveness of the asynchronous extensions of HyperLTL against S1S[E] [10]. S1S[E] is a monadic second-order logic with equality over the signature $\{<, E\} \cup \{P_a \mid a \in \mathsf{AP}\}$ which syntactically extends the monadic second-order logic of one successor S1S with the equal-level binary predicate E. While S1S is interpreted over traces, S1S[E] is interpreted over sets of traces. A set $\mathcal{L}$ of traces induces the relational structure with domain $\mathcal{L} \times \mathbb{N}$ (i.e., the set of pointed traces associated with $\mathcal{L}$), where

- the binary predicate $<$ is interpreted as the set of pairs of pointed traces in $\mathcal{L} \times \mathbb{N}$ of the form $((\pi, i_1), (\pi, i_2))$ such that $i_1 < i_2$, and
- the equal-level predicate E is interpreted as the set of pairs of pointed traces in $\mathcal{L} \times \mathbb{N}$ of the form $((\pi_1, i), (\pi_2, i))$.

Hence, $<$ allows to compare distinct timestamps along the same trace, while the equal-level predicate allows to compare distinct traces at the same timestamp. For a formal definition of the syntax and semantics of $\mathsf{S1S[E]}$, see [7].

We show that the considered asynchronous extensions of $\mathsf{HyperLTL}$ are not subsumed by $\mathsf{S1S[E]}$. Intuitively, for some $k \geq 1$, the logic $\mathsf{A\text{-}HyperLTL}$ (resp., $\mathsf{HyperLTL}_S$, resp., $\mathsf{HyperLTL}_C$) can express hyperproperties whose set of models having cardinality k (*k-models*) can be encoded by a non-regular set of traces. On the other hand, we show that the encoding of the k-models of a $\mathsf{S1S[E]}$ formula always leads to a regular language.

Let $k \geq 1$. We consider the set of atomic propositions given by $\mathsf{AP} \times [1, k]$ for encoding sets $\mathcal{L}$ of traces (over AP) having cardinality k by traces over $\mathsf{AP} \times [1, k]$.

A trace ν over $\mathsf{AP} \times [1, k]$ is *well-formed* if for all $\ell, \ell' \in [1, k]$ with $\ell \neq \ell'$, there is $i \in \mathbb{N}$ and $p \in \mathsf{AP}$ so that $(p, \ell) \in \nu(i)$ iff $(p, \ell') \notin \nu(i)$. A well-formed trace ν over $\mathsf{AP} \times [1, k]$ encodes the set $\mathcal{L}(\nu)$ of the traces π (over AP) such that there is $\ell \in [1, k]$ where π corresponds to the projection of ν over $\mathsf{AP} \times \{\ell\}$, i.e. for each $i \geq 0$, $\pi(i) = \{p \in \mathsf{AP} \mid (p, \ell) \in \nu(i)\}$. Since ν is well–formed, $|\mathcal{L}(\nu)| = k$ and we say that ν is a *k-code* of $\mathcal{L}(\nu)$. Note that for a set $\mathcal{L}$ of traces (over AP) of cardinality k, each ordering of the traces in $\mathcal{L}$ induces a distinct k-code.

Given an hyperproperty specification ξ over AP, a *k-model of ξ* is a set of traces satisfying ξ having cardinality k. The *k-language of ξ* is the set of k-codes associated with the k-models of ξ. The specification ξ is *k-regular* if its k-language is a regular language over $\mathsf{AP} \times [1, k]$.

We first show that for each $k \geq 1$, $\mathsf{S1S[E]}$ sentences are k-regular.

▶ **Lemma 17.** *Let $k \geq 1$ and φ be a $\mathsf{S1S[E]}$ sentence over AP. Then, one can construct a S1S sentence φ' over AP $\times [1, k]$ whose set of models is the k-language of φ.*

A proof of Lemma 17 appears in [7]. Since $\mathsf{S1S}$ sentences capture only regular languages of traces, by Lemma 17, we obtain the following result.

▶ **Proposition 18.** *Let $k \geq 1$ and φ be a $\mathsf{S1S[E]}$ sentence over AP. Then, φ is k-regular.*

We now show that given one of the considered asynchronous extensions L of $\mathsf{HyperLTL}$, there is $k \geq 1$ and a L formula φ such that φ is *not* k-regular.

▶ **Proposition 19.** *There is a HyperLTL$_C$ formula over $\{p\}$ which is not 1-regular, and there are A-HyperLTL and HyperLTL$_S$ formulas over $\{p\}$ which are not 2-regular.*

Proof. Let $\mathsf{AP} = \{p\}$. The $\mathsf{HyperLTL}_C$ formula φ_{suff} defined in Section 3.3 whose models consist of the singletons $\{\pi\}$ such that π satisfies the suffix property is not 1-regular.

Consider now $\mathsf{A\text{-}HyperLTL}$ and $\mathsf{HyperLTL}_S$. For all $k, n \geq 1$, let $\pi_{k,n} \overset{\text{def}}{=} \emptyset^k \cdot (\{p\} \cdot \emptyset)^n \cdot \emptyset^\omega$ and $\mathcal{L}_{k,n} \overset{\text{def}}{=} \{\pi_{1,n}, \pi_{k,n}\}$. We denote by $\mathcal{L}_2$ the set of traces over $\mathsf{AP} \times [1, 2]$ which are 2-codes of the sets $\mathcal{L}_{k,n}$ for $k > 1$ and $n \geq 1$. Clearly, $\mathcal{L}_2$ is not regular. Let $\theta(x)$ be a one-variable quantifier-free $\mathsf{HyperLTL}$ formula capturing the traces $\pi_{k,n}$ for $k, n \geq 1$. We define a $\mathsf{HyperLTL}_S$ formula ψ_S and an $\mathsf{A\text{-}HyperLTL}$ formula ψ_A whose 2-language is $\mathcal{L}_2$:

$$\psi_S \overset{\text{def}}{=} \exists x_1. \forall x_2. \mathsf{X}p[x_1] \wedge \theta(x_1) \wedge \theta(x_2) \wedge \mathsf{G}_{\{p\}}(p[x_1] \leftrightarrow p[x_2]);$$

$$\psi_A \overset{\text{def}}{=} \exists x_1. \forall x_2. \forall x_3. \mathsf{E}. \mathsf{X}p[x_1] \wedge \theta(x_2) \wedge \theta(x_3) \wedge \mathsf{G}(p[x_2] \leftrightarrow p[x_3]). \quad ◀$$

By Propositions 18 and 19, and since $\mathsf{A\text{-}HyperLTL}$, $\mathsf{HyperLTL}_S$, and $\mathsf{HyperLTL}_C$ are subsumed by H_μ (Corollary 16), we obtain the following result.

▶ **Corollary 20.** *A-HyperLTL, HyperLTL$_S$, HyperLTL$_C$, and H$_\mu$ are not subsumed by $\mathsf{S1S[E]}$.*

5 Conclusions

Two interesting questions are left open. The first concerns the expressiveness of $\mathsf{HyperLTL}_C$ versus A-HyperLTL and $\mathsf{HyperLTL}_S$. We have shown that $\mathsf{HyperLTL}_C$ is not subsumed by A-HyperLTL or $\mathsf{HyperLTL}_S$. We conjecture that the converse holds too. The intuition is that (unlike $\mathsf{HyperLTL}_C$) A-HyperLTL and $\mathsf{HyperLTL}_S$ implicitly allow a restricted form of monadic second-order quantification. In particular, we conjecture that the hyperproperty characterizing the sets consisting of stuttering-equivalent traces, which can be easily expressed both in A-HyperLTL and $\mathsf{HyperLTL}_S$, cannot be captured by $\mathsf{HyperLTL}_C$.

The second question is whether S1S[E] is subsumed or not by H_μ. It is known that contrary to S1S[E] and FO[<,E], HyperLTL cannot express requirements which relate at some point an unbounded number of traces [5]. The main reason is that – differently from S1S[E] and FO[<,E] – quantifiers in HyperLTL only refer to the initial positions of the traces. Since in H_μ the semantics of quantifiers is the same as HyperLTL, we conjecture that the inexpressiveness result for HyperLTL in [5] can be extended to H_μ as well. This would imply together with the results of Corollary 20 that S1S[E] and H_μ are expressively incomparable and that so are FO[<,E] and H_μ.

Future work also includes an expressive comparison with Hypertrace Logic [1], a logical framework recently introduced which extends FO[<] with quantification over traces to express sequential information flow-properties. Also, we plan to study other extensions of temporal logic for asynchronous hyperproperties, in particular for recursive programs [20] and for multi-agent systems [3].

References

1 E. Bartocci, T. Ferrère, T.A. Henzinger, D. Nickovic, and A.O. da Costa. Flavors of Sequential Information Flow. In *Proc. 23rd VMCAI*, volume 13182 of *LNCS*, pages 1–19. Springer, 2022. doi:10.1007/978-3-030-94583-1_1.

2 J. Baumeister, N. Coenen, B. Bonakdarpour, B. Finkbeiner, and C. Sánchez. A Temporal Logic for Asynchronous Hyperproperties. In *Proc. 33rd CAV*, LNCS 12759, pages 694–717. Springer, 2021. doi:10.1007/978-3-030-81685-8_33.

3 R. Beutner and B. Finkbeiner. A Logic for Hyperproperties in Multi-Agent Systems. *CoRR*, abs/2203.07283, 2022. doi:10.48550/arXiv.2203.07283.

4 B. Bonakdarpour, C. Sánchez, and G. Schneider. Monitoring Hyperproperties by Combining Static Analysis and Runtime Verification. In *Proc. 8th ISoLA*, LNCS 11245, pages 8–27. Springer, 2018. doi:10.1007/978-3-030-03421-4_2.

5 L. Bozzelli, B. Maubert, and S. Pinchinat. Unifying Hyper and Epistemic Temporal Logics. In *Proc. 18th FoSSaCS*, LNCS 9034, pages 167–182. Springer, 2015. doi:10.1007/978-3-662-46678-0_11.

6 L. Bozzelli, A. Peron, and C. Sánchez. Asynchronous Extensions of HyperLTL. In *Proc. 36th LICS*, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470583.

7 L. Bozzelli, A. Peron, and C. Sánchez. Expressiveness and Decidability of Temporal Logics for Asynchronous Hyperproperties. *CoRR*, abs/2207.02956, 2022. doi:10.48550/arXiv.2207.02956.

8 M.R. Clarkson, B. Finkbeiner, M. Koleini, K.K. Micinski, M.N. Rabe, and C. Sánchez. Temporal Logics for Hyperproperties. In *Proc. 3rd POST*, LNCS 8414, pages 265–284. Springer, 2014. doi:10.1007/978-3-642-54792-8_15.

9 M.R. Clarkson and F.B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010. doi:10.3233/JCS-2009-0393.

10 N. Coenen, B. Finkbeiner, C. Hahn, and J. Hofmann. The hierarchy of hyperlogics. In *Proc. 34th LICS*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785713.

11 R. Dimitrova, B. Finkbeiner, M. Kovács, M.N. Rabe, and H. Seidl. Model Checking Information Flow in Reactive Systems. In *Proc. 13th VMCAI*, LNCS 7148, pages 169–185. Springer, 2012. doi:10.1007/978-3-642-27940-9_12.

12 E.A. Emerson and J.Y. Halpern. “Sometimes” and “Not Never” revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986. doi:10.1145/4904.4999.

13 B. Finkbeiner and C. Hahn. Deciding Hyperproperties. In *Proc. 27th CONCUR*, LIPIcs 59, pages 13:1–13:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.13.

14 B. Finkbeiner, C. Hahn, P. Lukert, M. Stenger, and L. Tentrup. Synthesis from hyperproperties. *Acta Informatica*, 57(1-2):137–163, 2020. doi:10.1007/s00236-019-00358-2.

15 B. Finkbeiner, M.N. Rabe, and C. Sánchez. Algorithms for Model Checking HyperLTL and HyperCTL*. In *Proc. 27th CAV Part I*, LNCS 9206, pages 30–48. Springer, 2015. doi:10.1007/978-3-319-21690-4_3.

16 B. Finkbeiner and M. Zimmermann. The first-order logic of hyperproperties. In *Proc. 34th STACS*, LIPIcs 66, pages 30:1–30:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.30.

17 M.J. Fischer and R.E. Ladner. Propositional Dynamic Logic of Regular Programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979. doi:10.1016/0022-0000(79)90046-1.

18 J.A. Goguen and J. Meseguer. Security Policies and Security Models. In *IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society, 1982. doi:10.1109/SP.1982.10014.

19 J.O. Gutsfeld, A. Meier, C. Ohrem, and J. Virtema. Temporal Team Semantics Revisited. *CoRR*, abs/2110.12699, 2021. doi:10.48550/arXiv.2110.12699.

20 J.O. Gutsfeld, M. Müller-Olm, and C. Ohrem. Deciding Asynchronous Hyperproperties for Recursive Programs. *CoRR*, abs/2201.12859, 2022. doi:10.48550/arXiv.2201.12859.

21 J.Oliver. Gutsfeld, M. Müller-Olm, and C. Ohrem. Propositional dynamic logic for hyperproperties. In *Proc. 31st CONCUR*, LIPIcs 171, pages 50:1–50:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.50.

22 J.Oliver. Gutsfeld, M. Müller-Olm, and C. Ohrem. Automata and fixpoints for asynchronous hyperproperties. *Proc. ACM Program. Lang.*, 4(POPL), 2021. doi:10.1145/3434319.

23 D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986. doi:10.1145/4904.4993.

24 A. Krebs, A. Meier, J. Virtema, and M. Zimmermann. Team Semantics for the Specification and Verification of Hyperproperties. In *Proc. 43rd MFCS*, LIPIcs 117, pages 10:1–10:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.10.

25 M. Lück. On the complexity of linear temporal logic with team semantics. *Theor. Comput. Sci.*, 837:1–25, 2020. doi:10.1016/j.tcs.2020.04.019.

26 J. McLean. A General Theory of Composition for a Class of “Possibilistic” Properties. *IEEE Trans. Software Eng.*, 22(1):53–67, 1996. doi:10.1109/32.481534.

27 A. Pnueli. The Temporal Logic of Programs. In *Proc. 18th FOCS*, pages 46–57. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.32.

28 M.N. Rabe. *A temporal logic approach to information-flow control.* PhD thesis, Saarland University, 2016.

29 A.P. Sistla, M.Y. Vardi, and P. Wolper. The Complementation Problem for Büchi Automata with Applications to Temporal Logic. *Theoretical Computer Science*, 49:217–237, 1987. doi:10.1016/0304-3975(87)90008-9.

30 M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Inf. Comput.*, 115(1):1–37, 1994. doi:10.1006/inco.1994.1092.

31 J. Virtema, J. Hofmann, B. Finkbeiner, J. Kontinen, and F. Yang. Linear-Time Temporal Logic with Team Semantics: Expressivity and Complexity. In *Proc. 41st IARCS FSTTCS*, LIPIcs 213, pages 52:1–52:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.52.

32 S. Zdancewic and A.C. Myers. Observational Determinism for Concurrent Program Security. In *Proc. 16th IEEE CSFW-16*, pages 29–43. IEEE Computer Society, 2003. doi:10.1109/CSFW.2003.1212703.