Unifying Asynchronous Logics for Hyperproperties

Alberto Bombardelli
Fondazione Bruno Kessler, Trento, Italy

Laura Bozzelli
University of Napoli "Federico II", Italy

César Sánchez
IMDEA Software Institute, Madrid, Spain

Stefano Tonetta
Fondazione Bruno Kessler, Trento, Italy

Abstract

We introduce and investigate a powerful hyper logical framework in the linear-time setting that we call *generalized* HyperLTL *with stuttering and contexts* ($\mathsf{GHyperLTL}_{\mathsf{S+C}}$ for short). $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ unifies the asynchronous extensions of HyperLTL called $\mathsf{HyperLTL}_{\mathsf{S}}$ and $\mathsf{HyperLTL}_{\mathsf{C}}$, and the well-known extension KLTL of LTL with knowledge modalities under both the synchronous and asynchronous perfect recall semantics. As a main contribution, we identify a meaningful fragment of $\mathsf{GHyperLTL}_{\mathsf{S+C}}$, that we call *simple* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$, with a decidable model-checking problem, which is more expressive than HyperLTL and known fragments of asynchronous extensions of HyperLTL with a decidable model-checking problem. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ subsumes KLTL under the synchronous semantics and the one-agent fragment of KLTL under the asynchronous semantics and to the best of our knowledge, it represents the unique hyper logic with a decidable model-checking problem which can express powerful non-regular trace properties when interpreted on singleton sets of traces. We justify the relevance of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ by showing that it can express diagnosability properties, interesting classes of information-flow security policies, both in the synchronous and asynchronous settings, and bounded termination (more in general, global promptness in the style of Prompt LTL).

2012 ACM Subject Classification Theory of computation → Logic and verification

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

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.14

Related Version *Full Version*: `https://arxiv.org/abs/2404.16778` [4]

Funding *César Sánchez*: funded in part by PRODIGY Project (TED2021-132464B-I00) – funded by MCIN/AEI/10.13039/501100011033/ and the European Union NextGenerationEU/PRTR – by DECO Project (PID2022-138072OB-I00) – funded by MCIN/AEI/10.13039/501100011033 and by the ESF+ – and by a research grant from Nomadic Labs and the Tezos Foundation.

1 Introduction

Temporal logics [27] play a fundamental role in the formal verification of the dynamic behaviour of complex reactive systems. Classic *regular* temporal logics such as LTL, CTL, and CTL* [29, 13] are suited for the specification of *trace properties* which describe the ordering of events along individual execution traces of a system. In the last 15 years, a novel specification paradigm has been introduced that generalizes traditional regular trace properties by properties of sets of traces, the so called *hyperproperties* [10]. Hyperproperties relate distinct traces and are useful to formalize a wide range of properties of prime interest which go, in general, beyond regular properties and cannot be expressed in standard regular temporal logics. A relevant example concerns information-flow security policies like noninterference [18, 28] and observational determinism [36] which compare observations made by an external low-security agent along traces resulting from different values of not directly observable inputs. Other examples include bounded termination of programs, diagnosability of critical systems (which amounts to checking whether the available sensor information is sufficient to infer the presence of faults on the hidden behaviour of the system) [31, 5, 3], and epistemic properties describing the knowledge of agents in distributed systems [23, 33, 22].

In the context of model checking of finite-state reactive systems, many temporal logics for hyperproperties have been proposed [12, 9, 6, 30, 15, 11, 20] for which model checking is decidable, including HyperLTL [9], HyperCTL* [9], HyperQPTL [30, 11], and HyperPDL$-\Delta$ [20] which extend LTL, CTL*, QPTL [32], 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*: the temporal modalities are evaluated by a lockstepwise traversal of all the traces assigned to the quantified trace variables. Other approaches for the formalization of synchronous hyper logics are either based on hyper variants of monadic second-order logic over traces or trees [11], or the adoption of a *team semantics* for standard temporal logics, in particular, LTL [24, 26, 35]. For the first approach in the linear-time setting, we recall the logic S1S[E] [11] (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 relates the same time point on different traces. More recently, an extension of HyperLTL with second-order quantification over traces has been introduced [2] which allows to express common knowledge in multi-agent distributed systems. Like S1S[E], model checking of this extension of HyperLTL is highly undecidable [2].

Hyper logics supporting asynchronous features have been introduced recently [21, 1, 7]. These logics allow to relate traces at distinct time points which can be arbitrarily far from each other. Asynchronicity is ubiquitous in many real-world systems, for example, in multithreaded environments in which threads are not scheduled lockstepwise, and traces associated with distinct threads progress with different speed. Asynchronous hyperproperties are also useful in information-flow security and diagnosability settings where an observer cannot distinguish consecutive time points along an execution having the same observations. This requires to match asynchronously sequences of observations along distinct execution traces. The first systematic study of asynchronous hyperproperties was done by Gutsfeld et al. [21], who introduced the temporal fixpoint calculus H_μ and its automata-theoretic counterpart for expressing such properties in the linear-time setting.

More recently, three temporal logics [1, 7] which syntactically extend HyperLTL have been introduced for expressing asynchronous hyperproperties: *Asynchronous* HyperLTL (A-HyperLTL) [1] and *Stuttering* HyperLTL ($\mathsf{HyperLTL_S}$) [7], both useful for asynchronous security analysis, and *Context* HyperLTL ($\mathsf{HyperLTL_C}$) [7], useful for expressing hyper-bounded-time response requirements. The logic A-HyperLTL, which is expressively incomparable with both HyperLTL and $\mathsf{HyperLTL_S}$ [8], models asynchronicity by means of an additional quantification layer over the so called *trajectories* which control the relative speed at which traces progress by choosing at each instant which traces move and which traces stutter. On the other hand, the logic $\mathsf{HyperLTL_S}$ exploits relativized versions of the temporal modalities with respect to finite sets Γ of LTL formulas: 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. Finally, the logic $\mathsf{HyperLTL_C}$ is more expressive than HyperLTL and is not expressively subsumed by either A-HyperLTL or $\mathsf{HyperLTL_S}$ [8]. $\mathsf{HyperLTL_C}$ 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.

Note that the temporal modalities in $\mathsf{HyperLTL}_\mathsf{C}$ are evaluated by a lockstepwise traversal of the traces assigned to the variables in the current context, and unlike HyperLTL, the current time points of these traces from which the evaluation starts are in general different. It is known that these three syntactical extensions of HyperLTL are less expressive than H_μ [8] and like H_μ, model checking the respective quantifier alternation-free fragments are already undecidable [1, 7]. The works [1, 7] identify practical fragments of the logics A-HyperLTL and $\mathsf{HyperLTL}_\mathsf{S}$ with a decidable model checking problem. In particular, we recall the so called *simple fragment* of $\mathsf{HyperLTL}_\mathsf{S}$ [7], which is more expressive than HyperLTL [8] and can specify interesting security policies in both the asynchronous and synchronous settings.

Formalization of asynchronous hyperproperties in the *team semantics setting* following an approach similar to the *trajectory construct* of A-HyperLTL has been investigated in [19]. It is worth noting that unlike other hyper logics (including logics with team semantics) which only capture regular trace properties when interpreted on singleton sets of traces, the logics $\mathsf{HyperLTL}_\mathsf{C}$, A-HyperLTL, and H_μ can express non-regular trace properties [8].

Our contribution. Specifications in HyperLTL and in the known asynchronous extensions of HyperLTL, whose most expressive representative is H_μ [21], consist of a prefix of trace quantifiers followed by a quantifier-free formula which expresses temporal requirements on a fixed number of traces. Thus, these hyper logics lack mechanisms to relate directly an unbounded number of traces, which are required for example to express bounded termination or diagnosability properties [31, 5, 3]. This ability is partially supported by temporal logics with team semantics [24, 26, 35] and extensions of temporal logics with the knowledge modalities of epistemic logic [14], which relate computations whose histories are observationally equivalent for a given agent. In this paper, we introduce and investigate a hyper logical framework in the linear-time setting which unifies two known asynchronous extensions of HyperLTL and the well-known extension KLTL [23] of LTL with knowledge modalities under both the synchronous and asynchronous perfect recall semantics (where an agent remembers the whole sequence of its observations). The novel logic, that we call *generalized* HyperLTL *with stuttering and contexts* ($\mathsf{GHyperLTL}_{\mathsf{S+C}}$ for short), merges $\mathsf{HyperLTL}_\mathsf{S}$ and $\mathsf{HyperLTL}_\mathsf{C}$ and adds two new natural modeling facilities: past temporal modalities for asynchronous hyperproperties and general trace quantification where trace quantifiers can occur in the scope of temporal modalities. Past temporal modalities used in combination with context modalities provide a powerful mechanism to compare histories of computations at distinct time points. Moreover, unrestricted trace quantification allows to relate an unbounded number of traces.

As a main contribution, we identify a meaningful fragment of $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ with a decidable model-checking problem, that we call *simple* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$. This fragment is obtained from $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ by carefully imposing restrictions on the use of the stuttering and context modalities. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ allows quantification over arbitrary *pointed* traces (i.e., traces plus time points) in the style of FO[<,E] [16], it is more expressive than the simple fragment of $\mathsf{HyperLTL}_\mathsf{S}$ [7], and it is expressively incomparable with full $\mathsf{HyperLTL}_\mathsf{S}$ and S1S[E]. Moreover, this fragment subsumes both KLTL under the synchronous semantics and the one-agent fragment of KLTL under the asynchronous semantics. In fact, simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ can be seen as a very large fragment of $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ with a decidable model checking problem which (1) strictly subsumes HyperLTL and the simple fragment of $\mathsf{HyperLTL}_\mathsf{S}$, (2) is closed under Boolean connectives, and (3) allows an unrestricted nesting of temporal modalities. We justify the relevance of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ by showing that it can express diagnosability properties, interesting classes of information-flow security policies, both in the synchronous and asynchronous settings, and bounded termination (more in general, global promptness in the style of Prompt LTL [25]). To the best of our knowledge, simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ represents the unique hyper logic with a decidable model-checking problem which can express powerful non-regular trace properties when interpreted over singleton sets of traces.

2 Background

We denote by $\mathbb{N}$ the set of natural numbers. Given $i, j \in \mathbb{N}$, we write $[i, j]$ for the set of natural numbers h such that $i \leq h \leq j$, we use $[i, j)$ for the set $[i, j] \setminus \{j\}$, we use $(i, j]$ for the set $[i, j] \setminus \{i\}$, and $[i, \infty)$ for the set of natural numbers h such that $h \geq i$. 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, w^i is the suffix of w from position i, that is, the word $w(i)w(i+1)\ldots$, and $w[0, i]$ is the prefix of w that ends at position i.

We fix a finite set AP of atomic propositions. A *trace* is an infinite word over 2^{AP}, while a *finite trace* is a nonempty finite word over 2^{AP}. A *pointed trace* is a pair (σ, i) consisting of a trace σ and a position (timestamp) $i \in \mathbb{N}$ along σ.

Kripke structures. We define the dynamic behaviour of reactive systems by *Kripke structures* $\mathcal{K} = \langle S, S_0, E, Lab\rangle$ over a finite set AP of atomic propositions, where S is a set of states, $S_0 \subseteq S$ is the set of initial states, $E \subseteq S \times S$ is a transition relation which is total in the first argument (i.e., for each $s \in S$ there is $s' \in S$ with $(s, s') \in E$), and $Lab : S \to 2^{\mathsf{AP}}$ is a labeling map assigning to each state s the set of propositions holding at s. The Kripke structure $\mathcal{K}$ is finite if S is finite. A *path* π of $\mathcal{K}$ is an infinite word $\pi = s_0, s_1, \ldots$ over S such that $s_0 \in S_0$ and for all $i \geq 0$, $(s_i, s_{i+1}) \in E$. The path $\pi = s_0, s_1, \ldots$ induces the trace $Lab(s_0)Lab(s_1)\ldots$. A *trace of* $\mathcal{K}$ is a trace induced by some path of $\mathcal{K}$. We denote by $\mathcal{L}(\mathcal{K})$ the set of traces of $\mathcal{K}$. A *finite path* of $\mathcal{K}$ is a non-empty infix of some path of $\mathcal{K}$. We also consider *fair finite Kripke structures* $(\mathcal{K}, F)$, that is, finite Kripke structures $\mathcal{K}$ equipped with a subset F of $\mathcal{K}$-states. A path π of $\mathcal{K}$ is *F-fair* if π visits infinitely many times some state in F. We denote by $\mathcal{L}(\mathcal{K}, F)$ the set of traces of $\mathcal{K}$ associated with the F-fair paths of $\mathcal{K}$.

Standard LTL with past (PLTL for short) [29]. Formulas ψ of PLTL over the given finite set AP of atomic propositions are defined by the following grammar:

$$\psi ::= \top \mid p \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \mathbf{Y}\psi \mid \psi\,\mathbf{U}\,\psi \mid \psi\,\mathbf{S}\,\psi$$

where $p \in \mathsf{AP}$, $\mathbf{X}$ and $\mathbf{U}$ are the *next* and *until* temporal modalities respectively, and $\mathbf{Y}$ (*previous* or *yesterday*) and $\mathbf{S}$ (*since*) are their past counterparts. LTL is the fragment of PLTL that does not contain the past temporal modalities $\mathbf{Y}$ and $\mathbf{S}$. We also use the following abbreviations: $\mathbf{F}\psi := \top\,\mathbf{U}\,\psi$ (*eventually*), $\mathbf{O}\psi := \top\,\mathbf{S}\,\psi$ (*past eventually* or *once*), and their duals $\mathbf{G}\psi := \neg\,\mathbf{F}\,\neg\psi$ (*always*) and $\mathbf{H}\psi := \neg\,\mathbf{O}\,\neg\psi$ (*past always* or *historically*).

The semantics of PLTL is defined over pointed traces (σ, i). The satisfaction relation $(\sigma, i) \models \psi$, that defines whether formula ψ holds at position i along σ, is inductively defined as follows (we omit the semantics for the Boolean connectives which is standard):

$$\begin{array}{ll}
(\sigma, i) \models p & \Leftrightarrow p \in \sigma(i) \\
(\sigma, i) \models \mathbf{X}\psi & \Leftrightarrow (\sigma, i+1) \models \psi \\
(\sigma, i) \models \mathbf{Y}\psi & \Leftrightarrow i > 0 \text{ and } (\sigma, i-1) \models \psi \\
(\sigma, i) \models \psi_1\,\mathbf{U}\,\psi_2 & \Leftrightarrow \text{for some } j \geq i : (\sigma, j) \models \psi_2 \text{ and } (\sigma, k) \models \psi_1 \text{ for all } i \leq k < j \\
(\sigma, i) \models \psi_1\,\mathbf{S}\,\psi_2 & \Leftrightarrow \text{for some } j \leq i : (\sigma, j) \models \psi_2 \text{ and } (\sigma, k) \models \psi_1 \text{ for all } j < k \leq i
\end{array}$$

A trace σ is a *model* of ψ, written $\sigma \models \psi$, whenever $(\sigma, 0) \models \psi$.

The logic HyperLTL [9]. The syntax of HyperLTL formulas φ over the given finite set AP of atomic propositions and a finite set VAR of trace variables is as follows:

$$\varphi := \exists x.\, \varphi \mid \forall x.\, \varphi \mid \psi \qquad \psi := \top \mid p[x] \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\, \mathbf{U}\, \psi$$

where $p \in \mathsf{AP}$, $x \in \mathsf{VAR}$, and $\exists x$ and $\forall x$ are the *hyper* existential and universal trace quantifiers for variable x, respectively, which allow relating different traces of the given set of traces. Note that a HyperLTL formula consists of a prefix of traces quantifiers followed by a quantifier-free formula, where the latter corresponds to an LTL formula whose atomic propositions p are replaced with x-relativized versions $p[x]$. Intuitively, $p[x]$ asserts that p holds at the pointed trace assigned to variable x. A *sentence* is a formula where each relativized proposition $p[x]$ occurs in the scope of trace quantifier $\exists x$ or $\forall x$.

In order to define the semantics of HyperLTL, we need additional definitions. The *successor* $\mathit{succ}(\sigma, i)$ of a pointed trace (σ, i) is the pointed trace $(\sigma, i+1)$, which captures the standard local successor of a position along a trace.

Given a set of traces $\mathcal{L}$, a (*pointed*) *trace assignment* over $\mathcal{L}$ is a partial mapping $\Pi : \mathsf{VAR} \to \mathcal{L} \times \mathbb{N}$ assigning to each trace variable x – where Π is defined – a pointed trace (σ, i) such that $\sigma \in \mathcal{L}$. We use $\mathit{Dom}(\Pi)$ to refer to the trace variables for which Π is defined. The *successor* $\mathit{succ}(\Pi)$ *of* Π is the trace assignment over $\mathcal{L}$ having domain $\mathit{Dom}(\Pi)$ such that $\mathit{succ}(\Pi)(x) = \mathit{succ}(\Pi(x))$ for each $x \in \mathit{Dom}(\Pi)$. For each $i \geq 0$, we use succ^i for the function obtained by i applications of the function succ: $\mathit{succ}^0(\Pi) := \Pi$ and $\mathit{succ}^{i+1}(\Pi) := \mathit{succ}(\mathit{succ}^i(\Pi))$.

Given $x \in \mathsf{VAR}$ and a pointed trace (σ, i) with $\sigma \in \mathcal{L}$, we denote by $\Pi[x \mapsto (\sigma, i)]$ the trace assignment that is identical to Π besides for x, which is mapped to (σ, i).

Given a formula φ, a set of traces $\mathcal{L}$, and a trace assignment Π over $\mathcal{L}$ such that $\mathit{Dom}(\Pi)$ contains all the trace variables occurring free in φ, the satisfaction relation $\Pi \models_{\mathcal{L}} \varphi$ is inductively defined as follows (we again omit the semantics of the Boolean connectives):

$$\begin{array}{ll}
\Pi \models_{\mathcal{L}} p[x] & \Leftrightarrow \Pi(x) = (\sigma, i) \text{ and } p \in \sigma(i) \\
\Pi \models_{\mathcal{L}} \exists x.\, \varphi & \Leftrightarrow \text{ for some } \sigma \in \mathcal{L},\ \Pi[x \mapsto (\sigma, 0)] \models_{\mathcal{L}} \varphi \\
\Pi \models_{\mathcal{L}} \forall x.\, \varphi & \Leftrightarrow \text{ for all } \sigma \in \mathcal{L},\ \Pi[x \mapsto (\sigma, 0)] \models_{\mathcal{L}} \varphi \\
\Pi \models_{\mathcal{L}} \mathbf{X}\psi & \Leftrightarrow \mathit{succ}(\Pi) \models \psi \\
\Pi \models_{\mathcal{L}} \psi_1\, \mathbf{U}\, \psi_2 & \Leftrightarrow \text{for some } i \geq 0 :\ \mathit{succ}^i(\Pi) \models_{\mathcal{L}} \psi_2 \text{ and } \mathit{succ}^j(\Pi) \models_{\mathcal{L}} \psi_1 \text{ for all } 0 \leq j < i
\end{array}$$

Note that trace quantification ranges over *initial* pointed traces $(\sigma, 0)$ over $\mathcal{L}$ (the timestamp is 0). As an example, the sentence $\forall x_1.\, \forall x_2.\, \bigwedge_{p \in \mathsf{AP}} \mathbf{G}(p[x_1] \leftrightarrow p[x_2])$ captures the sets of traces which are singletons.

For a sentence φ and a set of traces $\mathcal{L}$, $\mathcal{L}$ is a *model* of φ, written $\mathcal{L} \models \varphi$, if $\Pi_\emptyset \models_{\mathcal{L}} \varphi$ where $\Pi_\emptyset$ is the trace assignment with empty domain.

3 Unifying Framework for Asynchronous Extensions of HyperLTL

In this section, we introduce a novel logical framework for specifying both asynchronous and synchronous linear-time hyperproperties which unifies two known more expressive extensions of HyperLTL [9], namely *Stuttering* HyperLTL ($\mathsf{HyperLTL}_\mathsf{S}$ for short) [7] and *Context* HyperLTL ($\mathsf{HyperLTL}_\mathsf{C}$ for short) [7]. The proposed hyper logic, that we call *generalized* HyperLTL *with stuttering and contexts* ($\mathsf{GHyperLTL}_{\mathsf{S+C}}$ for short), merges $\mathsf{HyperLTL}_\mathsf{S}$ and $\mathsf{HyperLTL}_\mathsf{C}$ and adds two new features: past temporal modalities for asynchronous/synchronous hyperproperties and general trace quantification where trace quantifiers can occur in the scope of temporal modalities. Since model checking of the logics $\mathsf{HyperLTL}_\mathsf{S}$ and $\mathsf{HyperLTL}_\mathsf{C}$ is already undecidable [7], we also consider a meaningful fragment of $\mathsf{GHyperLTL_{S+C}}$ which is strictly more expressive than the known *simple fragment* of $\mathsf{HyperLTL_S}$ [7]. Our fragment is able to express relevant classes of hyperproperties and, as we show in Section 4, its model checking problem is decidable.

3.1 PLTL-Relativized Stuttering and Context Modalities

Classically, a trace is stutter-free if there are no consecutive positions having the same propositional valuation unless the valuation is repeated ad-infinitum. We can associate to each trace a unique stutter-free trace by removing "redundant" positions. The logic $\mathsf{HyperLTL_S}$ [7] generalizes these notions with respect to the pointwise evaluation of a finite set of LTL formulas. Here, we consider LTL with past (PLTL).

▶ **Definition 3.1** (PLTL stutter factorization [7]). *Let Γ be a finite set of* PLTL *formulas and σ a trace. The* Γ-stutter factorization of σ *is the unique increasing sequence of positions $\{i_k\}_{k\in[0,m_\infty]}$ for some $m_\infty \in \mathbb{N}\cup\{\infty\}$ such that the following holds for all $j < m_\infty$:*

- *$i_0 = 0$ and $i_j < i_{j+1}$;*
- *for each $\theta \in \Gamma$, the truth value of θ along the segment $[i_j, i_{j+1})$ does not change, that is, for all $h, k \in [i_j, i_{j+1})$, $(\sigma, h) \models \theta$ iff $(\sigma, k) \models \theta$, and the same holds for the infinite segment $[m_\infty, \infty]$ in case $m_\infty \neq \infty$;*
- *the truth value of some formula in Γ changes along adjacent segments, that is, for some $\theta \in \Gamma$ (depending on j), $(\sigma, i_j) \models \theta$ iff $(\sigma, i_{j+1}) \not\models \theta$.*

Thus, the Γ-stutter factorization $\{i_k\}_{k\in[0,m_\infty]}$ of σ partitions the trace in adjacent non-empty segments such that the valuation of formulas in Γ does not change within a segment, and changes in moving from a segment to the adjacent ones. This factorization induces in a natural way a trace obtained by selecting the first positions of the finite segments and all the positions of the unique tail infinite segment, if any. These positions form an infinite increasing sequence $\{\ell_k\}_{k\in\mathbb{N}}$ called (Γ,ω)-*stutter factorization* of σ, where:

$$\ell_0, \ell_1, \ldots := \begin{cases} i_0, i_1, \ldots & \text{if } m_\infty = \infty \\ i_0, i_1, \ldots, i_{m_\infty}, i_{m_\infty}+1, \ldots & \text{otherwise} \end{cases}$$

The Γ-*stutter trace* $\mathit{stfr}_\Gamma(\sigma)$ *of* σ (see [7]) is defined as follows: $\mathit{stfr}_\Gamma(\sigma) := \sigma(\ell_0)\sigma(\ell_1)\ldots$. Note that for $\Gamma = \emptyset$, $\mathit{stfr}_\Gamma(\sigma) = \sigma$. A trace σ is Γ-*stutter free* if it coincides with its Γ-stutter trace, i.e. $\mathit{stfr}_\Gamma(\sigma) = \sigma$.

As an example, assume that $\mathsf{AP} = \{p, q, r\}$ and let $\Gamma = \{p\,\mathbf{U}\,q\}$. Given $h, k \geq 1$, let $\sigma_{h,k}$ be the trace $\sigma_{h,k} = p^h q^k r^\omega$. These traces have the same Γ-stutter trace given by pr^ω.

The semantics of the Γ-relativized temporal modalities in $\mathsf{HyperLTL_S}$ is based on the notion of Γ-*successor* $\mathit{succ}_\Gamma(\sigma, i)$ of a pointed trace (σ, i) [7]: $\mathit{succ}_\Gamma(\sigma, i)$ is the pointed trace (σ, ℓ) where ℓ is the smallest position ℓ_j in the (Γ,ω)-stutter factorization $\{\ell_k\}_{k\in\mathbb{N}}$ of σ which is greater than i. Note that for $\Gamma = \emptyset$, $\mathit{succ}_\emptyset(\sigma, i) = \mathit{succ}(\sigma, i) = (\sigma, i+1)$. Hence, $\emptyset$-relativized temporal modalities in $\mathsf{HyperLTL_S}$ correspond to the temporal modalities of HyperLTL.

In this paper we extend $\mathsf{HyperLTL_S}$ with past temporal modalities, so that we introduce the past counterpart of the successor function. The Γ-*predecessor* $\mathit{pred}_\Gamma(\sigma, i)$ of a pointed trace (σ, i) is undefined if $i = 0$ (written $\mathit{pred}_\Gamma(\sigma, i) = \mathtt{und}$); otherwise, $\mathit{pred}_\Gamma(\sigma, i)$ is the pointed trace (σ, ℓ) where ℓ is the greatest position ℓ_j in the (Γ,ω)-stutter factorization $\{\ell_k\}_{k\in\mathbb{N}}$ of σ which is smaller than i (since $\ell_0 = 0$ such an ℓ_j exists). Note that for $\Gamma = \emptyset$, $\mathit{pred}_\emptyset(\sigma, i)$ captures the standard local predecessor of a position along a trace.

Successors and predecessors of trace assignments. We now define a generalization of the successor $\mathit{succ}(\Pi)$ of a trace assignment Π in HyperLTL. This generalization is based on the notion of Γ-successor $\mathit{succ}_\Gamma(\sigma, i)$ of a pointed trace (σ, i) and also takes into account the context modalities $\langle C \rangle$ of HyperLTL$_\mathsf{C}$ [7], where a *context* C is a non-empty subset of VAR. Intuitively, modality $\langle C \rangle$ allows reasoning over a subset of the traces assigned to the variables in the formula, by restricting the temporal progress to those traces.

Formally, let Π be a trace assignment over some set of traces $\mathcal{L}$, Γ be a finite set of PLTL formulas, and C be a context. The (Γ, C)*-successor of* Π, denoted by $\mathit{succ}_{(\Gamma,C)}(\Pi)$, is the trace assignment over $\mathcal{L}$ having domain $\mathit{Dom}(\Pi)$, and defined as follows for each $x \in \mathit{Dom}(\Pi)$:

$$\mathit{succ}_{(\Gamma,C)}(\Pi)(x) := \begin{cases} \mathit{succ}_\Gamma(\Pi(x)) & \text{if } x \in C \\ \Pi(x) & \text{otherwise} \end{cases}$$

Note that $\mathit{succ}_{(\emptyset,\mathsf{VAR})}(\Pi) = \mathit{succ}(\Pi)$. Moreover, we define the (Γ, C)*-predecessor* $\mathit{pred}_{(\Gamma,C)}(\Pi)$ *of* Π as follows: $\mathit{pred}_{(\Gamma,C)}(\Pi)$ is *undefined*, written $\mathit{pred}_{(\Gamma,C)}(\Pi) = \mathtt{und}$, if there is $x \in \mathit{Dom}(\Pi)$ such that $\mathit{pred}_\Gamma(\Pi(x)) = \mathtt{und}$. Otherwise, $\mathit{pred}_{(\Gamma,C)}(\Pi)$ is the trace assignment over $\mathcal{L}$ having domain $\mathit{Dom}(\Pi)$, and defined as follows for each $x \in \mathit{Dom}(\Pi)$:

$$\mathit{pred}_{(\Gamma,C)}(\Pi)(x) := \begin{cases} \mathit{pred}_\Gamma(\Pi(x)) & \text{if } x \in C \\ \Pi(x) & \text{otherwise} \end{cases}$$

Finally, for each $i \geq 0$, we define the i^{th} application $\mathit{succ}^i_{(\Gamma,C)}$ of $\mathit{succ}_{(\Gamma,C)}$ and the i^{th} application $\mathit{pred}^i_{(\Gamma,C)}$ of $\mathit{pred}_{(\Gamma,C)}$ as follows, where $\mathit{pred}_{(\Gamma,C)}(\mathtt{und}) := \mathtt{und}$:

- $\mathit{succ}^0_{(\Gamma,C)}(\Pi) := \Pi$ and $\mathit{succ}^{i+1}_{(\Gamma,C)}(\Pi) := \mathit{succ}_{(\Gamma,C)}(\mathit{succ}^i_{(\Gamma,C)}(\Pi))$.
- $\mathit{pred}^0_{(\Gamma,C)}(\Pi) := \Pi$ and $\mathit{pred}^{i+1}_{(\Gamma,C)}(\Pi) := \mathit{pred}_{(\Gamma,C)}(\mathit{pred}^i_{(\Gamma,C)}(\Pi))$.

3.2 Generalized HyperLTL with Stuttering and Contexts

We introduce now the novel logic GHyperLTL$_{\mathsf{S+C}}$. GHyperLTL$_{\mathsf{S+C}}$ formulas φ over AP and a finite set VAR of trace variables are defined by the following syntax:

$$\varphi := \top \mid p[x] \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\, \varphi \mid \langle C \rangle \varphi \mid \mathbf{X}_\Gamma \varphi \mid \mathbf{Y}_\Gamma \varphi \mid \varphi\, \mathbf{U}_\Gamma\, \varphi \mid \varphi\, \mathbf{S}_\Gamma\, \varphi$$

where $p \in \mathsf{AP}$, $x \in \mathsf{VAR}$, $\langle C \rangle$ is the context modality with $\emptyset \neq C \subseteq \mathsf{VAR}$, Γ is a finite set of PLTL formulas, and $\mathbf{X}_\Gamma$, $\mathbf{Y}_\Gamma$, $\mathbf{U}_\Gamma$ and $\mathbf{S}_\Gamma$ are the stutter-relativized versions of the PLTL temporal modalities. Intuitively, the context modality $\langle C \rangle$ restricts the evaluation of the temporal modalities to the traces associated with the variables in C, while the temporal modalities $\mathbf{X}_\Gamma$, $\mathbf{Y}_\Gamma$, $\mathbf{U}_\Gamma$ and $\mathbf{S}_\Gamma$ are evaluated by a lockstepwise traversal of the Γ-stutter traces associated to the traces assigned to the variables in the current context C. Note that the hyper universal quantifier $\forall x$ can be introduced as an abbreviation: $\forall x.\, \varphi \equiv \neg \exists x.\, \neg \varphi$. For a variable x, we write $\langle x \rangle$ instead of $\langle \{x\} \rangle$. Moreover, we write $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{S}$ instead of $\mathbf{X}_\emptyset$, $\mathbf{Y}_\emptyset$, $\mathbf{U}_\emptyset$ and $\mathbf{S}_\emptyset$, respectively. Furthermore, for a PLTL formula ψ and a variable x, $\psi[x]$ is the formula obtained from ψ by replacing each proposition p with its x-version $p[x]$. A *sentence* is a formula where each relativized proposition $p[x]$ occurs in the scope of trace quantifier $\exists x$ or $\forall x$, and each temporal modality occurs in the scope of a trace quantifier.

The known logics HyperLTL$_\mathsf{S}$, HyperLTL$_\mathsf{C}$, and simple HyperLTL$_\mathsf{S}$. A formula φ of GHyperLTL$_{\mathsf{S+C}}$ is in *prenex* form if it is of the form $\mathsf{Q}_1 x_1. \ldots \mathsf{Q}_n x_n.\, \psi$ where ψ is quantifier-free and $\mathsf{Q}_i \in \{\exists, \forall\}$ for all $i \in [1, n]$. The logics HyperLTL$_\mathsf{S}$ and HyperLTL$_\mathsf{C}$ introduced in [7] correspond to syntactical fragments of $\mathsf{GHyperLTL_{S+C}}$ where the formulas are in prenex form and past temporal modalities are not used. Moreover, in $\mathsf{HyperLTL_S}$, the context modalities are not allowed, while in $\mathsf{HyperLTL_C}$, the subscript Γ of every temporal modality must be the empty set. Note that in HyperLTL [9], both the context modalities and the temporal modalities where the subscript Γ is not empty are disallowed. Finally, we recall the *simple* fragment of $\mathsf{HyperLTL_S}$ [7], which is more expressive than HyperLTL and is parameterized by a finite set Γ of LTL formulas. The *quantifier-free* formulas of simple $\mathsf{HyperLTL_S}$ for the parameter Γ are defined as Boolean combinations of formulas of the form $\psi[x]$, where ψ is an LTL formula, and formulas ψ_Γ defined by the following grammar:

$$\psi_\Gamma := \top \mid p[x] \mid \neg\psi_\Gamma \mid \psi_\Gamma \vee \psi_\Gamma \mid \mathbf{X}_\Gamma \psi_\Gamma \mid \psi_\Gamma \,\mathbf{U}_\Gamma\, \psi_\Gamma$$

Semantics of $\mathsf{GHyperLTL_{S+C}}$. Given a formula φ, a set of traces $\mathcal{L}$, a trace assignment Π over $\mathcal{L}$ such that $Dom(\Pi)$ contains all the trace variables occurring free in φ, and a context $C \subseteq \mathsf{VAR}$, the satisfaction relation $(\Pi, C) \models_\mathcal{L} \varphi$, meaning that the assignment Π over $\mathcal{L}$ satisfies φ under the context C, is inductively defined as follows (we again omit the semantics of the Boolean connectives):

$$\begin{array}{ll}
(\Pi, C) \models_\mathcal{L} p[x] & \Leftrightarrow \Pi(x) = (\sigma, i) \text{ and } p \in \sigma(i) \\
(\Pi, C) \models_\mathcal{L} \exists x.\, \varphi & \Leftrightarrow \text{ for some } \sigma \in \mathcal{L},\ (\Pi[x \mapsto (\sigma, 0)], C) \models_\mathcal{L} \varphi \\
(\Pi, C) \models_\mathcal{L} \langle C' \rangle \varphi & \Leftrightarrow (\Pi, C') \models_\mathcal{L} \varphi \\
(\Pi, C) \models_\mathcal{L} \mathbf{X}_\Gamma \varphi & \Leftrightarrow (succ_{(\Gamma,C)}(\Pi), C) \models \varphi \\
(\Pi, C) \models_\mathcal{L} \mathbf{Y}_\Gamma \varphi & \Leftrightarrow pred_{(\Gamma,C)}(\Pi) \neq \mathtt{und} \text{ and } (pred_{(\Gamma,C)}(\Pi), C) \models_\mathcal{L} \varphi \\
(\Pi, C) \models_\mathcal{L} \varphi_1 \,\mathbf{U}_\Gamma\, \varphi_2 & \Leftrightarrow \text{for some } i \geq 0\text{: } (succ^i_{(\Gamma,C)}(\Pi), C) \models_\mathcal{L} \varphi_2 \text{ and} \\
& \quad (succ^j_{(\Gamma,C)}(\Pi), C) \models_\mathcal{L} \varphi_1 \text{ for all } 0 \leq j < i, \\
(\Pi, C) \models_\mathcal{L} \varphi_1 \,\mathbf{S}_\Gamma\, \varphi_2 & \Leftrightarrow \text{for some } i \geq 0 \text{ such that } pred^i_{(\Gamma,C)}(\Pi) \neq \mathtt{und}\text{: } (pred^i_{(\Gamma,C)}(\Pi), C) \models_\mathcal{L} \varphi_2 \\
& \quad \text{and } (pred^j_{(\Gamma,C)}(\Pi), C) \models_\mathcal{L} \varphi_1 \text{ for all } 0 \leq j < i
\end{array}$$

For a sentence φ and a set of traces $\mathcal{L}$, $\mathcal{L}$ is a *model* of φ, written $\mathcal{L} \models \varphi$, if $(\Pi_\emptyset, \mathsf{VAR}) \models_\mathcal{L} \varphi$ where $\Pi_\emptyset$ is the trace assignment with empty domain.

Fair model checking and standard model checking. For a fragment $\mathcal{F}$ of $\mathsf{GHyperLTL_{S+C}}$, the *fair model checking problem* for $\mathcal{F}$ consists on deciding, given a fair finite Kripke structure $(\mathcal{K}, F)$ and a sentence φ of $\mathcal{F}$, whether $\mathcal{L}(\mathcal{K}, F) \models \varphi$. The previous problem is simply called *model checking problem* whenever F coincides with the set of $\mathcal{K}$-states. We consider fair model checking just for technical convenience. For the decidable fragment of $\mathsf{GHyperLTL_{S+C}}$ introduced in Section 3.3, we will obtain the same complexity bounds for both fair model checking and standard model checking (see Section 4).

3.3 The Simple Fragment of $\mathsf{GHyperLTL_{S+C}}$

We introduce now a fragment of $\mathsf{GHyperLTL_{S+C}}$, that we call *simple* $\mathsf{GHyperLTL_{S+C}}$, which syntactically subsumes the simple fragment of $\mathsf{HyperLTL_S}$ [7].

In order to define the syntax of simple $\mathsf{GHyperLTL_{S+C}}$, we first consider some shorthands, obtained by a restricted use of the context modalities. The *pointed existential quantifier* $\exists^\mathsf{P} x$ and the *pointed universal quantifier* $\forall^\mathsf{P} x$ are defined as follows: $\exists^\mathsf{P} x.\, \varphi := \exists x.\, \langle x \rangle\, \mathbf{F}\, \langle \mathsf{VAR} \rangle \varphi$ and $\forall^\mathsf{P} x.\, \varphi ::= \neg \exists^\mathsf{P} x.\, \neg\varphi$. Thus the pointed quantifiers quantify on arbitrary pointed traces over the given set of traces and set the global context for the given operand. Formally, $(\Pi, C) \models_\mathcal{L} \exists^\mathsf{P} x.\, \varphi$ if for some pointed trace (σ, i) with $\sigma \in \mathcal{L}$, $(\Pi[x \mapsto (\sigma, i)], \mathsf{VAR}) \models_\mathcal{L} \varphi$. For example, the sentence $\exists x_1.\, \exists^\mathsf{P} x_2.\, \big(\bigwedge_{p \in \mathsf{AP}} \mathbf{G}(p[x_1] \leftrightarrow p[x_2]) \wedge \langle x_2 \rangle \mathbf{Y} \top\big)$ asserts that there are two traces σ_1 and σ_2 in the given model s.t. some *proper* suffix of σ_2 coincides with σ_1.

Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ is parameterized by a finite set Γ of PLTL formulas. The set of formulas φ_Γ in the Γ-fragment is defined as follows:

$$\varphi_\Gamma := \top \mid \langle x\rangle\psi[x] \mid \neg\varphi_\Gamma \mid \varphi_\Gamma \vee \varphi_\Gamma \mid \exists^{\mathsf{P}} x.\, \varphi_\Gamma \mid \mathbf{X}_\Gamma\varphi_\Gamma \mid \mathbf{Y}_\Gamma\varphi_\Gamma \mid \varphi_\Gamma\, \mathbf{U}_\Gamma\, \varphi_\Gamma \mid \varphi_\Gamma\, \mathbf{S}_\Gamma\, \varphi_\Gamma$$

where ψ is a PLTL formula. Note that $\exists x.\, \varphi$ can be expressed as $\exists^{\mathsf{P}} x.\, (\varphi \wedge \langle x\rangle\neg\mathbf{Y}\top)$. $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$ is the class of formulas obtained with the syntax above for a given Γ. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ is the union $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$ for all Γ. We say that a formula φ of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ is *singleton-free* if for each subformula $\langle x\rangle\psi[x]$ of φ, ψ is an atomic proposition. Evidently, for an atomic proposition p, $\langle x\rangle p[x]$ is equivalent to $p[x]$.

In Section 4, we will show that (fair) model checking of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ is decidable. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ can be seen as a very large fragment of $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ with a decidable model checking problem which subsumes the simple fragment of $\mathsf{HyperLTL}_{\mathsf{S}}$, is closed under Boolean connectives, and allows an unrestricted nesting of temporal modalities. We conjecture (without proof) that this is the largest such sub-class of $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ because:

1. Model checking of $\mathsf{HyperLTL}_{\mathsf{S}}$ is already undecidable [7] for sentences whose relativized temporal modalities exploit two distinct sets of LTL formulas and, in particular, for two-variable quantifier alternation-free sentences of the form $\exists x_1.\, \exists x_2.\, (\varphi \wedge \mathbf{G}_\Gamma\psi)$, where ψ is a propositional formula, Γ is a nonempty set of propositions, and φ is a quantifier-free formula which use only the temporal modalities $\mathbf{F}_\emptyset$ and $\mathbf{G}_\emptyset$.
2. Model checking of $\mathsf{HyperLTL}_{\mathsf{C}}$ is undecidable [8] even for the fragment consisting of two-variable quantifier alternation-free sentences of the form $\exists x_1.\exists x_2.\, \psi_0 \wedge \langle x_2\rangle\, \mathbf{F}\, \langle\{x_1, x_2\}\rangle\psi$, where ψ_0 and ψ are quantifier-free $\mathsf{HyperLTL}$ formulas (note that ψ_0 is evaluated in the global context $\langle\{x_1, x_2\}\rangle$).

The second undecidability result suggests to consider the extension of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ where singleton-context subformulas of the form $\langle x\rangle\psi[x]$ are replaced with *quantifier-free* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ formulas with multiple variables of the form $\langle x\rangle\xi$, where ξ only uses singleton contexts $\langle y\rangle$ and temporal modalities with subscript $\emptyset$. However, we can show that the resulting logic is not more expressive than simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$: a sentence in the considered extension can be translated into an equivalent simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ sentence though with a non-elementary blowup (for details, see [4]).

3.4 Examples of Specifications in Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$

We consider some relevant properties which can be expressed in simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ subsumes the simple fragment of $\mathsf{HyperLTL}_{\mathsf{S}}$, and this fragment (as shown in [7]) can express relevant information-flow security properties for asynchronous frameworks such as distributed systems or cryptographic protocols. An example is the asynchronous variant of the *noninterference* property, as defined by Goguen and Meseguer [18], which asserts that the observations of low users (users accessing only to public information) do not change when all inputs of high users (users accessing secret information) are removed.

Observational Determinism

An important information-flow property is observational determinism, which states that traces which have the same initial low inputs are indistinguishable to a low user. In an asynchronous setting, a user cannot infer that a transition occurred if consecutive observations remain unchanged. Thus, for instance, observational determinism with equivalence of traces up to stuttering (as formulated in [36]) can be captured by the following simple $\mathsf{HyperLTL_S}$ sentence (where LI is the set of propositions describing inputs of low users and LO is set of propositions describing outputs of low users):

$$\forall x. \forall y. \bigwedge_{p \in LI} (p[x] \leftrightarrow p[y]) \rightarrow \mathbf{G}_{LO} \bigwedge_{p \in LO} (p[x] \leftrightarrow p[y])$$

After-initialization Properties

Simple $\mathsf{GHyperLTL_{S+C}}$ also allows to specify complex combinations of asynchronous and synchronous constraints. As an example, we consider the property [21] that for an HyperLTL sentence $\mathsf{Q}_1 x_1. \ldots \mathsf{Q}_n x_n. \psi(x_1, \ldots, x_n)$, the quantifier-free formula $\psi(x_1, \ldots, x_n)$ holds along the traces bound by variables $x_1, \ldots, x_n$ after an initialization phase. Note that this phase can take a different (and unbounded) number of steps on each trace. Let in be a proposition characterizing the initialization phase. The formula $PI(x) := \langle x \rangle (\neg in[x] \wedge (\neg \mathbf{Y} \top \vee \mathbf{Y} in[x]))$ is a simple $\mathsf{GHyperLTL_{S+C}}$ formula that asserts that for the pointed trace (σ, i) assigned to variable x, the position i is the first position of σ following the initialization phase. In other words, i is the first position at which $\neg in[]$ holds. Then, the previous requirement can be expressed in simple $\mathsf{GHyperLTL_{S+C}}$ as follows:

$$\mathsf{Q}_1 x_1. \ldots \mathsf{Q}_n x_n. \big(PI(x_1) \circ_1 \ldots PI(x_n) \circ_n \psi(x_1, \ldots x_n)$$

where $\circ_i$ is $\wedge$ if $\mathsf{Q}_i = \exists$ and $\circ_i$ is $\rightarrow$ if $\mathsf{Q}_i = \forall$.

Global Promptness

As another meaningful example, we consider global promptness (in the style of Prompt LTL [25]), where one need to check that there is a uniform bound on the response time in all the traces of the system, that is, "*there is k such that for every trace, each request q is followed by a response p within k steps*". Global promptness is expressible in simple $\mathsf{GHyperLTL_{S+C}}$ as follows:

$$\exists^{\mathsf{P}} x. \big(q[x] \wedge \forall^{\mathsf{P}} y. (q[y] \rightarrow (\neg p[x] \ \mathbf{U} \ p[y]))\big)$$

The previous sentence asserts that there is request (x in the formula) that has the longest response. Note that y is quantified universally (so it can be instantiated to the same trace as x), and that the use of until in $(\neg p[x] \ \mathbf{U} \ p[y]))\big)$ implies that the response $p[y]$ eventually happens. Hence, all requests, including receive a response. Now, the pointed trace (σ_x, i) assigned to x is such that $\sigma_x(i)$ is a request ($q[x]$) and for every pointed trace (σ_y, j) in the model such that $\sigma_y(j)$ is a request ($q[y]$), it holds that (i) the request $\sigma_y(j)$ is followed by a response $\sigma_y(j+k)$ for some $k \geq 0$, and (ii) no response occurs in σ_x in the interval $[i, i+k)$. Therefore, the response time h for x is the smallest h such that $\sigma_x(i+h)$ is a response is a global bound on the response time.

Diagnosability

We now show that simple $\mathsf{GHyperLTL_{S+C}}$ is also able to express *diagnosability* of systems [31, 5, 3] in a general asynchronous setting. In the diagnosis process, faults of a critical system (referred as the plant) are handled by a dedicated module (the *diagnoser*) which runs in parallel with the plant. The diagnoser analyzes the observable information from the plant – made available by predefined sensors – and triggers suitable alarms in correspondence to (typically unobservable) conditions of interest, called faults. An alarm condition specifies the relation (delay) between a given diagnosis condition and the raising of an alarm. A plant $\mathcal{P}$ is *diagnosable* with respect to a given alarm condition α, if there is a diagnoser $\mathcal{D}$ which satisfies α when $\mathcal{D}$ runs in parallel with $\mathcal{P}$.

The given set of propositions AP is partitioned into a set of observable propositions Obs and a set of unobservable propositions Int. Two finite traces σ and σ' are *observationally equivalent* iff the projections of $\mathit{stfr}_{\mathsf{Obs}}(\sigma \cdot P^\omega)$ and $\mathit{stfr}_{\mathsf{Obs}}(\sigma' \cdot (P')^\omega)$ over Obs coincide, where P is the last symbol of σ and similarly P' is the last symbol of σ'. Given a pointed trace (σ, i), i is an *observation point* of σ if either $i = 0$, or $i > 0$ and $\sigma(i-1) \cap \mathsf{Obs} \neq \sigma(i) \cap \mathsf{Obs}$. Then a plant $\mathcal{P}$ can be modeled as a finite Kripke structure $\langle S, S_0, E, \mathit{Lab}\rangle$, where E is partitioned into internal transitions (s, s') where $\mathit{Lab}(s) \cap \mathsf{Obs} = \mathit{Lab}(s') \cap \mathsf{Obs}$ and observable transitions where $\mathit{Lab}(s) \cap \mathsf{Obs} \neq \mathit{Lab}(s') \cap \mathsf{Obs}$. A diagnoser $\mathcal{D}$ is modelled as a finite deterministic Kripke structure over $\mathsf{AP}' \supseteq \mathsf{Obs}$ (with $\mathsf{AP}' \cap \mathsf{Int} = \emptyset$). In the behavioural composition of the plant $\mathcal{P}$ with $\mathcal{D}$, the diagnoser only reacts to the observable transitions of the plant, that is, every transition of the diagnoser is associated with an observable transition of the plant. Simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ can express diagnosability with *finite delay*, *bounded delay*, or *exact delay* as defined in [5, 3]. Here, we focus for simplicity on finite delay diagnosability. Consider a diagnosis condition specified by a PLTL formula β. A plant $\mathcal{P}$ is *finite delay diagnosable* with respect to β whenever for every pointed trace (σ, i) of $\mathcal{P}$ such that $(\sigma, i) \models \beta$, there exists an observation point $k \geq i$ of σ such that for all pointed traces (σ', k') of $\mathcal{P}$ so that k' is an observation point of σ' and $\sigma[0, k]$ and $\sigma'[0, k']$ are observationally equivalent, it holds that $(\sigma', i') \models \beta$ for some $i' \leq k'$. Finite delay diagnosability w.r.t. β can be expressed in simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ as follows:

$$\forall^{\mathsf{P}} x.\, \Big(\langle x\rangle \beta[x] \rightarrow \mathbf{F}_{\mathsf{Obs}}\big(\mathsf{ObsPt}(x) \wedge \forall^{\mathsf{P}} y.\, \{(\mathsf{ObsPt}(y) \wedge \theta_{\mathsf{Obs}}(x,y)) \rightarrow \langle y\rangle\, \mathbf{O}\, \beta[y]\}\big)\Big)$$

where

$$\theta_{\mathsf{Obs}}(x,y) \;:=\; \bigwedge_{p \in \mathsf{Obs}} \mathbf{H}_{\mathsf{Obs}}\,(p[x] \leftrightarrow p[y]) \;\wedge\; \mathbf{O}_{\mathsf{Obs}}(\langle x\rangle \neg \mathbf{Y}\top \wedge \langle y\rangle \neg \mathbf{Y}\top)$$

$$\mathsf{ObsPt}(x) \;:=\; \langle x\rangle(\neg \mathbf{Y}\top \wedge \bigvee_{p \in \mathsf{Obs}} (p[x] \leftrightarrow \neg \mathbf{Y} p[x])$$

Essentially $\mathsf{ObsPt}(x)$ determines the observation points and θ_{Obs} captures that both traces have the same history of observations. The main formula establishes that if x detects a failure β then there is future observation point in x and for all other traces that are observationally equivalent to x have also detected β.

3.5 Expressiveness Issues

In this section, we present some results and conjectures about the expressiveness comparison among $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ (which subsumes $\mathsf{HyperLTL}_{\mathsf{S}}$ and $\mathsf{HyperLTL}_{\mathsf{C}}$), its simple fragment and the logic $\mathsf{HyperLTL}_{\mathsf{S}}$. We also consider the logics for linear-time hyperproperties based on the equal-level predicate whose most powerful representative is $\mathsf{S1S[E]}$. Recall that the first-order fragment $\mathsf{FO[<,E]}$ of $\mathsf{S1S[E]}$ is already strictly more expressive than $\mathsf{HyperLTL}$ [16] and, unlike $\mathsf{S1S[E]}$, its model-checking problem is decidable [11]. Moreover, we show that $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ and its simple fragment represent a unifying framework in the linear-time setting for specifying both hyperproperties and the knowledge modalities of epistemic temporal logics under both the synchronous and asynchronous perfect recall semantics.

Our expressiveness results about linear-time hyper logics can be summarized as follows.

▶ **Theorem 3.2.** *The following hold:*

- $\mathsf{GHyperLTL_{S+C}}$ *is more expressive than* $\mathsf{HyperLTL_S}$*, simple* $\mathsf{GHyperLTL_{S+C}}$*, and* $\mathsf{FO[<,E]}$.
- *Simple* $\mathsf{GHyperLTL_{S+C}}$ *is more expressive than simple* $\mathsf{HyperLTL_S}$.
- *Simple* $\mathsf{GHyperLTL_{S+C}}$ *and* $\mathsf{HyperLTL_S}$ *are expressively incomparable.*
- *Simple* $\mathsf{GHyperLTL_{S+C}}$ *and* $\mathsf{S1S[E]}$ *are expressively incomparable.*

Proof. We first show that there are hyperproperties expressible in simple $\mathsf{GHyperLTL_{S+C}}$ but not in $\mathsf{HyperLTL_S}$ and in $\mathsf{S1S[E]}$. Given a sentence φ, the *trace property denoted by* φ is the set of traces σ such that the singleton set of traces $\{\sigma\}$ satisfies φ. It is known that $\mathsf{HyperLTL_S}$ and $\mathsf{S1S[E]}$ capture only *regular* trace properties [8]. In contrast simple $\mathsf{GHyperLTL_{S+C}}$ can express powerful non-regular trace properties. For example, consider the so called *suffix property* over $\mathsf{AP} = \{p\}$: a trace σ satisfies the suffix property if there exists a proper suffix σ^k of σ for some $k > 0$ such that $\sigma^k = \sigma$. This non-regular trace property can be expressed in $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ as follows:

$$\forall x_1.\,\forall x_2.\ \bigwedge_{p\in\mathsf{AP}} \mathbf{G}(p[x_1] \leftrightarrow p[x_2]) \wedge \forall x_1.\,\exists^{\mathsf{P}} x_2.\,\Big(\bigwedge_{p\in\mathsf{AP}} \mathbf{G}(p[x_1] \leftrightarrow p[x_2]) \wedge \langle x_2\rangle \mathbf{Y}\top\Big)$$

The first conjunct asserts that each model is a singleton, and the second conjunct requires that for the unique trace σ in a model, there is $k > 0$ such that $\sigma(i) = \sigma(i + k)$ for all $i \geq 0$.

Next, we observe that $\mathsf{FO[<,E]}$ can be easily translated into $\mathsf{GHyperLTL_{S+C}}$, since the pointer quantifiers of $\mathsf{GHyperLTL_{S+C}}$ correspond to the quantifiers of $\mathsf{FO[<,E]}$. Moreover, the predicate $x \leq x'$ of $\mathsf{FO[<,E]}$, expressing that for the pointed traces (σ, i) and (σ', i') bound to x and x', $\sigma = \sigma'$ and $i \leq i'$, can be easily captured in $\mathsf{GHyperLTL_{S+C}}$. This is also the case for the equal-level predicate $\mathsf{E}(x, x')$, which can be expressed as $\langle\{x, x'\}\rangle\, \mathbf{O}\,(\langle x\rangle\neg\mathbf{Y}\top \wedge \langle x'\rangle\neg\mathbf{Y}\top)$.

In Section 4 we show that model checking of simple $\mathsf{GHyperLTL_{S+C}}$ is decidable. Thus, since model checking of both $\mathsf{HyperLTL_S}$ and $\mathsf{S1S[E]}$ are undecidable [7, 11] and $\mathsf{GHyperLTL_{S+C}}$ subsumes $\mathsf{HyperLTL_S}$, by the previous argumentation, the theorem follows. ◀

It remains an open question whether $\mathsf{FO[<,E]}$ is subsumed by simple $\mathsf{GHyperLTL_{S+C}}$. We conjecture that neither $\mathsf{HyperLTL_C}$ nor the fix-point calculus H_μ [21] (which captures both $\mathsf{HyperLTL_C}$ and $\mathsf{HyperLTL_S}$ [8]) subsume simple $\mathsf{GHyperLTL_{S+C}}$. The motivation for our conjecture is that H_μ sentences consist of a prefix of quantifiers followed by a quantifier-free formula where quantifiers range over *initial* pointed traces $(\sigma, 0)$. Thus, unlike simple $\mathsf{GHyperLTL_{S+C}}$, H_μ cannot express requirements which relate at some point in time an unbounded number of traces. Diagnosability (see Subsection 3.4) falls in this class of requirements. It is known that the following property, which can be easily expressed in simple $\mathsf{GHyperLTL_{S+C}}$, is not definable in $\mathsf{HyperLTL}$ [6]: for some $i > 0$, every trace in the given set of traces does not satisfy proposition p at position i. We conjecture that similarly to $\mathsf{HyperLTL}$, such a property (and diagnosability as well) cannot be expressed in H_μ.

Epistemic Temporal Logic KLTL and its relation with $\mathsf{GHyperLTL_{S+C}}$. The logic KLTL [23] is a well-known extension of LTL obtained by adding the unary knowledge modalities $\mathbf{K}_a$, where a ranges over a finite set Agts of agents, interpreted under the synchronous or asynchronous (perfect recall) semantics. The semantics is given with respect to an observation map $\mathsf{Obs} : \mathsf{Agts} \mapsto 2^{\mathsf{AP}}$ that assigns to each agent a the set of propositions which agent a can observe. Given two finite traces σ and σ' and $a \in \mathsf{Agts}$, σ and σ' are *synchronously equivalent for agent* a, written $\sigma \sim^{sy}_a \sigma'$, if the projections of σ and σ' over $\mathsf{Obs}(a)$ coincide. The finite traces σ and σ' are *asynchronously equivalent for agent* a, written $\sigma \sim^{as}_a \sigma'$, if the projections of $stfr_{\mathsf{Obs}(a)}(\sigma \cdot P^\omega)$ and $stfr_{\mathsf{Obs}(a)}(\sigma' \cdot (P')^\omega)$ over $\mathsf{Obs}(a)$ coincide, where

P is the last symbol of σ and P' is the last symbol of σ'. For a set of traces $\mathcal{L}$ and a pointed trace (σ, i) over $\mathcal{L}$, the semantics of the knowledge modalities is as follows, where $\sim_a$ is $\sim_a^{sy}$ under the synchronous semantics, and $\sim_a^{as}$ otherwise: $(\sigma, i) \models_{\mathcal{L},\mathsf{Obs}} \mathbf{K}_a\, \varphi \;\Leftrightarrow$ for all pointed traces (σ', i') on $\mathcal{L}$ such that $\sigma[0,i] \sim_a \sigma'[0,i']$, $(\sigma', i') \models_{\mathcal{L},\mathsf{Obs}} \varphi$.

We say that $\mathcal{L}$ *satisfies* φ *w.r.t. the observation map Obs*, written $\mathcal{L} \models_{\mathsf{Obs}} \varphi$, if for all traces $\sigma \in \mathcal{L}$, $(\sigma, 0) \models_{\mathcal{L},\mathsf{Obs}} \varphi$. The logic KLTL can be easily embedded into $\mathsf{GHyperLTL_{S+C}}$. In particular, the following holds (for details, see [4]).

▶ **Theorem 3.3.** *Given an observation map Obs and a* KLTL *formula* ψ *over AP, one can construct in linear time a* $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ *sentence* $\varphi_\emptyset$ *and a* $\mathsf{GHyperLTL_{S+C}}$ *sentence* φ *such that* $\varphi_\emptyset$ *is equivalent to* ψ *w.r.t. Obs under the synchronous semantics and* φ *is equivalent to* ψ *w.r.t. Obs under the asynchronous semantics. Moreover,* φ *is a simple* $\mathsf{GHyperLTL_{S+C}}$ *sentence if* ψ *is in the single-agent fragment of* KLTL.

4 Decidability of Model Checking against Simple $\mathsf{GHyperLTL_{S+C}}$

In this section, we show that (fair) model checking against simple $\mathsf{GHyperLTL_{S+C}}$ is decidable. We first prove the result for the fragment $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ of simple $\mathsf{GHyperLTL_{S+C}}$ by a linear-time reduction to satisfiability of *full* Quantified Propositional Temporal Logic (QPTL, for short) [32], where the latter extends PLTL by quantification over propositions. Then, we show that (fair) model checking of simple $\mathsf{GHyperLTL_{S+C}}$ can be reduced in time singly exponential in the size of the formula to fair model checking of $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. We also provide optimal complexity bounds for (fair) model checking the fragment $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ in terms of a parameter of the given formula called *strong alternation depth*. For this, we first give similar optimal complexity bounds for satisfiability of QPTL.
The syntax of QPTL formulas φ over a finite set AP of atomic propositions is as follows:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \mathbf{Y}\varphi \mid \varphi\, \mathbf{U}\, \varphi \mid \varphi\, \mathbf{S}\, \varphi \mid \exists p\,.\varphi$$

where $p \in \mathsf{AP}$ and $\exists p$ is the propositional existential quantifier. A QPTL formula φ is a *sentence* if each proposition p occurs in the scope of a quantifier binding p and each temporal modality occurs in the scope of a quantifier. By introducing $\wedge$ and the operators $\mathbf{R}$ (*release*, dual of $\mathbf{U}$), $\mathbf{P}$ (*past release*, dual of $\mathbf{S}$) and $\forall p$ (propositional universal quantifier), every QPTL formula can be converted into an equivalent formula in *negation normal form*, where negation only appears in front of atomic propositions. QPTL formulas are interpreted over pointed traces (σ, i) over AP. All QPTL temporal operators have the same semantics as in PLTL. The semantics of propositional quantification is as follows:

$$(\sigma, i) \models \exists p.\varphi \;\Leftrightarrow\; \text{there is a trace } \sigma' \text{ such that } \sigma =_{\mathsf{AP}\setminus\{p\}} \sigma' \text{ and } (\sigma', i) \models \varphi$$

where $\sigma =_{\mathsf{AP}\setminus\{p\}} \sigma'$ means that the projections of σ and σ' over $\mathsf{AP} \setminus \{p\}$ coincide. A formula φ is satisfiable if $(\sigma, 0) \models \varphi$ for some trace σ. We now give a generalization of the standard notion of alternation depth between existential and universal quantifiers which corresponds to the one given in [30] for HyperCTL*. Our notion takes into account also the occurrences of temporal modalities between quantifier occurrences, but the nesting depth of temporal modalities is not considered (intuitively, it is collapsed to one). Formally, the *strong alternation depth* $sad(\varphi)$ of a QPTL formula φ in negation normal form is inductively defined as follows, where an existential formula is a formula of the form $\exists p.\,\psi$, a universal formula is of the form $\forall p.\,\psi$, and for a formula ψ, $\widetilde{\psi}$ denotes the negation normal form of $\neg\psi$:

- For $\varphi = p$ and $\varphi = \neg p$ for a given $p \in \mathsf{AP}$: $sad(\varphi) := 0$.
- For $\varphi = \varphi_1 \vee \varphi_2$ and for $\varphi = \varphi_1 \wedge \varphi_2$: $sad(\varphi) := \max(\{sad(\varphi_1), sad(\varphi_2)\})$.
- For $\varphi = \exists p.\, \varphi_1$: if there is no universal sub-formula $\forall\psi$ of φ_1 such that $sad(\forall\psi) = sad(\varphi_1)$, then $sad(\varphi) := sad(\varphi_1)$. Otherwise, $sad(\varphi) := sad(\varphi_1) + 1$.
- For $\varphi = \forall p.\, \varphi_1$: $sad(\varphi) := sad(\exists p.\, \widetilde{\varphi_1})$.
- For $\varphi = \mathbf{X}\varphi_1$ or $\varphi = \mathbf{Y}\varphi_1$: $sad(\varphi) := sad(\varphi_1)$.
- For $\varphi = \varphi_1 \,\mathbf{U}\, \varphi_2$ or $\varphi = \varphi_1 \,\mathbf{S}\, \varphi_2$: let h be the maximum over the strong alternation depths of the universal and existential sub-formulas of φ_1 and φ_2 (the maximum of the empty set is 0). If the following conditions are met, then $sad(\varphi) := h$; otherwise, $sad(\varphi) := h + 1$:
  - there is no existential or universal sub-formula ψ of φ_1 with $sad(\psi) = h$;
  - there is no universal sub-formula ψ of φ_2 with $sad(\psi) = h$;
  - no existential formula ψ with $sad(\psi) = h$ occurs in the left operand (resp., right operand) of a sub-formula of φ_2 of the form $\psi_1 \mathcal{O} \psi_2$, where $\mathcal{O} \in \{\mathbf{U}, \mathbf{S}\}$ (resp., $\mathcal{O} \in \{\mathbf{R}, \mathbf{P}\}$).
- Finally, for $\varphi = \varphi_1 \,\mathbf{R}\, \varphi_2$ or $\varphi = \varphi_1 \,\mathbf{P}\, \varphi_2$: $sad(\varphi) := sad(\widetilde{\varphi})$.

For example, $sad(\exists p.(p \,\mathbf{U} \,\exists q.q)) = 0$ and $sad(\exists p.(\exists p.p \,\mathbf{U}\, q)) = 1$. The strong alternation depth of an arbitrary QPTL formula corresponds to the one of its negation normal form. The strong alternation depth of a simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ formula is defined similarly but we replace quantification over propositions with quantification over trace variables. For all $n, h \in \mathbb{N}$, $\mathsf{Tower}(h, n)$ denotes a tower of exponentials of height h and argument n: $\mathsf{Tower}(0, n) = n$ and $\mathsf{Tower}(h + 1, n) = 2^{\mathsf{Tower}(h,n)}$. Essnetially, the strong alternation depth corresponds to the (unavoidable) power set construction related to the alternation of quantifiers to solve the model-checking problem.

The following result represents an improved version of Theorem 6 in [6] where h-EXPSPACE is the class of languages decided by deterministic Turing machines bounded in space by functions of n in $O(\mathsf{Tower}(h, n^c))$ for some constant $c \geq 1$. While the lower bound directly follows from [32], the upper bound improves the result in [6], since there, occurrences of temporal modalities immediately preceding propositional quantification always count as additional alternations (for details, see [4]).

▶ **Theorem 4.1.** *For all $h \geq 0$, satisfiability of* QPTL *sentences with strong alternation depth at most h is h-*EXPSPACE*-complete.*

(Fair) Model checking against $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. We provide now linear-time reductions of (fair) model checking against $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ to (and from) satisfiability of QPTL which preserve the strong alternation depth. We start with the reduction of (fair) model checking $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ to QPTL satisfiability.

▶ **Theorem 4.2.** *Given a fair finite Kripke structure $(\mathcal{K}, F)$ and a $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ sentence φ, one can construct in linear time a* QPTL *sentence ψ with the same strong alternation depth as φ such that ψ is satisfiable if and only if $\mathcal{L}(\mathcal{K}, F) \models \varphi$.*

Sketched proof. Let $\mathcal{K} = \langle S, S_0, E, Lab\rangle$. In the reduction of model checking $(\mathcal{K}, F)$ against φ to QPTL satisfiability, we need to merge multiple traces into a unique trace where just one position is considered at any time. An issue is that the hyper quantifiers range over arbitrary pointed traces so that the positions of the different pointed traces in the current trace assignment do not necessarily coincide (intuitively, the different pointed traces are not aligned with respect to the relative current positions). However, we can solve this issue because the offsets between the positions of the pointed traces in the current trace assignment remain constant during the evaluation of the temporal modalities. In particular, assume that

(σ, i) is the first pointed trace selected by a hyper quantifier during the evaluation along a path in the syntax tree of φ. We encode σ by keeping track also of the variable x to which (σ, i) is bound and the F-fair path of $\mathcal{K}$ whose associated trace is σ. Let (σ', i') be another pointed trace introduced by another hyper quantifier y during the evaluation of φ. If $i' < i$, we consider an encoding of σ' which is similar to the previous encoding but we precede it with a *padding prefix* of length $i - i'$ of the form $\{\#_{\overrightarrow{y}}\}^{i-i'}$. The arrow $\rightarrow$ indicates that the encoding is along the *forward direction.* Now, assume that $i' > i$. In this case, the encoding of σ' is the merging of two encodings over disjoint sets of propositions: one along the forward direction which encodes the suffix $(\sigma')^{i'-i}$ and another one along the *backward direction* which is of the form $\{\#_{\overleftarrow{y}}\} \cdot \rho \cdot \{\#_{\overleftarrow{y}}\}^{\omega}$, where ρ is a backward encoding of the *reverse* of the prefix of σ' of length $i' - i$. In such a way, the encodings of the pointed traces later introduced in the evaluation of φ are aligned with the reference pointed trace (σ, i). Since the positions in the backward direction overlap some positions in the forward direction, in the translation, we keep track of whether the current position refers to the forward or to the backward direction. The details of the reduction are in [4]. ◀

By an adaptation of the known reduction of satisfiability of QPTL without past to model checking of HyperCTL* [9], we obtain the following result (for details, see [4]).

▶ **Theorem 4.3.** *Given a* QPTL *sentence ψ over AP, one can build in linear time a finite Kripke structure $\mathcal{K}_{AP}$ (depending only on AP) and a singleton-free* $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ *sentence φ having the same strong alternation depth as ψ such that ψ is satisfiable* iff $\mathcal{L}(\mathcal{K}_{AP}) \models \varphi$.

Hence, by Theorems 4.1–4.3, we obtain the following result.

▶ **Corollary 4.4.** *For all $h \geq 0$, (fair) model checking against* $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ *sentences with strong alternation depth at most h is h-*EXPSPACE*-complete.*

Reduction to fair model checking against $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. We solve the (fair) model checking problem for simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ by a reduction to fair model checking against the fragment $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. Our reduction is exponential in the size of the given sentence and is an adaptation of the reduction from model checking simple $\mathsf{HyperLTL}_{\mathsf{S}}$ to model checking HyperLTL shown in [7]. As a preliminary step, we first show, by an easy adaptation of the standard automata-theoretic approach for PLTL [34], that the problem for a simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ sentence φ can be reduced in exponential time to the fair model checking problem against a singleton-free sentence in the fragment $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$ for some set Γ of *atomic propositions* depending on φ. For details, see [4].

▶ **Proposition 4.5.** *Given a simple* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ *sentence φ and a fair finite Kripke structure $(\mathcal{K}, F)$ over AP, one can build in single exponential time in the size of φ, a fair finite Kripke structure $(\mathcal{K}', F')$ over an extension AP' of AP and a singleton-free* $\mathsf{SHyperLTL}^{\Gamma'}_{\mathsf{S+C}}$ *sentence φ' for some $\Gamma' \subseteq AP'$ such that $\mathcal{L}(\mathcal{K}', F') \models \varphi'$ if and only if $\mathcal{L}(\mathcal{K}, F) \models \varphi$. Moreover, φ' has the same strong alternation depth as φ, $|\varphi'| = O(|\varphi|)$, and $|\mathcal{K}'| = O(|\mathcal{K}| * 2^{O(|\varphi|)})$.*

Let us fix a non-empty finite set $\Gamma \subseteq$ AP of atomic propositions. We now show that fair model checking of the singleton-free fragment of $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$ can be reduced in polynomial time to fair model checking of $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. We observe that in the singleton-free fragment of $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$, when a pointed trace (σ, i) is selected by a pointed quantifier $\exists^{\mathsf{P}} x$, the positions of σ which are visited during the evaluation of the temporal modalities are all in the (Γ, ω)-stutter factorization of σ with the possible exception of the position i chosen by $\exists^{\mathsf{P}} x$. Thus, given a set $\mathcal{L}$ of traces and a special proposition $\# \notin$ AP, we define an extension

$stfr^{\#}_{\Gamma}(\mathcal{L})$ of the set $stfr_{\Gamma}(\mathcal{L}) = \{stfr_{\Gamma}(\sigma) \mid \sigma \in \mathcal{L}\}$ as follows. Intuitively, we consider for each trace $\sigma \in \mathcal{L}$, its Γ-stutter trace $stfr_{\Gamma}(\sigma)$ and the extensions of $stfr_{\Gamma}(\sigma)$ which are obtained by adding an extra position marked by proposition $\#$ (this extra position does not belong to the (Γ,ω)-stutter factorization of σ). Formally, $stfr^{\#}_{\Gamma}(\mathcal{L})$ is the smallest set containing $stfr_{\Gamma}(\mathcal{L})$ and satisfying the following condition:

- for each trace $\sigma \in \mathcal{L}$ with (Γ,ω)-stutter factorization $\{\ell_k\}_{k\geq 0}$ and position $i \in (\ell_k, \ell_{k+1})$ for some $k \geq 0$, the trace $\sigma(\ell_0)\ldots\sigma(\ell_k)\,(\sigma(i)\cup\{\#\})\,\sigma(\ell_{k+1})\,\sigma(\ell_{k+2})\ldots \in stfr^{\#}_{\Gamma}(\mathcal{L})$.

Given a singleton-free formula φ in $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$, we denote by $\mathsf{T}_{\#}(\varphi)$ the formula in $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ obtained from φ by applying inductively the following transformations:

- the Γ-relativized temporal modalities are replaced with their $\emptyset$-relativized counterparts;
- each formula $\exists^{\mathsf{P}} x.\,\phi$ is replaced with $\exists^{\mathsf{P}} x.\,\big(\mathsf{T}_{\#}(\phi) \wedge \langle x\rangle(\mathbf{X}\,\mathbf{G}\,\neg\#[x] \wedge (\mathbf{Y}\top \rightarrow \mathbf{Y}\mathbf{H}\neg\#[x]))\big)$.

Intuitively, the formula $\mathsf{T}_{\#}(\exists^{\mathsf{P}} x.\,\phi)$ states that for the pointed trace (σ, i) selected by the pointed quantifier, at most position i may be marked by the special proposition $\#$. By the semantics of the logics considered, the following holds.

▶ Remark 4.6. Given a singleton-free sentence φ of $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$ and a set $\mathcal{L}$ of traces, it holds that $\mathcal{L} \models \varphi$ if and only if $stfr^{\#}_{\Gamma}(\mathcal{L}) \models \mathsf{T}_{\#}(\varphi)$.

Let us fix now a fair finite Kripke structure $(\mathcal{K}, F)$. We first show that one can build in polynomial time a finite Kripke structure $(\mathcal{K}_{\Gamma}, F_{\Gamma})$ and a LTL formula θ_{Γ} such that $stfr^{\#}_{\Gamma}(\mathcal{L}(\mathcal{K}, F))$ coincides with the traces of $\mathcal{L}(\mathcal{K}_{\Gamma}, F_{\Gamma})$ satisfying θ_{Γ} (details are in [4]).

▶ **Proposition 4.7.** *Given $\emptyset \neq \Gamma \subseteq AP$ and a fair finite Kripke structure $(\mathcal{K}, F)$ over AP, one can construct in polynomial time a fair finite Kripke structure $(\mathcal{K}_{\Gamma}, F_{\Gamma})$ and a* LTL *formula θ_{Γ} such that $stfr^{\#}_{\Gamma}(\mathcal{L}(\mathcal{K}, F))$ is the set of traces $\sigma \in \mathcal{L}(\mathcal{K}_{\Gamma}, F_{\Gamma})$ so that $\sigma \models \theta_{\Gamma}$.*

Fix now a singleton-free sentence φ of $\mathsf{SHyperLTL}^{\Gamma}_{\mathsf{S+C}}$. For the given fair finite Kripke structure $(\mathcal{K}, F)$ over AP, let $(\mathcal{K}_{\Gamma}, F_{\Gamma})$ and θ_{Γ} as in the statement of Proposition 4.7. We consider the $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ sentence $\mathsf{T}(\varphi)$ obtained from $\mathsf{T}_{\#}(\varphi)$ by inductively replacing each subformula $\exists^{\mathsf{P}} x.\,\phi$ of $\mathsf{T}_{\#}(\varphi)$ with $\exists^{\mathsf{P}} x.\,(\mathsf{T}(\phi) \wedge \langle x\rangle\,\mathbf{O}\,(\neg\mathbf{Y}\top \wedge \theta_{\Gamma}[x]))$. In other terms, we ensure that in $\mathsf{T}_{\#}(\varphi)$ the hyper quantification ranges over traces which satisfy the LTL formula θ_{Γ}. By Remark 4.6 and Proposition 4.7, we obtain that $\mathcal{L}(\mathcal{K}, F) \models \varphi$ if and only if $\mathcal{L}(\mathcal{K}_{\Gamma}, F_{\Gamma}) \models \mathsf{T}(\varphi)$. Thus, together with Proposition 4.5, we obtain the following result.

▶ **Theorem 4.8.** *The (fair) model checking problem against simple* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ *can be reduced in singly exponential time to fair model checking against* $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$.

5 Conclusion

We have introduced a novel hyper logic $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ which merges two known asynchronous temporal logics for hyperproperties, namely stuttering HyperLTL and context HyperLTL. Even though model checking of the resulting logic $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ is undecidable, we have identified a useful fragment, called *simple* $\mathsf{GHyperLTL}_{\mathsf{S+C}}$, that has a decidable model checking, is strictly more expressive than HyperLTL and than previously proposed fragments of asynchronous temporal logics for hyperproperties with a decidable model checking. For the fragment $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$ of simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$, we have given optimal complexity bounds of (fair) model checking in terms of the strong alternation depth of the given sentence. For arbitrary sentences in simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$, (fair) model checking is reduced in exponential time to fair model checking of $\mathsf{SHyperLTL}^{\emptyset}_{\mathsf{S+C}}$. It is worth noting that simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ can express non-regular trace properties over singleton sets of traces which are not definable in S1S[E]. An intriguing open question is whether FO[<,E] can be embedded in simple

$\mathsf{GHyperLTL}_{\mathsf{S+C}}$. In a companion paper, we study asynchronous properties on finite traces by adapting simple $\mathsf{GHyperLTL}_{\mathsf{S+C}}$ in prenex form to finite traces, and introduce practical model-checking algorithms for useful fragments of this logic.

References

1 Jan Baumeister, Norine Coenen, Borzoo Bonakdarpour, Bernd Finkbeiner, and César Sánchez. A Temporal Logic for Asynchronous Hyperproperties. In *Proc. 33rd CAV*, volume 12759 of *LNCS 12759*, pages 694–717. Springer, 2021. `doi:10.1007/978-3-030-81685-8_33`.

2 Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. Second-Order Hyperproperties. In *Proc. 35th CAV*, volume 13965 of *Lecture Notes in Computer Science*, pages 309–332. Springer, 2023. `doi:10.1007/978-3-031-37703-7_15`.

3 Benjamin Bittner, Marco Bozzano, Alessandro Cimatti, Marco Gario, Stefano Tonetta, and Viktoria Vozárová. Diagnosability of fair transition systems. *Artif. Intell.*, 309:103725, 2022. `doi:10.1016/J.ARTINT.2022.103725`.

4 Alberto Bombardelli, Laura Bozzelli, César Sánchez, and Stefano Tonetta. Unifying asynchronous logics for hyperproperties, 2024. `doi:10.48550/arXiv.2404.16778`.

5 Marco Bozzano, Alessandro Cimatti, Marco Gario, and Stefano Tonetta. Formal Design of Asynchronous Fault Detection and Identification Components using Temporal Epistemic Logic. *Log. Methods Comput. Sci.*, 11(4), 2015. `doi:10.2168/LMCS-11(4:4)2015`.

6 Laura Bozzelli, Bastien Maubert, and Spophie 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`.

7 Laura Bozzelli, Adriano Peron, and César Sánchez. Asynchronous Extensions of HyperLTL. In *Proc. 36th LICS*, pages 1–13. IEEE, 2021. `doi:10.1109/LICS52264.2021.9470583`.

8 Laura Bozzelli, Adriano Peron, and César Sánchez. Expressiveness and Decidability of Temporal Logics for Asynchronous Hyperproperties. In *Proc. 33rd CONCUR*, volume 243 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPICS.CONCUR.2022.27`.

9 Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César 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`.

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

11 Norine Coenen, Bernd Finkbeiner, Christopher Hahn, and Jana Hofmann. The hierarchy of hyperlogics. In *Proc. 34th LICS*, pages 1–13. IEEE, 2019. `doi:10.1109/LICS.2019.8785713`.

12 Rayna Dimitrova, Bernd Finkbeiner, Máté M Kovács, Markus N. Rabe, and Helmut 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`.

13 E. Allen Emerson and Joseph 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`.

14 Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about knowledge*, volume 4. MIT Press Cambridge, 1995. `doi:10.7551/mitpress/5803.001.0001`.

15 Bernd Finkbeiner and Christopher 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`.

16 Bernd Finkbeiner and Martin 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 Michael J. Fischer and Richard 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 Joseph A. Goguen and José 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 Jens Oliver Gutsfeld, Arne Meier, Christoph Ohrem, and Jonni Virtema. Temporal Team Semantics Revisited. In *Proc. 37th LICS*, pages 44:1–44:13. ACM, 2022. doi:10.1145/3531130.3533360.

20 Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph 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.

21 Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph Ohrem. Automata and fixpoints for asynchronous hyperproperties. *Proc. ACM Program. Lang.*, 4(POPL), 2021. doi:10.1145/3434319.

22 Joseph Y. Halpern and Kevin R. O'Neill. Secrecy in multiagent systems. *ACM Trans. Inf. Syst. Secur.*, 12(1), 2008. doi:10.1145/1410234.1410239.

23 Joseph Y. Halpern and Moshe Y. Vardi. The Complexity of Reasoning about Knowledge and Time: Extended Abstract. In *Proc. 18th STOC*, pages 304–315. ACM, 1986. doi:10.1145/12130.12161.

24 Andreas Krebs, Arne Meier, Jonni Virtema, and Martin 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 Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. From liveness to promptness. *Formal Methods Syst. Des.*, 34(2):83–103, 2009. doi:10.1007/S10703-009-0067-Z.

26 Martin 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.

27 Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems - Specification*. Springer-Verlag, 1992. doi:10.1007/978-1-4612-0931-7.

28 John D. 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.

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

30 Markus N. Rabe. *A temporal logic approach to information-flow control*. PhD thesis, Saarland University, 2016. URL: http://scidok.sulb.uni-saarland.de/volltexte/2016/6387/.

31 Meera Sampath, Raja Sengupta, Stephen Lafortune, Kazin Sinnamohideen, and Demosthenis Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Autom. Control.*, 40(9):1555–1575, 1995. doi:10.1109/9.412626.

32 A. Prasad Sistla, Moshe Y. Vardi, and Pierre 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.

33 Ron van der Meyden and Nikolay V. Shilov. Model checking knowledge and time in systems with perfect recall (extended abstract). In *Proc. 19th FSTTCS*, LNCS 1738, pages 432–445. Springer, 1999. doi:10.1007/3-540-46691-6_35.

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

35 Jonni Virtema, Jana Hofmann, Bernd Finkbeiner, Juha Kontinen, and Fan 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.

36 Steve Zdancewic and Andrew 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.