# Reasoning About Quality in Hyperproperties

**Samuel Graepler**
Fakultät für Mathematik und Informatik, Universität Leipzig, Germany

**Benjamin Monmege**
Aix Marseille Univ, CNRS, LIS, Marseille, France

**Jean-Marc Talbot**
Aix Marseille Univ, CNRS, LIS, Marseille, France
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

## Abstract

Hyperproperties allow one to specify properties of systems that inherently involve not single executions of the system, but several of them at once: observational determinism and non-inference are two examples of such properties used to study the security of systems. Logics like HyperLTL have been studied in the past to model check hyperproperties of systems. However, most of the time, requiring strict security properties is actually ineffective as systems do not meet such requirements. To overcome this issue, we introduce qualitative reasoning in HyperLTL, inspired by a similar work on LTL by Almagor, Boker and Kupferman [2] where a formula has a value in the interval $[0, 1]$, obtained by considering either a propositional quality (how much the specification is satisfied), or a temporal quality (when the specification is satisfied). We show decidability of the approximated model checking problem, as well as the model checking of large fragments.

**2012 ACM Subject Classification** Theory of computation → Logic and verification; Security and privacy

**Keywords and phrases** Hyperlogics, Automata-based model checking, Quantitative verification

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2026.45

**Related Version** *Full Version*: `https://arxiv.org/abs/2512.00500` [18]

**Funding** This work was partially done while the first author was an intern at Aix-Marseille Université. This work was partly supported by ANR-23-PECL-0009 TRUSTINCloudS.

**Acknowledgements** We thank the reviewers of the different versions of this article for their valuable feedback.

## 1 Introduction

Software safety and security have become a critical concern. Hence, ensuring that applications are free from safety issues or security vulnerabilities is an important area of the verification process. Model checking [5] is a successful formal method to automatically verify whether a system meets a specification. This specification is often described as formulas from dedicated logics, for instance LTL to express properties on execution traces of a system [22] regarding safety. Model checking, in particular of LTL, has been widely studied and leads to several tools like SPIN [19] and Spot [9].

Over the last years, hyperlogics have been found out to be another convenient tool to express properties about systems. These logics define hyperproperties [7] that consider not single executions as LTL, but several of them at once. Hence, hyperlogics are often used to express properties relating multiple executions of a system, e.g., network properties like congestion [4], or security and information-flow properties [15, 24, 26]. Let us focus here on the security property of non-interference [17], which holds when executions of systems do not reveal sensitive information to malicious observers: secret information cannot be deduced


from information of low sensitivity exposed by the system executions. As running examples, we consider two desirable properties to meet such requirements. We express them in the logic HyperLTL which extends LTL with a prefix of quantifications over traces variables [6]. First, the property called *observational determinism* [27] expresses the fact that when any two executions start with the same information of low sensitivity, then when observed all along the computation, the two traces cannot be distinguished on these low information (the system behaves deterministically with respect to them). To describe this property, we write a HyperLTL formula $\varphi_{\mathsf{OD}}$

$$\varphi_{\mathsf{OD}} \quad = \quad \forall\pi\,\forall\pi' \quad \mathrm{same}_{\mathrm{low}}(\pi,\pi') \;\rightarrow\; \mathbf{G}\,\mathrm{same}_{\mathrm{low}}(\pi,\pi') \tag{1}$$

which uses the LTL formula $\mathrm{same}_{\mathrm{low}}(\pi,\pi')$, with two free trace variables, that states that at the first instant, $\pi$ and $\pi'$ fulfil the same set of low atomic propositions.

The second property called *non-inference* [20] expresses the fact that high sensitive information are not exposed through low level one: for any trace, there must exist another one with arbitrary or dummy high sensitive information but with similar low-level information all along the computation. This property can be written as a HyperLTL formula as:

$$\varphi_{\mathsf{NI}} \quad = \quad \forall\pi\,\exists\pi' \quad \mathbf{G}\,\Lambda_{\mathrm{high}}(\pi') \wedge \mathbf{G}\,\mathrm{same}_{\mathrm{low}}(\pi,\pi') \tag{2}$$

where the formula $\Lambda_{\mathrm{high}}(\pi')$ is an LTL formula checking that the only high (i.e. non low) atomic propositions are a dummy one.

Unfortunately, most of the time, requiring strict security properties is actually ineffective as programs do not meet such requirements. To overcome this issue, it has been proposed to quantify security breaches as, e.g., in quantitative information flow [23]. Such approaches allow some information leaks provided they are acceptable, i.e. of amplitude not beyond some threshold; they have been studied in the framework of hyperproperties and hyperlogics [25, 13], in particular by considering counting extensions of HyperLTL where the number of models of a formula can be compared with some threshold.

In this article, we follow a different approach and base our work on the extensions of LTL proposed in [2]. In the latter, the authors proposed a framework enriching the syntax and semantics of LTL for reasoning about the *quality* of systems; the semantics of a formula is a value in $[0,1]$ reflecting the quality of a trace, that is, at which extent it is satisfied by this trace (0 when the formula is not satisfied at all and 1 when the formula is fully satisfied). Two ways on how to reason about quality in the context of LTL are explored there. The first one, the "propositional quality", focuses on the different ways *how* a specification can be satisfied. It is embodied by $\mathrm{LTL}_{\mathrm{prop}}$, an extension of LTL with logical operators defined as mappings over $[0,1]$. The second one, the "temporal quality", focuses on *when* eventualities are satisfied and the corresponding LTL extension is $\mathrm{LTL}_{\mathrm{temp}}$. This latter considers temporal operators with discount which makes it possible to assign different values to a formula depending on *when* certain eventualities happen[1].

We lift these qualitative extensions of LTL to the setting of hyperproperties. In the Boolean setting, the models of a hyperproperty (e.g., a HyperLTL formula) are sets of traces. Hence, a HyperLTL formula associates a Boolean value with each set of traces. To incorporate quality, a formula thus must map each set of traces to a value in $[0,1]$. We propose qualitative extensions of HyperLTL, similar to the two logics $\mathrm{LTL}_{\mathrm{prop}}$ and $\mathrm{LTL}_{\mathrm{temp}}$ wrt LTL, named respectively $\mathrm{HyperLTL}_{\mathrm{prop}}$ and $\mathrm{HyperLTL}_{\mathrm{temp}}$. For the former, coming back to our example above, the condition on observational determinism can be relaxed by requiring the difference

[1] We use different names for the two logics, to be consistent with the hyperlogics we introduce afterwards.

on low information to remain "small" all along the computation: a formula of $\text{HyperLTL}_{\text{prop}}$ thus allows one to associate with a given system a propositional quality describing guarantees about observational determinism. One may also consider as acceptable two traces that finally differ quite far from their origin: this is enabled in $\text{HyperLTL}_{\text{temp}}$.

Our main objective is to study the model checking of the hyperlogics $\text{HyperLTL}_{\text{prop}}$ and $\text{HyperLTL}_{\text{temp}}$. It consists in deciding whether the semantics of a (closed) formula is above (or below) a given threshold $v \in [0,1]$ when the set of traces in consideration is the one generated by a system (a Kripke structure). Qualitative model checking will therefore indicates to which extent the set of traces of a Kripke structure satisfies a formula.

$\text{HyperLTL}_{\text{prop}}$ is studied in Section 3. We show that the semantics of a $\text{HyperLTL}_{\text{prop}}$ formula can take only a finite set of values in $[0,1]$. We use this crucial property to design an automata-based model checking algorithm by building a non-deterministic Büchi automaton (NBA) from a $\text{HyperLTL}_{\text{prop}}$ formula for which we test emptiness. This finiteness property also implies that $\text{HyperLTL}_{\text{prop}}$ is not more expressive than HyperLTL.

For $\text{HyperLTL}_{\text{temp}}$, studied in Section 4, the situation is much more delicate as the set of possible values of a formula is no longer finite. We develop nonetheless automata-based techniques for model checking following two directions. We first solve an approximate model checking problem for $\text{HyperLTL}_{\text{temp}}$ that is correct modulo some $\epsilon$ term. We then consider fragments of $\text{HyperLTL}_{\text{temp}}$ for which we solve the exact model checking problem: the first fragment allows arbitrary quantifier prefixes but limits the interplay between negations and discounted operators. We show that in this case, the set of possible values, although infinite, enjoys a nice structure allowing automata-based model checking. For the second fragment, one considers arbitrary quantifier-free parts of formulas but forbid quantifier alternation. For these two fragments, we provide examples showing that they are still sufficient to express interesting hyperproperties on the quality of systems.

We describe our results in a weighted setting where systems (Kripke structures) that generate traces associate with each timestep a mapping giving to atomic propositions a weight in $[0,1]$, instead of just the subset of atomic propositions that currently hold. This allows one to incorporate more meaningful qualitative reasoning since basic weights come from the system description itself. For instance, weights can encode costs, time delays or probabilities. A similar setting for LTL has been considered in [10].

Other kinds of logic than temporal logic, like HyperFO [14] that is a hyper-extension of the first-order logic, could also be extended with quality. However, as advocated in [2], weighted extensions of logics like FO (or even MSO, i.e. monadic second order logic) such as wMSO [8] or wFO [3] are not very suitable to define specifications due to their undecidable nature. Hence, we believe that a weighted extension of HyperFO [14] would not fit our goals.

A longer version, including all proofs, is available [18].

## 2 Preliminaries

**Kripke structures.** We let AP be a finite set of atomic propositions. We consider *traces* that are infinite sequences of atomic propositions each weighted by a weight in $[0,1]$, i.e. elements of $([0,1]^{\text{AP}})^\omega$. We denote $\mathbb{T}$ the set of all traces, and call $\mathbb{T}_{\mathbb{W}}$ the subset of traces where all weights are taken in a finite set $\mathbb{W} \subset [0,1]$. A particular case is when $\mathbb{W} = \{0,1\}$, in which case we recover the classical traces: we call such traces *Boolean traces* in the following. Amongst all traces, we call *lassos* the ones of the form $uv^\omega$ with $u, v \in ([0,1]^{\text{AP}})^*$ (a finite word, followed by the infinite repetition of some finite word). We denote $\mathbb{L}$ the set of all lassos, and $\mathbb{L}_{\mathbb{W}}$ if weights are restricted in $\mathbb{W}$. For a trace $t \in \mathbb{T}$, we write $t[i]$ to refer to the $i$-th element in $t$, for $i \in \mathbb{N}$. We denote by $t[i,\infty]$ the suffix path starting at the $i$-th element.

A *(weighted) Kripke structure* is a tuple $\mathcal{K} = \langle S, \mathbb{W}, I, \to, L \rangle$, where $S$ is a finite set of states, $\mathbb{W} \subset [0,1]$ is a finite set of weights, $I \subseteq S$ is the set of initial states, $\to \subseteq S \times S$ is a total transition relation, and $L \colon S \to \mathbb{W}^{\mathrm{AP}}$ is a labeling function, assigning a weight to each atomic proposition in each state. The weight can encode for example costs, time delays or probabilities for the atomic propositions to be true. In case the weights are only 0 and 1, we recover the classical Kripke structures used, e.g., in the model checking of LTL: in this article, we call such structures *Boolean Kripke structures*.

A *path* $s$ in $\mathcal{K}$ is a sequence $s = s_0 s_1 \cdots$ with $s_0 \in S$ and $s_i \to s_{i+1}$ for all $i \in \mathbb{N}$. Since $\to$ is total, there are no deadlocks and thus all paths are of infinite length. The set of all *traces* in $\mathcal{K}$, denoted $\mathbb{T}(\mathcal{K})$, is defined as the set of traces $t \in \mathbb{T}_{\mathbb{W}}$ of the form $t = L(s_0)L(s_1)\cdots$ for a path $s_0 s_1 \cdots$ in $\mathcal{K}$. $\mathbb{L}(\mathcal{K})$ denotes the set of all lassos of $\mathcal{K}$, that is, traces from $\mathbb{T}(\mathcal{K})$ that are lassos.

**Büchi automata.** The model checking of LTL as well as HyperLTL, consisting in checking whether a given (Boolean) Kripke structure satisfies a given closed formula, relies on automata-based techniques. We recall here the automata used in this context.

A *non-deterministic Büchi automaton* (NBA) is a tuple $\mathcal{A} = \langle \Sigma, Q, Q_0, \delta, F \rangle$, where $\Sigma$ is a finite alphabet, $Q$ is a finite set of *states*, $Q_0 \subseteq Q$ is the set of *initial states*, $\delta : Q \times \Sigma \to 2^Q$ is a *transition function* and $F \subseteq Q$ is the set of *accepting states*. An infinite sequence $r = r_0 r_1 \cdots$ of states is called a *run* on a word $w = w_1 w_2 \cdots \in \Sigma^\omega$ if $r_0 \in Q_0$ and for every $i \geq 0$, $r_{i+1} \in \delta(r_i, w_{i+1})$. An NBA *accepts* a word if there exists a run $r$ on it that visits $F$ infinitely often, i.e. for $\inf(r) = \{q \mid r_i = q \text{ for infinitely many } i \in \mathbb{N}\}$, it holds that $\inf(r) \cap F \neq \emptyset$. The accepted language of an NBA, denoted by $\mathcal{L}(\mathcal{A})$, is the set of words accepted by $\mathcal{A}$.

The emptiness problem of an NBA, i.e. deciding whether the accepted language is empty, is decidable in linear time and is NLOGSPACE-complete [16].

Given a Kripke structure $\mathcal{K}$ defined over a set AP of atomic propositions and a finite set of weights $\mathbb{W}$, one can define an NBA $\mathcal{A}_{\mathcal{K}}$ over the alphabet $\mathbb{W}^{\mathrm{AP}}$ accepting the language $\mathbb{T}(\mathcal{K})$.

**HyperLTL.** The logic HyperLTL, a hyperlogic extension of Linear Time Logic (LTL) was introduced in [6]. It extends LTL with universal and existential quantifiers over traces allowing to express properties on several traces. Models for HyperLTL formulas are then sets of traces and not individual traces. Hence, the set of all models of a HyperLTL formula is a set of sets of traces, a hyperproperty.

We let $\mathcal{V}$ be the infinite set of (trace) variables. The *formulas of* HyperLTL are defined by the following grammar, for all atomic propositions $p \in \mathrm{AP}$ and variables $\pi \in \mathcal{V}$:

$$\psi ::= \exists \pi\, \psi \mid \forall \pi\, \psi \mid \varphi \qquad\qquad \varphi ::= p_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{U}\, \varphi$$

As usual, additional operators can be derived : $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1) \vee (\neg\varphi_2))$, $\mathrm{true} = p_\pi \vee \neg p_\pi$ and $\mathrm{false} = \neg\mathrm{true}$, $\varphi_1 \to \varphi_2 = \neg\varphi_1 \vee \varphi_2$, $\varphi_1 \leftrightarrow \varphi_2 = (\varphi_1 \to \varphi_2) \vee (\varphi_2 \to \varphi_1)$, the *Finally* operator $\mathbf{F}\, \varphi = \mathrm{true}\, \mathbf{U}\, \varphi$, and the *Globally* operator $\mathbf{G}\, \varphi = \neg\, \mathbf{F}(\neg\varphi)$.

If $\psi = Q_1 \pi_1 \cdots Q_n \pi_n \varphi$ for $Q_i \in \{\exists, \forall\}$ for $i \in \{1, \ldots, n\}$ and $\varphi$ quantifier-free, we call $\varphi$ the *quantifier-free part* of $\psi$.

We know explain the semantics of HyperLTL formulas over (weighted) traces. This differs from the classical semantics of HyperLTL associating with each formula a set of traces. We recover an equivalent definition in the case of a Boolean Kripke structure. Given a trace

assignment $\Pi$ over $\mathbb{T}$, mapping trace variables to traces, and a non-empty set of traces $T \subseteq \mathbb{T}$ used as a ranging set for quantified variables[2], the semantics $[\![\psi]\!]_T(\Pi)$ of a formula $\psi$, called the *satisfaction value* of $\psi$ over $T$ given $\Pi$, is the weight in $[0,1]$ defined recursively as:

$$[\![p_\pi]\!]_T(\Pi) = \Pi(\pi)[0](p) \qquad [\![\neg\varphi]\!]_T(\Pi) = 1 - [\![\varphi]\!]_T(\Pi)$$
$$[\![\varphi_1 \vee \varphi_2]\!]_T(\Pi) = \max([\![\varphi_1]\!]_T(\Pi), [\![\varphi_2]\!]_T(\Pi)) \qquad [\![\mathbf{X}\,\varphi]\!]_T(\Pi) = [\![\varphi]\!]_T(\Pi[1,\infty])$$
$$[\![\varphi_1 \,\mathbf{U}\, \varphi_2]\!]_T(\Pi) = \sup_{i\geq 0} \Big( \min \big( [\![\varphi_2]\!]_T(\Pi[i,\infty]), \min_{0\leq j<i} [\![\varphi_1]\!]_T(\Pi[j,\infty]) \big) \Big)$$
$$[\![\forall\pi\,\psi]\!]_T(\Pi) = \inf_{t\in T} [\![\psi]\!]_T(\Pi[\pi \mapsto t]) \qquad [\![\exists\pi\,\psi]\!]_T(\Pi) = \sup_{t\in T} [\![\psi]\!]_T(\Pi[\pi \mapsto t])$$

where $\Pi[\pi \mapsto t]$ is the trace assignment identical to $\Pi$ except that it maps $\pi$ to $t$, and $\Pi[i,\infty]$ denotes the suffix trace assignment, defined for all variables $\pi$ as $\Pi[i,\infty](\pi) = \Pi(\pi)[i,\infty]$. In case of a trace assignment that only contains Boolean traces (that we call *Boolean assignments*), it is easy to see that the semantics of HyperLTL formulas coincides with the classical semantics of HyperLTL, i.e. maps all Boolean assignments to 0 or 1 such that the set of assignments mapped to 1 are the ones satisfying the formula.

As one may expect, for the existentially quantified formula $\exists\pi\,\psi$, when $\psi$ in which $\pi$ is set to some trace $t$ is evaluated to 1, then so is $\exists\pi\,\psi$. Hence, existential quantification is intuitively a disjunction over the set of all traces from T and thus, corresponds to a supremum. Universal quantification is defined dually as an infimum.

The semantics of the additional operators $\mathbf{F}$ and $\mathbf{G}$ are as follows: $[\![\mathbf{F}\,\varphi]\!]_T(\Pi) = \sup_{i\geq 0}[\![\varphi]\!]_T(\Pi[i,\infty])$ and $[\![\mathbf{G}\,\varphi]\!]_T(\Pi) = \inf_{i\geq 0}[\![\varphi]\!]_T(\Pi[i,\infty])$.

For a closed formula $\psi$ (i.e. without free variables), the satisfaction value of $\psi$ over a Kripke structure $\mathcal{K}$ is the value $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}([\,])$ (denoted simply $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}$ from now on) taken over the empty assignment $[\,]$.

# 3 Propositional quality for HyperLTL

## 3.1 Definitions and examples

In the same way that HyperLTL is built upon LTL, $\text{HyperLTL}_{\text{prop}}$ corresponds to the hyper-logic extension of the linear temporal logic $\text{LTL}_{\text{prop}}$, for *LTL with propositional quality* [2].[3] This logic generalises LTL by replacing the Boolean operators by a set $\mathcal{F}$ of arbitrary functions over [0,1]. Hence, the syntax of LTL is enriched by the construct $f(\varphi_1, \ldots, \varphi_k)$ for $f \in \mathcal{F}$, the temporal operators remaining the same. Strictly speaking, it is then a family of logics, parameterized by $\mathcal{F}$. Hence, beyond classical Boolean operators $(\vee, \wedge, \neg)$ that can obviously be defined as such functions (and that we thus suppose to always be members of $\mathcal{F}$), the set $\mathcal{F}$ can contain quantitative functions like $\oplus_\alpha$ and $\triangledown_\alpha$ for $\alpha \in [0,1]$, that associates with $x$ and $y$ the weighted sum $x \oplus_\alpha y = \alpha x + (1-\alpha)y$ and the scalar multiplication $\triangledown_\alpha x = \alpha x$. Intuitively, the functions in $\mathcal{F}$ allow one to give weights to subformulas, i.e. parts of our specification, to moderate their "effect" in the formula where they occur. Let us point out that with such operators the semantics for formulas indeed belongs to the interval $[0,1]$.

To reason about the propositional quality of hyperproperties, we introduce $\text{HyperLTL}_{\text{prop}}$, a logic generalising both HyperLTL and $\text{LTL}_{\text{prop}}$. A similar attempt has been achieved in [11] to lift $\text{LTL}_{\text{prop}}$ to the setting of *alternating-time temporal logic* to be able to quantify over strategies of agents in a multi-agent system: the same kind of fuzzy functions are added to talk about the quality in strategic reasoning.

[2] Notice that we do not require $\Pi$ to map variables to the restricted set of traces $T$.
[3] This logic is called $\text{LTL}[\mathcal{F}]$ in [2].

▶ **Definition 1.** *Assuming $\mathcal{F}$ is a finite set of function symbols, the syntax of* $\text{HyperLTL}_{\text{prop}}$ *is given by*

$$\psi ::= \forall \pi\, \psi \mid \exists \pi\, \psi \mid \varphi \qquad\qquad \varphi ::= \text{true} \mid \text{false} \mid p_\pi \mid f(\varphi, \ldots, \varphi) \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{U}\, \varphi$$

*for all $p \in \text{AP}$, $\pi \in \mathcal{V}$, and $f \in \mathcal{F}$. These functions $f$ being interpreted as mappings from $[0,1]^k$ to $[0,1]$ (for functions of arity $k$), the* HyperLTL *semantics is extended for* $\text{HyperLTL}_{\text{prop}}$ *as*

$$[\![f(\varphi_1, \ldots, \varphi_k)]\!]_T(\Pi) = f([\![\varphi_1]\!]_T(\Pi), \ldots, [\![\varphi_k]\!]_T(\Pi))$$

*for all trace assignments $\Pi$ over $\mathbb{T}$, and non-empty sets of traces $T \subseteq \mathbb{T}$.*

We give two examples showing that this extension of HyperLTL with propositional quality allows to model hyperproperties with a very different flavour than other quantitative extensions of HyperLTL (like counting extensions considered in [13]). The two first examples deal with Boolean Kriple structures, while the last one is a more general example that can be applied to a weighted Kripke structure.

▶ **Example 2.** We propose some quantitative versions of the *observational determinism* property presented in the introduction. First, let us formally define the subformula $\text{same}_{\text{low}}(\pi, \pi')$ of the formula $\varphi_{\mathsf{OD}}$, having two free variables $\pi$ and $\pi'$, as $\bigwedge_{p \in \text{low}} p_\pi \leftrightarrow p_{\pi'}$, where low is the subset of "low sensitivity" atomic propositions. To compute the propositional quality of observational determinism, we adapt this formula by using the $\oplus$ weighted function in order to express the ratio of the agreement of two traces at the current step: $\text{ratio}_{\text{low}}(\pi, \pi') := \bigoplus_{p \in \text{low}} \frac{1}{|\text{low}|}(p_\pi \leftrightarrow p_{\pi'})$. Then, the formula $\mathbf{G}\, \text{ratio}_{\text{low}}(\pi, \pi')$ aggregates these values at every step by considering an infimum. Thus, we can modify formula $\varphi_{\mathsf{OD}}$ to compute the smallest total ratio separating two traces of the system that agree on the first step, thus relaxing the Boolean observational determinism:

$$\forall \pi\, \forall \pi' \quad \text{same}_{\text{low}}(\pi, \pi') \rightarrow \mathbf{G}\, \text{ratio}_{\text{low}}(\pi, \pi')$$

One may use the interval $[0,1]$ in a less strict way and split it into several pieces, each of them encoding different "truth" status of the specification. For instance, we can refine this formula by separating the values 1 obtained when the system non trivially fulfils observational determinism from the degenerated case where there are no two different traces which agree on the low atomic propositions on the first step:

$$\varphi_{\mathsf{OD}}^{\text{prop}} \quad = \quad \forall \pi\, \forall \pi' \quad \nabla_{\frac{1}{2}}\, \mathbf{G}\, \text{ratio}_{\text{low}}(\pi, \pi') \vee \nabla_{\frac{3}{4}}(\neg \text{same}_{\text{low}}(\pi, \pi') \vee \pi = \pi')$$

where the subformula $\pi = \pi'$ is used as a shortcut for the Boolean formula $\mathbf{G}(\bigwedge_{p \in \text{AP}} p_\pi \leftrightarrow p_{\pi'})$. Here we get half of the *smallest ratio of corresponding low atomic propositions* (and thus a value in interval $[0, 1/2]$), if there are two different traces which agree on the low atomic propositions of the first step, or $\frac{3}{4}$ (any value outside of $[0, 1/2]$ could be convenient) if there are no such traces. The universal quantifications in the formula correspond to some minimisation and thus enable to consider the worst-case scenario. We could also consider the formula $\forall \pi\, \forall \pi' \quad \text{same}_{\text{low}}(\pi, \pi') \rightarrow \mathbf{F}\, \mathbf{G}(\text{ratio}_{\text{low}}(\pi, \pi'))$ with the addition of an operator $\mathbf{F}$ to compute a lim sup ratio, allowing for traces to vary a lot at the beginning, as long as they become close on the long term.

▶ **Example 3.** We present also a modification of the *non-inference* property $\varphi_{\mathsf{NI}}$ from the introduction. First, the subformula $\Lambda_{\text{high}}(\pi')$ can be described as $\lambda_{\pi'} \wedge \bigwedge_{p \notin \text{low} \cup \{\lambda\}} \neg p_{\pi'}$. Then, to describe the propositional quality of non-inference, we could ask whether, for every

trace, we can find another trace with $\lambda$ as the only high atomic proposition, but which behaves *almost* the same for a low user. This approximation in the quality can be obtained by using a new weighted function $\mathit{threshold}_{>k}(x)$, with $k \in [0,1]$ that equals 1 if $x$ is greater than $k$, and 0 otherwise. The non-inference can then be relaxed as follows:

$$\varphi_{\mathsf{NI}}^{\mathrm{prop}} \quad = \quad \forall\pi\,\exists\pi' \quad \mathbf{G}\,\Lambda_{\mathrm{high}}(\pi') \wedge \mathit{threshold}_{>k}(\mathrm{ratio}_{\mathrm{low}}(\pi,\pi'))$$

The semantics of the formula is Boolean, though it uses weighted functions to compute it.

▶ **Example 4.** As a last example, we demonstrate what happens when we deal with a weighted Kripke structure that maps every atomic proposition to a probability that it holds in a given state. We can replace the formula $\mathrm{ratio}_{\mathrm{low}}(\pi,\pi')$ of Example 2 by its probabilistic version: $\bigwedge_{p\in\mathrm{low}} f(p_\pi, p_{\pi'})$, where $f$ is the mapping defined for $x, y \in [0,1]$ by $f(x,y) = xy + (1-x)(1-y)$, giving the probability that $p_\pi$ and $p_{\pi'}$ are the same, $x$ being the probability of $p_\pi$, and $y$ of $p_{\pi'}$. By considering the formula $\mathbf{G}\left(\bigwedge_{p\in\mathrm{low}} f(p_\pi, p_{\pi'})\right)$, we thus compute the smallest probability over all steps that low atomic propositions coincide. The formula below thus computes the smallest such probability over any two traces:

$$\forall\pi\,\forall\pi' \quad \mathrm{same}_{\mathrm{low}}(\pi,\pi') \to \mathbf{G}\left(\bigwedge_{p\in\mathrm{low}} f(p_\pi, p_{\pi'})\right)$$

If the semantics of this formula is 1, the system fulfils observational determinism. If the semantics of this formula is 0, the system contains two traces where a certain proposition $p$ differs with probability 1 at some position. The values between 0 and 1 give a probabilistic quality of observational determinism: value $1/2$, e.g., means that in any two traces, at any position, and for any low atomic proposition, the probability that the proposition coincides is at least $1/2$.

## 3.2 A finite-valued property

We fix in this section a finite set $\mathbb{W}$ of weights in $[0,1]$. Even though the family $\mathcal{F}$ may contain functions ranging over the whole interval $[0,1]$, it turns out that every $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula $\psi$ enjoys a finite-valued property: it has only a finite number of possible satisfaction values, that is $\{[\![\psi]\!]_T(\Pi) \mid \Pi\colon \mathcal{V} \rightharpoonup \mathbb{T}_{\mathbb{W}},\ T \subseteq \mathbb{T}_{\mathbb{W}} \text{ non empty}\}$ (denoted $V_{\mathbb{W}}([\![\psi]\!])$ from now on) is finite. Moreover, this set can be computationally over-approximated. This has been shown for $\mathrm{LTL}_{\mathrm{prop}}$ in [2, Lemma 2.5] (when $\mathbb{W} = \{0,1\}$), and we extend the proof here to $\mathrm{HyperLTL}_{\mathrm{prop}}$; this property will be used afterwards in our model checking algorithm.

▶ **Definition 5.** *For any* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *formula* $\psi$*, the set* $V_{\mathbb{W}}(\psi)$ *included in* $[0,1]$ *is defined inductively by:*

$$V_{\mathbb{W}}(p_\pi) = \mathbb{W} \cup \{0,1\} \qquad V_{\mathbb{W}}(\mathbf{X}\,\varphi) = V_{\mathbb{W}}(\varphi) \qquad V_{\mathbb{W}}(\varphi_1\,\mathbf{U}\,\varphi_2) = V_{\mathbb{W}}(\varphi_1) \cup V_{\mathbb{W}}(\varphi_2)$$
$$V_{\mathbb{W}}(f(\varphi_1,\dots,\varphi_k)) = \{f(v_1,\dots,v_k) \mid v_1 \in V_{\mathbb{W}}(\varphi_1),\dots,v_k \in V_{\mathbb{W}}(\varphi_k)\}$$
$$V_{\mathbb{W}}(\forall\pi\,\psi) = V_{\mathbb{W}}(\exists\pi\,\psi) = V_{\mathbb{W}}(\psi)$$

*for* $p \in \mathrm{AP}$*,* $\pi \in \mathcal{V}$*,* $f \in \mathcal{F}$*.*

This set $V_{\mathbb{W}}(\psi)$ turns out to be finite as, essentially, the number of approximated satisfaction values only depends on $\#\psi$, the number of occurrences of atomic propositions in the formula $\psi$. For instance, $\#(p_\pi\,\mathbf{U}\,f(p_\pi, q_\pi, p_{\pi'})) = 4$ since the formula uses atomic propositions $p_\pi$ (twice), $q_\pi$ and $p_{\pi'}$.

▶ **Lemma 6.** *For all formulas $\psi$ of* $\mathrm{HyperLTL}_{\mathrm{prop}}$, $V_{\mathbb{W}}([\![\psi]\!]) \subseteq V_{\mathbb{W}}(\psi)$ *and* $|V_{\mathbb{W}}(\psi)| \leq |\mathbb{W}|^{\#\psi}$.

The exponential bound is tight, since it is already so for $\mathrm{LTL}_{\mathrm{prop}}$ [2]. In general the inclusion $V_{\mathbb{W}}([\![\psi]\!]) \subseteq V_{\mathbb{W}}(\psi)$ is strict (see [18] for an example). Indeed, it is not surprising that the set $V_{\mathbb{W}}(\psi)$ is a strict over-approximation of the possible satisfaction values of $\psi$ as the satisfiability for HyperLTL (and thus of $\mathrm{HyperLTL}_{\mathrm{prop}}$) is undecidable [12]. Note that, as a consequence of this result, in the semantics of $\mathrm{HyperLTL}_{\mathrm{prop}}$, the various occurrences of operators inf and sup can be replaced by min and max respectively.

## 3.3 The model checking problem for $\mathbf{HyperLTL}_{\mathbf{prop}}$

The natural generalisation of the model checking of HyperLTL to $\mathrm{HyperLTL}_{\mathrm{prop}}$ is as follows

▶ **Definition 7.** *Given a Kripke structure $\mathcal{K}$, a closed* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *formula $\psi$, and a rational threshold $v \in [0,1]$, the model checking problem is to decide whether $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$ (or whether $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$, depending on the cases of interest).*

We propose a model checking algorithm for $\mathrm{HyperLTL}_{\mathrm{prop}}$. Its complexity will be non elementary with a tower of exponentials depending on the number of quantifier alternations of the HyperLTL formula. It relies, as a base case, on the construction proposed in [2] (Theorem 2.9 for the Boolean case, and section 3.1 for the extension to weighted Kripke structures) to translate a formula of $\mathrm{LTL}_{\mathrm{prop}}$ run on a Kripke structure into an NBA:

▶ **Proposition 8** ([2])**.** *Let $\varphi$ be an* $\mathrm{LTL}_{\mathrm{prop}}$ *formula, $\mathcal{K}$ be a Kripke structure, and $P \subseteq [0,1]$. There exists an NBA $\mathcal{A}^P_{\mathcal{K},\varphi}$ such that for every trace $t \in \mathbb{T}(\mathcal{K})$, it holds that $[\![\varphi]\!](t) \in P$ iff $\mathcal{A}^P_{\mathcal{K},\varphi}$ accepts $t$. Furthermore, $\mathcal{A}^P_{\mathcal{K},\varphi}$ has at most $|\mathbb{W}|^{|\varphi|^2}|\varphi|$ states where $\mathbb{W}$ is the set of weights in $\mathcal{K}$, and $|\varphi|$ is the size of $\varphi$.*

In this proposition, the set $P$ can take various forms, provided that testing membership in $P$ of values in $[0,1]$ is decidable, so that the NBA $\mathcal{A}^P_{\mathcal{K},\varphi}$ is computable. For instance, if $P$ is of the form $[0,a)$ or $(a,1]$, with some rational number $a$, the membership test can be performed.

From this, to decide the model checking problem $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$ for a given $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula $\psi$ and a Kripke structure $\mathcal{K}$, we build an NBA $\mathcal{A}^{[0,v)}_{\mathcal{K},\psi}$ and check the emptiness of its language. We base the construction of this NBA on two crucial ideas used before separately. First, we make use of the NBA from Proposition 8 on a Krikpe structure obtained as a product of multiple copies of $\mathcal{K}$ (to deal with the quantified variables of $\psi$), by treating the quantifier-free part of a $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula as an $\mathrm{LTL}_{\mathrm{prop}}$ formula (as done in [6] for HyperLTL model checking). Second, following the semantics of a quantified formula, we compute inf and sup over $\mathbb{T}(\mathcal{K})$, by checking finitely many values, as $V_{\mathbb{W}}(\psi)$ is finite (with $\mathbb{W}$ the set of weights in $\mathcal{K}$).

For the first step, we explain how to treat a non-quantified formula of $\mathrm{HyperLTL}_{\mathrm{prop}}$ as a formula of $\mathrm{LTL}_{\mathrm{prop}}$ over an extended set of atomic propositions. Formally, for all $n \in \mathbb{N}$, we will consider trace variables $\pi_1, \ldots, \pi_n$, and denote by $\mathrm{AP}_i = \{p_{\pi_i} \mid p \in \mathrm{AP}\}$ the set of atomic propositions indexed by the trace variable $\pi_i$. We then let $\mathrm{AP}_{\leq n}$ be the set $\bigcup_{1 \leq i \leq n} \mathrm{AP}_i$. Notice that $\mathrm{AP}_{\leq 0}$ is empty. Hence, a non-quantified formula $\psi$ of $\mathrm{HyperLTL}_{\mathrm{prop}}$, having $\pi_1, \ldots, \pi_m$ as free variables, can be considered as a formula of $\mathrm{LTL}_{\mathrm{prop}}$, a trace assignment being encoded as a single trace as follows:

▶ **Definition 9.** *Let $\Pi = [\pi_1 \mapsto t_1, \ldots, \pi_n \mapsto t_n]$ be a trace assignment with $t_i \in \mathbb{T}(\mathcal{K})$ for all $1 \leq i \leq n$. The trace $t_\Pi$ is an element of $(\mathbb{W}^{\mathrm{AP}_{\leq n}})^\omega$ and is defined for all $k \in \mathbb{N}$, for all $1 \leq i \leq n$, for all $p \in \mathrm{AP}$, by $(t_\Pi[k])(p_{\pi_i}) = (t_i[k])(p)$.*

By an inductive generalisation of Proposition 8, we are then able to obtain:

▶ **Proposition 10.** *Let $\psi$ be a* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *formula, $P \subseteq [0,1]$, and $\mathcal{K}$ be a Kripke structure. We can build an NBA $\mathcal{A}^P_{\mathcal{K},\psi}$ such that $\mathcal{L}(\mathcal{A}^P_{\mathcal{K},\psi})$ is the set of traces $t_\Pi$, with $\Pi$ an assignment of free variables to traces of $\mathcal{K}$, such that $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) \in P$. The size of $\mathcal{A}^P_{\mathcal{K},\psi}$ is non-elementary, with the tower of exponentials of linear height in the number of quantifier alternations of $\psi$.*

Note that for a closed formula and thus, the empty trace assignment, the NBA $\mathcal{A}_{\mathcal{K},\psi}$ can accept at most the unique trace $t_{[]}$ defined as $u^\omega$ with $u$ the mapping of empty domain. As a corollary, for a closed $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula $\psi$, we obtain:

▶ **Theorem 11.** *The model checking problem of* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *is decidable with a non-elementary complexity.*

The height of the tower of exponentials in the non elementary complexity depends only on the number of quantifier alternations, similarly to the case of HyperLTL [6]. Notice, as a final corollary, that we also obtain the decidability of the model checking problem for HyperLTL over weighted Kripke structures (which was not known so far, to the best of our knowledge), since HyperLTL is a fragment of $\mathrm{HyperLTL}_{\mathrm{prop}}$:

▶ **Corollary 12.** *The model checking problem of* HyperLTL *(over weighted Kripke structures) is decidable with a non-elementary complexity.*

## 3.4 Expressiveness of $\mathrm{HyperLTL}_{\mathrm{prop}}$ in the Boolean case

In this section, we focus on Boolean Kripke structures, and we show that any $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula can be translated into a somewhat equivalent HyperLTL formula. This is an extension of the analogous result for $\mathrm{LTL}_{\mathrm{prop}}$ and LTL [2]. This implies $\mathrm{HyperLTL}_{\mathrm{prop}}$ is not more expressive than HyperLTL. However, it makes it a lot easier to express quantitative specifications, since the resulting HyperLTL formulas get very big and not very intuitive.

▶ **Theorem 13.** *For every $P \subseteq [0,1]$ and* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *formula $\psi$, there exists a* HyperLTL *formula* $\mathrm{Bool}(\psi, P)$ *such that for all subset $T \subseteq \mathbb{T}_{\{0,1\}}$ and trace assignments $\Pi$, $[\![\psi]\!]_T(\Pi) \in P$ iff $[\![\mathrm{Bool}(\psi,P)]\!]_T(\Pi) = 1$. Moreover,* $\mathrm{Bool}(\psi, P)$ *has size at most $O(2^{(k+2)|\varphi|})$ if $k$ is the quantifier depth of $\varphi$, and $\varphi$ is the quantifier-free part of $\psi$.*

The proof of this theorem heavily uses the fact that the formula $\psi$ has only finitely many possible satisfaction values and thus, that only those which also belong to $P$ matter. Hence, as a first step, it is possible to consider $P$ by means of each of its elements $c$ that is also a possible satisfaction value of $\psi$. Then, recursively it is possible to define $\mathrm{Bool}(\psi, \{c\})$ as a Boolean combination of some $\mathrm{Bool}(\varphi_i, P_i)$ where $\varphi_i$ is an subformula of $\psi$ and $P_i$ is a subset of $P$ defined relatively to $c$.

Some Boolean functions can only be expressed by exponential size Boolean formulas [21]. Therefore, as noticed in [2] already, the translation of an $\mathrm{LTL}_{\mathrm{prop}}$ formula to an equivalent LTL formula can grow exponentially: in our setting, there exists a sequence of quantifier-free formulas $\varphi$, of growing length, and a predicate $P \subseteq [0,1]$ such that $|\mathrm{Bool}(\varphi, P)| \in \Omega(2^{|\varphi|})$. In [2, Theorem 2.8], again in the case of $\mathrm{LTL}_{\mathrm{prop}}$, a thorough discussion on the complexity of computing $\mathrm{Bool}(\varphi, P)$ explains that the computation can be performed in PSPACE (under reasonable assumptions on the complexity of the functions $f$ that appear in the formula), and that a super-polynomial blow-up (i.e. at least $|\varphi|^c$ for all constants $c$) is unavoidable.

The above translation of a $\text{HyperLTL}_{\text{prop}}$ formula $\psi$ into an equivalent Boolean HyperLTL formula (whose model checking is non elementary [6]) yields an alternative model checking algorithm for $\text{HyperLTL}_{\text{prop}}$, with non-elementary complexity. However the height of exponentials depends on the number of quantifiers in $\psi$ (instead on the number of quantifier alternations in Theorem 10); indeed, the number of alternations in $\text{Bool}(\psi, P)$ depends on the number of quantifiers in the original formula $\psi$ (and not only on its number of alternations).

# 4 Temporal quality for HyperLTL

## 4.1 Definitions and examples

In this section we consider another qualitative extension of HyperLTL named $\text{HyperLTL}_{\text{temp}}$ (for HyperLTL *with temporal quality*). This logic is based upon $\text{LTL}_{\text{temp}}$, another qualitative extension of LTL proposed in [2][4]. This latter generalises LTL by considering new temporal operators $\mathbf{U}_\eta$ where $\eta$ is taken from a set $\mathcal{D}$ of discounting sequences which parameterises the logic: $(\eta_i)_{i\in\mathbb{N}}$ is a *discounting sequence* if for all $i$, $\eta_i \in [0,1]$, $\lim_{i\to\infty} \eta_i = 0$, and $\eta$ is decreasing. Note, that this implies in particular that $\eta_i > 0$ for all $i \in \mathbb{N}$. Common examples for discounting sequences are $\eta_i = \alpha^i$, for some $\alpha \in (0,1)$, and $\eta_i = \frac{1}{i+1}$.

In contrast to propositional quality which allows to mitigate the weights of subformulas, temporal quality aims to give weights depending on *when* events happen. Intuitively, for a formula $\varphi_1 \,\mathbf{U}_\eta\, \varphi_2$, the further $\varphi_2$ is considered the least its value will impact the value of $\varphi_1 \,\mathbf{U}_\eta\, \varphi_2$.

▶ **Definition 14.** *Assuming $\mathcal{D}$ is a finite set of discounting sequences, the syntax of* $\text{HyperLTL}_{\text{temp}}$ *is given by*

$$\psi ::= \forall \pi\, \psi \mid \exists \pi\, \psi \mid \varphi \qquad\qquad \varphi ::= p_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \varphi\,\mathbf{U}_\eta\,\varphi$$

*for all $p \in \text{AP}$, $\pi \in \mathcal{V}$, and $\eta \in \mathcal{D}$. The* HyperLTL *semantics is extended for* $\text{HyperLTL}_{\text{temp}}$ *as*

$$[\![\varphi_1 \,\mathbf{U}_\eta\, \varphi_2]\!]_T(\Pi) = \sup_{i\geq 0}\Big(\min\big(\eta_i\,[\![\varphi_2]\!]_T(\Pi[i,\infty]),\ \min_{0\leq j<i}(\eta_j\,[\![\varphi_1]\!]_T(\Pi[j,\infty]))\big)\Big)$$

*for all trace assignments $\Pi$ over $\mathbb{T}$, and a non-empty set of traces $T \subseteq \mathbb{T}$.*

We add as syntactic sugar the operators $\mathbf{F}_\eta\,\varphi = \text{true}\,\mathbf{U}_\eta\,\varphi$ and $\mathbf{G}_\eta\,\varphi = \neg\,\mathbf{F}_\eta\,\neg\varphi$. Their semantics can be simplified as

$$[\![\mathbf{F}_\eta\,\varphi]\!]_T(\Pi) = \sup_{i\geq 0}(\eta_i\,[\![\varphi]\!]_T(\Pi[i,\infty])) \quad\text{and}\quad [\![\mathbf{G}_\eta\,\varphi]\!]_T(\Pi) = \inf_{i\geq 0}(1-\eta_i\,(1-[\![\varphi]\!]_T(\Pi[i,\infty])))$$

We propose now two examples showing this extension of HyperLTL with temporal quality allows to model hyperproperties; again it has a very different flavour than other quantitative extensions of HyperLTL as the ones from [13], for instance:

▶ **Example 15.** We can formulate another version of observational determinism in the logic $\text{HyperLTL}_{\text{temp}}$. We refine $\varphi_{\mathsf{OD}}$ by replacing the global operator with a discounted one:

$$\varphi_{\mathsf{OD}}^{\text{temp}} \quad = \quad \forall\pi\,\forall\pi' \quad \underbrace{\text{same}_{\text{low}}(\pi,\pi') \ \rightarrow\ \mathbf{G}_\eta\,\text{same}_{\text{low}}(\pi,\pi')}_{\varphi'} \tag{3}$$

[4] This logic is called $\text{LTL}^{Disc}[\mathcal{D}]$ in that paper.

Note that, since discounted sequences are injective, the subformula $\varphi'$ gives information about the moment where the two traces $\pi$ and $\pi'$ disagree on low atomic propositions. The prefix of universal quantifiers captures this earliest moment in the full system. This could be of practical interest since the further in the future we leak information about secrets, the less probable a malicious adversary will dedicate that much time on an attack. So this amounts to verify that the satisfaction value of $\varphi_{\mathsf{OD}}^{\mathrm{temp}}$ is greater than a certain value.

▶ **Example 16.** We can also formulate another version of non-inference in $\mathrm{HyperLTL}_{\mathrm{temp}}$. We refine the formula $\varphi_{\mathsf{NI}}$ by discounting the second part of the formula on the equality of low atomic propositions:

$$\varphi_{\mathsf{NI}}^{\mathrm{temp}} \quad = \quad \forall\pi\,\exists\pi' \quad \mathbf{G}\,\Lambda_{\mathrm{high}}(\pi') \wedge \mathbf{G}_{\eta}\,\mathrm{same}_{\mathrm{low}}(\pi,\pi') \tag{4}$$

For a Boolean Kripke structure $\mathcal{K}$, it computes either 1 if the non-inference is satisfied, or $1-\eta_i$ where $i$ corresponds to the greatest value where every possible trace $t \in \mathbb{T}(\mathcal{K})$ has a trace $t' \in \mathbb{T}(\mathcal{K})$ satisfying $\mathbf{G}\,\Lambda_{\mathrm{high}}(t')$ and $\mathrm{same}_{\mathrm{low}}(t,t')$ until step $i$. Hence, it computes a value giving us the longest possible time where the non-inference is satisfied in $\mathcal{K}$.

## 4.2 Model checking for $\mathrm{HyperLTL}_{\mathrm{temp}}$

The model checking for $\mathrm{HyperLTL}_{\mathrm{temp}}$ aims to compare with some threshold the semantical value of a formula when applied to the set of traces of some Kripke structure.

For computability reasons, we assume from now on that the discounting sequences $(\eta_i)_{i\in\mathbb{N}}$ are recursively enumerable sequences of rational numbers. Being recursively enumerable and strictly decreasing, the elements of such a sequence form therefore a recursive set of rationals.

Notice that, even in the case of Boolean Kripke structures, there is no hope to apply the strategy used in Section 3.4 to model check $\mathrm{HyperLTL}_{\mathrm{prop}}$ by translating a formula and a threshold to an equivalent Boolean HyperLTL formula. Indeed, it is already impossible to find an equivalent LTL formula for an $\mathrm{LTL}_{\mathrm{temp}}$ formula, since $\mathrm{LTL}_{\mathrm{temp}}$ formulas can generate non $\omega$-regular languages, as noticed in [2] and recalled in the example below.

▶ **Example 17.** Consider for instance the formula $\varphi = \mathbf{G}(\mathbf{F}_{\eta}\,p) \vee \mathbf{F}\,\mathbf{G}\,\neg p$. For a trace $t \in \mathbb{T}_{\{0,1\}}$, either $[\![\varphi]\!](t) = 1$ if $p$ is seen only finitely often in $t$ (with weight 1), or $[\![\varphi]\!](t) = \liminf_k \eta_{i_{k+1}-i_k-1}$ where $(i_k)_{k\in\mathbb{N}}$ is the sequence of positions where $p$ holds in $t$. In particular, $[\![\varphi]\!](t)$ is always different from 0 when $t$ is a lasso trace; however $[\![\varphi]\!](t)$ could be equal to 0 for some non-lasso trace $t$, for instance when $i_k = k^2$. This shows that the language of the traces $t$ such that $[\![\varphi]\!](t) = 0$ is not $\omega$-regular as it is non empty, yet does not contain any lasso trace (although, there exist lassos with a semantics arbitrarily close to 0).

Nonetheless, the notion of lassos remains useful regarding model checking purposes and we introduce the notion of *lasso assignments*:

▶ **Definition 18.** *A trace assignment $\Pi$ is a lasso assignment if for all trace variables $\pi$ where it is defined, $\Pi(\pi) \in \mathbb{L}$. Equivalently, the associated trace $t_\Pi$ is also a lasso.*

Over lasso assignments, we show that Büchi automata can be used to describe the set of traces satisfying some $\mathrm{HyperLTL}_{\mathrm{temp}}$ formula wrt some threshold. This will also imply that, on a semantical point of view, formulas coincide when interpreted over traces and over lassos, and thus, will provide some automata-based tools to decide model checking for some fragments of $\mathrm{HyperLTL}_{\mathrm{temp}}$.

**Automata models for $\mathbf{HyperLTL_{temp}}$.** The following result from [2] (Theorem 4.6, and section 5.1 for the extension to weighted Kripke structures) relates models of $\mathrm{LTL}_{\mathrm{temp}}$ formulas in a Krikpe structure, and traces accepted by some NBA.

▶ **Proposition 19** ([2]). *Given an* $\mathrm{LTL}_{\mathrm{temp}}$ *formula* $\varphi$*, a Kripke structure* $\mathcal{K}$*, and a threshold* $v \in [0,1]$*, for an ordering relation* $\bowtie$ *in* $\{<,>\}$*, there exists an NBA* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\varphi}$ *such that for every trace* $t \in \mathbb{T}$*:*
1. *If* $[\![\varphi]\!](t) \bowtie v$*, then* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\varphi}$ *accepts* $t$*.*
2. *If* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\varphi}$ *accepts* $t$ *and* $t$ *is a lasso, then* $[\![\varphi]\!](t) \bowtie v$*.*

We build upon this result by extending it to formulas with quantifiers, and going from the classical semantics (in the first item) to the lasso semantics (in the second item). We use the same encodings of trace assignments as described in Section 3.3 for $\mathrm{HyperLTL}_{\mathrm{prop}}$.

▶ **Lemma 20.** *For an ordering relation* $\bowtie$ *in* $\{<,>\}$ *and* $\bowtie$ *its reflexive closure, for all formulas* $\psi$ *of* $\mathrm{HyperLTL}_{\mathrm{temp}}$*, for all Kripke structures* $\mathcal{K}$*, for all* $v \in [0,1]$*, we can build an NBA* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\psi}$ *such that for every trace assignment* $\Pi$ *over the free variables of* $\psi$*, it holds that:*
1. *if* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) \bowtie v$ *and all traces in* $\Pi$ *belong to* $\mathbb{T}(\mathcal{K})$*, then* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\psi}$ *accepts* $t_\Pi$*;*
2. *if* $\mathcal{A}^{\bowtie v}_{\mathcal{K},\psi}$ *accepts a lasso* $t_\Pi$*, then all traces in* $\Pi$ *belong to* $\mathbb{L}(\mathcal{K})$ *and* $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi) \bowtie v$*.*

If we only count the dependency in the quantifier part of $\mathrm{HyperLTL}_{\mathrm{temp}}$, the NBA $\mathcal{A}^{\bowtie v}_{\mathcal{K},\psi}$ has a size that is bounded by a tower of exponentials of height depending linearly on the quantifier alternation of $\psi$. Thanks to this technical result, we obtain as expected:

▶ **Corollary 21.** *For all formulas* $\psi$ *of* $\mathrm{HyperLTL}_{\mathrm{temp}}$*, Kripke structures* $\mathcal{K}$*, and lasso assignments* $\Pi$ *such that* $\Pi(\pi) \in \mathbb{L}(\mathcal{K})$ *for all free variables* $\pi$ *in* $\psi$*,* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) = [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi)$*.*

**Proof.** The proof is by induction on the quantifier depth of the formula $\psi$.

If $\psi$ is non quantified, the result is trivial since no quantifications remains to be evaluated.

If $\psi = \exists \pi\ \psi'$, by definition of the semantics of the existential quantifier, and then by induction hypothesis, we have $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) \geq [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi)$. Suppose that the inequality is strict, and let $v \in ([\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi), [\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi))$. Since $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) > v$, by Lemma 20.1, $t_\Pi$ is then accepted by $\mathcal{A}^{>v}_{\mathcal{K},\psi}$. Now, by Lemma 20.2, since $\Pi$ is a lasso assignment, $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi) \geq v$, which contradicts the hypothesis that $v > [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi)$.

If $\psi = \forall \pi\ \psi'$, by definition and by induction hypothesis, we have $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi) \leq [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi)$. Suppose that the inequality is strict, and let $v \in ([\![\psi]\!]_{\mathbb{T}(\mathcal{K})}(\Pi), [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi))$. By Lemma 20.1, $t_\Pi$ is then accepted by $\mathcal{A}^{<v}_{\mathcal{K},\psi}$. Now, by Lemma 20.2, as $\Pi$ is a lasso assignment, $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi) \leq v < [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}(\Pi)$. Contradiction. ◀

In Example 17, notice that although all lassos are associated by $\varphi$ with a positive semantics, we indeed get $[\![\forall \pi\ \varphi]\!]_{\mathbb{L}(\mathcal{K})} = 0$ by a convergence phenomenon. In contrast, we have also $[\![\forall \pi\ \varphi]\!]_{\mathbb{T}(\mathcal{K})} = 0$ as there exists (non-lasso) traces associated by $\varphi$ with a zero semantics.

**Issues relative to automata-based model checking.** In [2], the model checking problem amounts to deciding whether all traces of a Kripke structure $\mathcal{K}$ have a semantics wrt to an $\mathrm{LTL}_{\mathrm{temp}}$ formula $\varphi$ greater than or equal to a threshold $v$. This translates to our setting in deciding whether $[\![\forall \pi\ \varphi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$. They check this by testing emptiness of the NBA $\mathcal{A}^{<v}_{\mathcal{K},\varphi}$: this is correct because of item 2 of Proposition 19, since if $\mathcal{A}^{<v}_{\mathcal{K},\varphi}$ would accept a lasso $t$, it would be such that $[\![\varphi]\!](t) < v$.

In the setting of $\mathrm{HyperLTL}_{\mathrm{temp}}$, this leads to two difficulties. First, the second item of our Lemma 20 is weaker than the one of Proposition 19 as the comparison with the value $v$ is no longer strict, preventing the same argument as before in $\mathrm{LTL}_{\mathrm{temp}}$. Second, our formulas

contain $\forall$ and $\exists$ quantifiers (whose semantics are sup and inf operators) and thus, we should be able to solve the model checking problems $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$ and $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$, for a closed formula $\psi$. Once again, the reasoning above fails in this case since it is not true a priori that $[\![\forall \pi\ \varphi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$ if and only if the NBA $\mathcal{A}^{\geq v}_{\mathcal{K},\varphi}$ has an empty language. We demonstrate this more carefully in the example below.

▶ **Example 22.** We continue Example 17. Consider $\varphi_\pi = \mathbf{G}(\mathbf{F}_\eta\, p_\pi) \vee \mathbf{F}\,\mathbf{G}\,\neg p_\pi$, and $\psi = \forall \pi\ \varphi_\pi$. As we discussed, for a trivial Kripke structure $\mathcal{K}$ allowing every trace, we have $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} = [\![\psi]\!]_{\mathbb{L}(\mathcal{K})} = 0$: there exist witnesses $t \in \mathbb{T}(\mathcal{K})$ where $\varphi_\pi$ evaluates to 0, but no such witness is a lasso of $\mathcal{K}$. For this formula, we would like to solve the model checking problem $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq 0$ (which indeed implies that $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} = 0$). If we would use Lemma 20, this would require to build the NBA $\mathcal{A}^{>0}_{\mathcal{K},\psi}$ to check its emptiness. This automaton could be obtained by computing first the NBA $\mathcal{A}^{>0}_{\mathcal{K},\varphi_\pi}$ that accepts all lassos, complementing it (obtaining then an NBA with empty language) before taking the intersection with the traces generated by $\mathcal{K}$, then projecting away the trace $\pi$, and finally, complementing it again. The so-obtained NBA accepts the trace $t_{[]}$, and the model checking algorithm would thus wrongly declare that $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} > 0$.

We therefore cannot directly rely on the construction of an NBA to fully and exactly model check $\mathrm{HyperLTL}_{\mathrm{temp}}$. However, we propose two kinds of solutions in the rest of this section based on automata. First, we relax the problem, solving approximate model checking; second, we study several fragments for which the (exact) model checking task can be solved.

**Approximate model checking.** We weaken the model checking problem to only give an approximate solution instead:

▶ **Definition 23.** *The* approximate model checking *of* $\mathrm{HyperLTL}_{\mathrm{temp}}$ *(associated with the exact problem* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$*) is the following: given a Kripke structure* $\mathcal{K}$*, a closed* $\mathrm{HyperLTL}_{\mathrm{prop}}$ *formula* $\psi$*, a rational threshold* $v \in [0,1]$*, and a rational approximation factor* $\varepsilon > 0$*, it answers positively when* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v + \varepsilon$*, negatively when* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v - \varepsilon$*, and arbitrarily otherwise. We can consider alternatively the symmetric question associated with the exact question* $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$*.*

▶ **Theorem 24.** *The approximate model checking of* $\mathrm{HyperLTL}_{\mathrm{temp}}$ *is decidable with a non-elementary complexity.*

**Proof.** Let $\mathcal{K}$ be a Kripke structure, $\psi$ be a closed $\mathrm{HyperLTL}_{\mathrm{prop}}$ formula, $v \in [0,1]$ a rational threshold, and $\varepsilon > 0$ a rational approximation factor. Thanks to Corollary 21, $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} = [\![\psi]\!]_{\mathbb{L}(\mathcal{K})}$. Consider then the NBA $\mathcal{A}^{<v}_{\mathcal{K},\psi}$ of Lemma 20. Note that, since $\psi$ is closed, it runs on trace assignments over no traces: it either accepts the single infinite trace $t_{[]}$ that is a lasso (and thus, it is not empty), or it rejects it (and then, it is empty). The approximate model checking algorithm amounts to test emptiness for $\mathcal{A}^{<v}_{\mathcal{K},\psi}$. Indeed, if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v + \varepsilon$, i.e. $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \geq v + \varepsilon$, we have $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \leq v$ for no lassos, and thus (by property 2 of Lemma 20) the NBA $\mathcal{A}^{<v}_{\mathcal{K},\psi}$ cannot accept any lasso: the emptiness check of this NBA declares that its language is empty and the algorithm answers positively. Symmetrically, if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v - \varepsilon$, i.e. $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \leq v - \varepsilon$, we have $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \leq v$, and thus the single infinite lasso fulfills that; then, by property 2 of Lemma 20, it is accepted by the NBA $\mathcal{A}^{<v}_{\mathcal{K},\psi}$: the emptiness check of this NBA declares that its language is non empty. Hence, the algorithm answers negatively. ◀

We now show in the next two subsections the fragments of $\mathrm{HyperLTL}_{\mathrm{temp}}$ for which one can answer the (exact) model checking problem.

**Model checking of the positive and negative fragments.** We propose two first fragments for which we show the decidability of model checking. One is the fragment where negation is no longer allowed in $\text{HyperLTL}_{\text{temp}}$ formulas above $\mathbf{U}_\eta$ operator. For this reason, we call this restriction the *positive fragment*.

▶ **Definition 25.** *The fragment* $\text{HyperLTL}^+_{\text{temp}}$ *of* $\text{HyperLTL}_{\text{temp}}$ *is the subset of formulas* $\psi$ *described by the grammar (where* $p \in \text{AP}$*,* $\pi \in \mathcal{V}$*, and* $\eta \in \mathcal{D}$*):*

$$\psi ::= \exists \pi\, \psi \mid \forall \pi\, \psi \mid \varphi \qquad \varphi ::= \beta \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{U}\, \varphi \mid \varphi\, \mathbf{U}_\eta\, \varphi$$
$$\beta ::= p_\pi \mid \neg\beta \mid \beta \vee \beta \mid \mathbf{X}\, \beta \mid \beta\, \mathbf{U}\, \beta \qquad \text{(Boolean LTL formulas)}$$

The two model checking problems we solve for $\text{HyperLTL}^+_{\text{temp}}$ are the following ones:

- given a closed formula $\psi$ of $\text{HyperLTL}^+_{\text{temp}}$, a Kripke structure $\mathcal{K}$ and a threshold $v \in [0,1]$, decide if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$;
- given a closed formula $\psi$ of $\text{HyperLTL}^+_{\text{temp}}$, a Kripke structure $\mathcal{K}$ and a threshold $v \in (0,1]$, decide if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$. [5]

In a symmetric fashion, we can define the fragment $\text{HyperLTL}^-_{\text{temp}}$ obtained by considering negations of formulas of $\text{HyperLTL}^+_{\text{temp}}$. By pushing negations, this corresponds to the syntax

$$\psi ::= \exists \pi\, \psi \mid \forall \pi\, \psi \mid \varphi \qquad \varphi ::= \beta \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\, \varphi \mid \varphi\, \mathbf{R}\, \varphi \mid \varphi\, \mathbf{R}_\eta\, \varphi$$
$$\beta ::= p_\pi \mid \neg\beta \mid \beta \vee \beta \mid \mathbf{X}\, \beta \mid \beta\, \mathbf{U}\, \beta \qquad \text{(Boolean LTL formulas)}$$

where the semantics of the *release* operators $\mathbf{R}$ and $\mathbf{R}_\eta$ is obtained by a negation of the until operators, as usual. The model checking problems that we solve for this dual fragment are the following ones: given a closed formula $\psi$ of $\text{HyperLTL}^-_{\text{temp}}$, a Kripke structure $\mathcal{K}$ and a threshold $v \in [0,1]$ (resp. $v \in [0,1)$), decide if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$ (resp. $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$).

The formula $\varphi^{\text{temp}}_{\mathsf{OD}}$ of Example 15 is in the fragment $\text{HyperLTL}^-_{\text{temp}}$. Indeed, the negation of the formula can be rewritten as $\psi := \exists \pi\, \exists \pi'\ \ \text{same}_{\text{low}}(\pi, \pi') \wedge \mathbf{F}_\eta\, \neg \text{same}_{\text{low}}(\pi, \pi')$, where $\text{same}_{\text{low}}(\pi, \pi')$ is a Boolean LTL formula, and the operator $\mathbf{F}_\eta$ does not involve any negation. This formula belongs to $\text{HyperLTL}^+_{\text{temp}}$. To model check the formula $\varphi^{\text{temp}}_{\mathsf{OD}}$ against a threshold $\geq v$ (resp. $\leq v$), it suffices to model check the formula $\psi$ of $\text{HyperLTL}^+_{\text{temp}}$ against the threshold $\leq 1 - v$ (resp. $\geq 1 - v$). Similarly, the formula $\varphi^{\text{temp}}_{\mathsf{NI}}$ of Example 16 is in the fragment $\text{HyperLTL}^-_{\text{temp}}$.

The first ingredient for decidability of the model checking comes from Corollary 21. This allows us to only model check the *lasso semantics*, which can faithfully be done by using an automata-based construction. Another ingredient is the characterisation of the set of possible satisfaction values that can generate a positive formula, helping us controlling the convergence phenomena in infimums/supremums of the semantics. To make this ingredient more precise, we introduce, for all $k \in \mathbb{N}$, finite sets $H \subseteq \mathcal{D}$, and finite sets $\mathbb{W} \subset [0,1]$, the infinite set of values

$$V_{k,H,\mathbb{W}} = \{0,1\} \cup \mathbb{W} \cup \{w \times \textstyle\prod_{\ell=1}^{k'} \eta_{i_\ell}^{(\ell)} \mid w \in \mathbb{W}, k' \leq k, (i_\ell)_{1 \leq \ell \leq k'} \in \mathbf{N}^{k'}, (\eta^{(\ell)})_{1 \leq \ell \leq k'} \in H^{k'}\}$$

This set has the following properties, using as natural hypotheses that we only consider computable discounted sequences $\eta$:

[5] Notice that the threshold 0 is not allowed here, to avoid convergence issues.

▶ **Lemma 26.** *For all $k \in \mathbb{N}$, finite sets $H \subseteq \mathcal{D}$, and finite sets $\mathbb{W} \subset [0,1]$,*

- *for all $a \in (0,1)$, the set $V_{k,H,\mathbb{W}} \cap [a,1]$ is finite and computable;*
- *every non-decreasing sequence of $V_{k,H,\mathbb{W}}$ is stationary;*
- *every non-increasing sequence of $V_{k,H,\mathbb{W}}$ either converges to 0 or is stationary.*

**Proof.**

- Let $a \in (0,1)$. Since the functions $\eta \in H$ are decreasing and $H$ is finite, there is $i_0 \in \mathbf{N}$ such that for all $i \geq i_0$ and $\eta \in H$, $\eta(i) < a$. Thus, $V_{k,H,\mathbb{W}} \cap [a,1] \subseteq \{1\} \cup \mathbb{W} \cup \{w \times \prod_{\ell \leq k'} \eta_{i_\ell} \mid w \in \mathbb{W}, k' \leq k, \forall \ell \leq k \quad i_\ell \leq i_0, \eta \in H\}$ is finite and enumerable.
- Let $(u_n)$ be a non-decreasing sequence of $V_{k,H,\mathbb{W}}$. Thus, it has values in $V_{k,H,\mathbb{W}} \cap [u_0,1]$ that is finite, and thus is stationary.
- Let $(u_n)$ be a non-increasing sequence of $V_{k,H,\mathbb{W}}$. If the sequence does not converge to 0, it has a positive lower bound $a$, and thus has values in $[a,1]$. Since this set is finite, the sequence is stationary. ◀

Moreover, this set can indeed be used to obtain an over-approximation of the set of satisfaction values of a formula in $\mathrm{HyperLTL}^{+}_{\mathrm{temp}}$.

▶ **Lemma 27.** *Let $\psi$ be a formula of $\mathrm{HyperLTL}^{+}_{\mathrm{temp}}$. Let $k_\psi$ be the depth of $\mathbf{U}_\eta$ operators, and $H_\psi$ be the finite set of functions $\eta$ appearing in these operators. Let $\mathbb{W} \subset [0,1]$ be a finite set. Then, for all $T \subseteq \mathbb{T}_{\mathbb{W}}$, and $\Pi$ assignments to traces of $\mathbb{T}_{\mathbb{W}}$, $[\![\psi]\!]_T(\Pi) \in V_{k_\psi,H_\psi,\mathbb{W}}$.*

This is enough to obtain an automata-based algorithm to model check $\mathrm{HyperLTL}^{+}_{\mathrm{temp}}$ and $\mathrm{HyperLTL}^{-}_{\mathrm{temp}}$:

▶ **Theorem 28.** *The model checking problems for $\mathrm{HyperLTL}^{+}_{\mathrm{temp}}$ and $\mathrm{HyperLTL}^{-}_{\mathrm{temp}}$ are decidable with a non-elementary complexity.*

**Proof.** Let $\psi$ be a closed formula of $\mathrm{HyperLTL}^{+}_{\mathrm{temp}}$, $\mathcal{K}$ be a Kripke structure. Let $\mathbb{W}$ be the set of weights appearing in $\mathcal{K}$.

- Let $v \in [0,1]$. To decide whether $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$ (with $v \neq 0$ otherwise the answer is trivially true), we consider, by Lemma 27, the greatest value $v'$ of $V_{k_\psi,H_\psi,\mathbb{W}}$ smaller than $v$. This value can be computed as a search in the finite and enumerable set of possibles values $S = V_{k_\psi,H_\psi,\mathbb{W}} \cap [a,1]$ with $a = \sup_{i \mid \eta_i < v} \eta_i$ for any $\eta \in H_\psi$ (since the set $S$ contains $\eta_i$ and thus values smaller than $v$). We then let $v'' = (v' + v)/2$, and build the NBA $\mathcal{A}^{<v''}_{\mathcal{K},\psi}$ to check the emptiness of its language. If the language of $\mathcal{A}^{<v''}_{\mathcal{K},\psi}$ is empty, by Lemma 20.1, we cannot have $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} < v''$, and thus $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v''$ and then, $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$ as there are no values in $V_{k_\psi,H_\psi}$ in-between $v''$ and $v$. Reciprocally, if the language of $\mathcal{A}^{<v''}_{\mathcal{K},\psi}$ is non empty, it accepts the unique word over the alphabet with a single $\emptyset$ letter, which is thus a lasso. By Lemma 20.2, we have $[\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \leq v''$. By Corollary 21, $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v'' < v$.
- Let $v \in (0,1]$. To decide whether $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$ (with $v \neq 1$ otherwise the answer is trivially true), we consider, by Lemma 27, the lowest value $v'$ of $V_{k_\psi,H_\psi,\mathbb{W}}$ greater than $v$: it exists (and can be computed) since $v$ is taken different from 0. We then let $v'' = (v + v')/2$, and conclude the proof in this case symmetrically as in the previous case, showing that $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} = [\![\psi]\!]_{\mathbb{L}(\mathcal{K})} \geq v'' > v$.

We obtain the decidability for $\mathrm{HyperLTL}^{-}_{\mathrm{temp}}$ by a dual argument. ◀

The complexity of this model checking algorithm is at least as high as computing the automata given in Lemma 20, and thus at least non-elementary (with respect to the quantifier depth of the formulas): this is non-surprising since the model checking of HyperLTL has already a non-elementary lower bound in complexity. Formally, it also depends on the search

for the value $v'$ that is the greatest (resp. lowest) value smaller (resp. greater) than a value $v$ in a set of the form $V_{k,H,\mathbb{W}}$. This heavily depends on the sequences $\eta$ in $H$. If we suppose that these sequences are indeed geometrical, as it is the case for usual discounted automata where the current weight is always multiplied by a power $\lambda^i$ of a discounted parameter, this is easy, and thus does not change the complexity of the model checking algorithm.

**Model checking of the alternation-free fragments.** The other two fragments that we consider, orthogonal to the previous ones, are the *alternation-free* fragments: $\text{HyperLTL}^{\exists}_{\text{temp}}$ is the fragment containing formulas of the form $\exists \pi_1 \cdots \exists \pi_n \ \varphi$ with $\varphi$ quantifier-free, and $\text{HyperLTL}^{\forall}_{\text{temp}}$ is the fragment containing formulas of the form $\forall \pi_1 \cdots \forall \pi_n \ \varphi$.

The formula $\varphi^{\text{temp}}_{\mathsf{OD}}$ describing observational determinism is in $\text{HyperLTL}^{\forall}_{\text{temp}}$ (as it contains simply two universal quantifications), but not the formula $\varphi^{\text{temp}}_{\mathsf{NI}}$ describing non-inference (as it alternates between a universal quantification and an existential one).

The model checking problems we consider for these two logics are asymmetric: given a closed formula $\psi$ of $\text{HyperLTL}^{\exists}_{\text{temp}}$ (resp. $\text{HyperLTL}^{\forall}_{\text{temp}}$), a Kripke structure $\mathcal{K}$ and a threshold $v \in [0,1]$, decide if $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \leq v$ (resp. $[\![\psi]\!]_{\mathbb{T}(\mathcal{K})} \geq v$). These are tailored to mimick the model checking problem of $\text{LTL}_{\text{temp}}$ in [2], in order to correspond to the Boolean variants of classical model checking questions: for the existential fragment, the semantics of the existential quantification being a supremum over traces, asking whether the semantics of the formula is at most a certain threshold requires to decide if *all* traces have an associated value at most the threshold; for the universal fragment, the semantics of the universal quantification being a infimum over traces, asking whether the semantics of the formula is at least a certain threshold requires to decide if *all* traces have an associated value at least the threshold. By a proof building upon Proposition 19, and the techniques of Theorem 10, we obtain:

▶ **Theorem 29.** *The model checking of* $\text{HyperLTL}^{\exists}_{\text{temp}}$ *and* $\text{HyperLTL}^{\forall}_{\text{temp}}$ *is decidable with a non-elementary complexity..*

## 5 Conclusion

We have proposed a framework for qualitative reasoning about hyperproperties, by considering formulas whose semantics range in $[0,1]$. We introduce two logics extending HyperLTL, namely $\text{HyperLTL}_{\text{prop}}$ and $\text{HyperLTL}_{\text{temp}}$, and study their model checking problem. For the former, we propose an algorithm with a complexity similar to the one of HyperLTL. For the latter, we propose some model checking algorithms for fragments of this logic as well an approximate algorithm for the full logic. Notice that our approximation scheme is different from the one mentioned in [2] for $\text{LTL}_{\text{temp}}$. There, approximation is performed on the "discounted" sequences whose values are set to 0 when they become smaller than a given threshold $\delta$. Such approach could also be followed for $\text{HyperLTL}_{\text{temp}}$. We leave as open the decidability of the exact model checking problem for the whole logic $\text{HyperLTL}_{\text{temp}}$. As future works, we would like to study qualitative extensions of HyperPCTL [1], a temporal logic for reasoning about probabilistic hyperproperties of discrete-time Markov chains.

## References


1 Erika Ábrahám and Borzoo Bonakdarpour. HyperPCTL: A temporal logic for probabilistic hyperproperties. In *Quantitative Evaluation of Systems*. Springer, 2018. doi:10.1007/978-3-319-99154-2_2.

2 Shaull Almagor, Udi Boker, and Orna Kupferman. Formally reasoning about quality. *Journal of the ACM*, 63(3):1–56, 2016. doi:10.1145/2875421.

3 Benedikt Bollig, Paul Gastin, Benjamin Monmege, and Marc Zeitoun. Pebble weighted automata and weighted logics. *ACM Trans. Comput. Log.*, 15(2):15:1–15:35, 2014. doi:10.1145/2579819.

4 Marco Chiesa, Guy Kindler, and Michael Schapira. Traffic engineering with equal-cost-multipath: An algorithmic perspective. *IEEE/ACM Transactions on Networking*, 25(2):779–792, 2017. doi:10.1109/TNET.2016.2614247.

5 Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer Cham, 2020. doi:10.1007/978-3-319-10575-8.

6 M. R. Clarkson, Bernd Finkbeiner, M. Koleini, K. K. Micinski, Markus N. Rabe, and César Sánchez. Temporal logics for hyperproperties. In *POST 2014*, volume 8414 of *LNCS*, pages 265–284. Springer, 2014. doi:10.1007/978-3-642-54792-8_15.

7 M. R. Clarkson and F. Schneider. Hyperproperties. *Journal of Computer Security*, 18:1157–1210, 2010. doi:10.1109/CSF.2008.7.

8 Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007. doi:10.1016/J.TCS.2007.02.055.

9 Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. From Spot 2.0 to Spot 2.10: What's new? In *CAV 2022*, volume 13372 of *LNCS*, pages 174–187. Springer, 2022. doi:10.1007/978-3-031-13188-2_9.

10 Marco Faella, Axel Legay, and Mariëlle Stoelinga. Model checking quantitative linear time logic. *Electronic Notes in Theoretical Computer Science*, 220:61–77, 2008. doi:10.1016/j.entcs.2008.11.019.

11 Angelo Ferrando, Giulia Luongo, Vadim Malvone, and Aniello Murano. Theory and practice of quantitative ATL. In *PRIMA 2024: Principles and Practice of Multi-Agent Systems*, volume 15395 of *LNCS*. Springer, 2024. doi:10.1007/978-3-031-77367-9_18.

12 Bernd Finkbeiner and Christopher Hahn. Deciding hyperproperties. In *CONCUR 2016*, volume 59 of *LIPIcs*, pages 13:1–13:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.13.

13 Bernd Finkbeiner, Christopher Hahn, and Hazem Torfah. Model checking quantitative hyperproperties. In *CAV 2018*, volume 10981 of *LNCS*. Springer, 2018. doi:10.1007/978-3-319-96145-3_8.

14 Bernd Finkbeiner and Martin Zimmermann. The first-order logic of hyperproperties. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.30.

15 R. Focardi and R. Gorrieri. Classification of security properties (part I: Information flow). In *FOSAD 2000*, volume 2171 of *LNCS*, pages 331–396. Springer, 2000. doi:10.1007/3-540-45608-2_6.

16 Rob Gerth, Doron Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *PSTV 1995*, pages 3–15. Springer, 1995. doi:10.1007/978-0-387-34892-6_1.

17 Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20. IEEE, 1982. doi:10.1109/SP.1982.10014.

18 Samuel Graepler, Benjamin Monmege, and Jean-Marc Talbot. Reasoning about quality in hyperproperties. Technical Report 2512.00500, arXiv, 2025. doi:10.48550/arXiv.2512.00500.

19 Gerard J. Holzmann. The model checker SPIN. *IEEE Trans. Software Eng.*, 23(5):279–295, 1997. doi:10.1109/32.588521.

20 John McLean. A general theory of composition for trace sets closed under selective interleaving functions. In *Symposium on Research in Security and Privacy*, pages 79–93. IEEE Computer Society, 1994. doi:10.1109/RISP.1994.296590.

21 C. E. Shannon. The synthesis of two terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

22 A. P. Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logic. *Journal of the ACM*, 32:733–749, 1985. doi:10.1145/3828.3837.

23 G. Smith. On the foundations of quantitative information flow. In *FoSSaCS 2009*, volume 5504 of *LNCS*. Springer, 2009. doi:10.1007/978-3-642-00596-1_21.

24 Tachio Terauchi and Alex Aiken. Secure information flow as a safety problem. In Chris Hankin and Igor Siveroni, editors, *Static Analysis*, pages 352–367, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11547662_24.

25 Hirotoshi Yasuoka and Tachio Terauchi. Quantitative information flow as safety and liveness hyperproperties. *Theoretical Computer Science*, 538:167–182, 2014. doi:10.1016/J.TCS.2013.07.031.

26 A. Zakinthinos and E.S. Lee. A general theory of security properties. In *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*, pages 94–102, 1997. doi:10.1109/SECPRI.1997.601322.

27 Steve Zdancewic and Andrew C. Myers. Observational determinism for concurrent program security. In *CSFW 2003*. IEEE Computer Society, 2003. doi:10.1109/CSFW.2003.1212703.