The Keys to Decidable HyperLTL Satisfiability: Small Models or Very Simple Formulas

Corto Mascle
ENS Paris-Saclay, Cachan, France
corto.mascle@ens-paris-saclay.fr

Martin Zimmermann
University of Liverpool, Liverpool, United Kingdom
martin.zimmermann@liverpool.ac.uk

Abstract

HyperLTL, the extension of Linear Temporal Logic by trace quantifiers, is a uniform framework for expressing information flow policies by relating multiple traces of a security-critical system. HyperLTL has been successfully applied to express fundamental security policies like noninterference and observational determinism, but has also found applications beyond security, e.g., distributed protocols and coding theory. However, HyperLTL satisfiability is undecidable as soon as there are existential quantifiers in the scope of a universal one. To overcome this severe limitation to applicability, we investigate here restricted variants of the satisfiability problem to pinpoint the decidability border.

First, we restrict the space of admissible models and show decidability when restricting the search space to models of bounded size or to finitely representable ones. Second, we consider formulas with restricted nesting of temporal operators and show that nesting depth one yields decidability for a slightly larger class of quantifier prefixes. We provide tight complexity bounds in almost all cases.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Hyperproperties, Linear Temporal Logic, Satisfiability

Digital Object Identifier 10.4230/LIPIcs.CSL.2020.29

Related Version Full version containing all proofs omitted due to space restrictions available at `https://arxiv.org/abs/1907.05070`.

Funding Partially funded by EPSRC grant EP/S032207/1.

1 Introduction

The introduction of temporal logics for the specification of information flow policies [3] was a significant milestone in the long and successful history of applying logics in computer science [16]. Probably the most important representative of these logics is HyperLTL [3], which extends Linear Temporal Logic (LTL) [23] by trace quantifiers. This addition allows to express properties that relate multiple execution traces, which is typically necessary to capture the flow of information [4]. In contrast, LTL, currently the most influential specification language for reactive systems, is only able to express properties of single traces.

HyperLTL provides a uniform framework for expressing information flow policies in a formalism with intuitive syntax and semantics, and for the automated verification of these policies: A wide range of policies from the literature [15, 19, 20, 21, 22, 27] with specialized verification algorithms is expressible in HyperLTL, i.e., universal HyperLTL verification algorithms are applicable to all of them.

As an example, consider a system with a set I of inputs, which contains a hidden input $h \in I$, and an output o. Now, noninterference [15] between h and o requires that no information about h is leaked via o, i.e., for all execution traces π and π', if the inputs in

28th EACSL Annual Conference on Computer Science Logic (CSL 2020).
Editors: Maribel Fernández and Anca Muscholl; Article No. 29; pp. 29:1–29:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

π and π' only differ in h, then they have the same output at all times. Formally, this is captured by the HyperLTL formula

$$\forall\pi.\forall\pi'.\left(\mathbf{G}\bigwedge\nolimits_{i\in I\setminus\{h\}}(i_\pi \Leftrightarrow i_{\pi'})\right) \Rightarrow \mathbf{G}\,(o_\pi \Leftrightarrow o_{\pi'}).$$

As another example, consider a system with a public output o and a secret output s (we consider only one of each for simplicity). One may want to express that the behaviour of the secret output cannot be inferred from the behaviour of the public one. HyperLTL can express the property that for all executions of the system, there exists another execution with the same behaviour of o but a different behaviour of s, using the formula

$$\forall\pi.\exists\pi'.\mathbf{G}\,(o_\pi \Leftrightarrow o_{\pi'}) \wedge \mathbf{F}\,\neg(s_\pi \Leftrightarrow s_{\pi'}).$$

Today, there are tools for model checking HyperLTL properties [6, 13], for checking satisfiability of HyperLTL properties [9, 11], for synthesizing reactive systems from HyperLTL properties [10], and for runtime monitoring of HyperLTL properties [1, 2, 12]. Furthermore, the extraordinary expressiveness of HyperLTL has been exhibited [14] and connections to first and second-order predicate logics have been established [5, 14].

The major drawback of HyperLTL is the usual price one has to pay for great expressiveness: prohibitively high worst-case complexity. In particular, model checking finite Kripke structures against HyperLTL formulas is nonelementary [3] and satisfiability is even undecidable [8]. These results have to be contrasted with model checking and satisfiability being PSpace-complete for LTL [25], problems routinely solved in real-life applications [18].

Due to the sobering state of affairs, it is imperative to find fragments of the logic with (more) tractable complexity. In this work, we focus on the satisfiability problem, the most fundamental decision problem for a logic. Nevertheless, it has many applications in verification, e.g., checking the equivalence and implication of specifications can be reduced to satisfiability. Finally, the question whether a property given by some HyperLTL formula is realizable by some system is also a satisfiability problem.

A classical attempt to overcome the undecidability of the satisfiability problem is to restrict the number of quantifier alternations of the formulas under consideration. In fact, the alternation depth is the measure underlying the nonelementary complexity of the HyperLTL model checking problem [3]. However, the situation is different for the satisfiability problem: It is undecidable even when restricted to $\forall\exists$ formulas, i.e., formulas starting with one universal quantifier followed by a single existential one [8]. All remaining prefix classes are decidable by reductions to the LTL satisfiability problem, e.g., the satisfiability problem is PSpace-complete for the alternation-free prefix classes $\exists^*$ and $\forall^*$ and ExpSpace-complete for the class $\exists^*\forall^*$ [8].

However, there are more complexity measures beyond the alternation depth that can be restricted in order to obtain tractable satisfiability problems, both on formulas and on models. The latter case is of particular interest, since it is known that not every satisfiable HyperLTL has a “simple” model, for various formalizations of “simple” [14]. Thus, for those formulas, such a restriction could make a significant difference. Furthermore, from a more practical point of view, one is often interested in whether there is a, say, finite model while the existence of an intricate infinite model may not be useful.

We study the satisfiability problem for formulas with restricted quantifier prefixes and restricted temporal depth [7], which measures the nesting of temporal operators. Our main result here shows that satisfiability is even undecidable for formulas of the form $\forall^2\exists^*\varphi$, where φ has temporal depth one and only uses eventually $\mathbf{F}$ and always $\mathbf{G}$, i.e., it is a

Boolean combination of formulas $\mathbf{F}\,\varphi'$ with propositional φ'. Thereby, we strengthen the previous undecidability result for $\forall\exists$ by bounding the temporal depth to one, but at the price of a second universal quantifier. Moreover, we clarify the border between decidability and undecidability at temporal depth two: Using only one universally quantified variable, temporal depth one, and only $\mathbf{F}$, $\mathbf{G}$, and nested applications of next $\mathbf{X}$ leads to decidability. Finally, we show that every HyperLTL formula can be transformed into an equisatisfiable $\forall^2\exists^*$ formula of temporal depth two, i.e., this fragment already captures the full complexity of the satisfiability problem.

Thus, the overall picture is still rather bleak: if one only restricts the formula then the islands of decidability are very small. Phrased differently, even very simple formulas are extremely expressive and allow to encode computations of Turing-complete devices in their models. However, note that such models are necessarily complex, as they need to be able to encode an unbounded amount of information.

Thus, we also consider satisfiability problems for arbitrary formulas, but with respect to restricted models which do not allow to encode such computations. In particular, we consider three variants of increasing complexity: Checking whether a given HyperLTL formula has a model of a given cardinality k is ExpSpace-complete, whether it has a model containing only ultimately periodic traces of length at most k is N2ExpTime-complete, and checking whether it has a model induced by a Kripke structure with k states is Tower-complete. The last result is even true for a fixed Kripke structure, which therefore has implications for the complexity of the model checking problem as well. Thus, the situation is more encouraging when checking for the existence of small models: satisfiability becomes decidable, even with (relatively) moderate complexity in the first two cases.

However, as argued above, all three approaches are (necessarily) incomplete: There are satisfiable formulas that have only infinite models, satisfiable formulas that have only non-ultimately periodic models, and satisfiable formulas that have no ω-regular models [14], a class of models that includes all those that are induced by a finite Kripke structure.

All in all, our work shows that HyperLTL satisfiability remains a challenging problem, but we have provided a complete classification of the tractable cases in terms of alternation depth, temporal depth, and representation of the model (for formulas without until).

2 Definitions

Fix a finite set AP of atomic propositions. A *valuation* is a subset of AP. A *trace* over AP is a map $t\colon \mathbb{N} \to 2^{\mathrm{AP}}$, denoted by $t(0)t(1)t(2)\cdots$, i.e., an infinite sequence of valuations. The set of all traces over AP is denoted by $(2^{\mathrm{AP}})^\omega$. The *projection* of t to AP' is the trace $(t(0)\cap \mathrm{AP}')(t(1)\cap \mathrm{AP}')(t(2)\cap \mathrm{AP}')\cdots$ over AP'. A trace t is *ultimately periodic*, if $t = x \cdot y^\omega$ for some $x, y \in (2^{\mathrm{AP}})^+$, i.e., there are $s, p > 0$ with $t(n) = t(n+p)$ for all $n \geq s$.

The formulas of HyperLTL are given by the grammar

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

where a ranges over atomic propositions in AP and where π ranges over a fixed countable set $\mathcal{V}$ of *trace variables*. Conjunction, implication, equivalence, and exclusive disjunction $\oplus$, as well as the temporal operators eventually $\mathbf{F}$ and always $\mathbf{G}$ are derived as usual. A *sentence* is a closed formula, i.e., a formula without free trace variables. The *size* of a formula φ, denoted by $|\varphi|$, is its number of distinct subformulas.

The semantics of HyperLTL is defined with respect to a *trace assignment*, a partial mapping $\Pi\colon \mathcal{V} \to (2^{\mathrm{AP}})^\omega$. The assignment with empty domain is denoted by $\Pi_\emptyset$. Given a trace assignment Π, a trace variable π, and a trace t we denote by $\Pi[\pi \to t]$ the assignment that coincides with Π everywhere but at π, which is mapped to t. We also use shorthand notation like $[\pi_1 \to t_1, \ldots, \pi_n \to t_n]$ and $[(\pi_i \to t_i)_{1\leq i\leq n}]$ for $\Pi_\emptyset[\pi_1 \to t_1]\ldots[\pi_n \to t_n]$, if the π_i are pairwise different. Furthermore, $\Pi[j,\infty)$ denotes the trace assignment mapping every π in Π's domain to $\Pi(\pi)(j)\Pi(\pi)(j+1)\Pi(\pi)(j+2)\cdots$.

For sets T of traces and trace assignments Π we define

- $(T,\Pi) \models a_\pi$, if $a \in \Pi(\pi)(0)$,
- $(T,\Pi) \models \neg\psi$, if $(T,\Pi) \not\models \psi$,
- $(T,\Pi) \models \psi_1 \vee \psi_2$, if $(T,\Pi) \models \psi_1$ or $(T,\Pi) \models \psi_2$,
- $(T,\Pi) \models \mathbf{X}\,\psi$, if $(T,\Pi[1,\infty)) \models \psi$,
- $(T,\Pi) \models \psi_1 \,\mathbf{U}\, \psi_2$, if there is a $j \geq 0$ such that $(T,\Pi[j,\infty)) \models \psi_2$ and for all $0 \leq j' < j$: $(T,\Pi[j',\infty)) \models \psi_1$,
- $(T,\Pi) \models \exists\pi.\varphi$, if there is a trace $t \in T$ such that $(T,\Pi[\pi \to t]) \models \varphi$, and
- $(T,\Pi) \models \forall\pi.\varphi$, if for all traces $t \in T$: $(T,\Pi[\pi \to t]) \models \varphi$.

We say that T *satisfies* a sentence φ if $(T,\Pi_\emptyset) \models \varphi$. In this case, we write $T \models \varphi$ and say that T is a *model* of φ. Conversely, satisfaction of quantifier-free formulas does not depend on T. Hence, we say that Π *satisfies* a quantifier-free ψ if $(\emptyset,\Pi) \models \psi$ and write $\Pi \models \psi$ (assuming Π is defined on all trace variables that appear in ψ).

The *alternation depth* of a HyperLTL sentence φ, denoted by $\mathrm{ad}(\varphi)$, is defined as its number of quantifier alternations. Its *temporal depth*, denoted by $\mathrm{td}(\varphi)$, is defined as the maximal depth of the nesting of temporal operators in the sentence. Formally, td and ad are defined as follows:

- $\mathrm{td}(a_\pi) = 0$
- $\mathrm{td}(\neg\psi) = \mathrm{td}(\psi)$
- $\mathrm{td}(\psi_1 \vee \psi_2) = \max(\mathrm{td}(\psi_1), \mathrm{td}(\psi_2))$
- $\mathrm{td}(\mathbf{X}\,\psi) = 1 + \mathrm{td}(\psi)$
- $\mathrm{td}(\psi_1 \,\mathbf{U}\, \psi_2) = 1 + \max(\mathrm{td}(\psi_1), \mathrm{td}(\psi_2))$
- $\mathrm{td}(\exists\pi.\varphi) = \mathrm{td}(\varphi)$
- $\mathrm{td}(\forall\pi.\varphi) = \mathrm{td}(\varphi)$

- $\mathrm{ad}(\exists\pi.\psi) = 0$ for quantifier-free ψ
- $\mathrm{ad}(\forall\pi.\psi) = 0$ for quantifier-free ψ
- $\mathrm{ad}(\exists\pi.\exists\pi'.\varphi) = \mathrm{ad}(\exists\pi'.\varphi)$
- $\mathrm{ad}(\forall\pi.\forall\pi'.\varphi) = \mathrm{ad}(\forall\pi'.\varphi)$
- $\mathrm{ad}(\exists\pi.\forall\pi'.\varphi) = 1 + \mathrm{ad}(\forall\pi'.\varphi)$
- $\mathrm{ad}(\forall\pi.\exists\pi'.\varphi) = 1 + \mathrm{ad}(\exists\pi'.\varphi)$

Although HyperLTL sentences are required to be in prenex normal form, they are closed under Boolean combinations, which can easily be seen by transforming such formulas into prenex normal form. Note that this transformation can be implemented such that it changes neither the temporal nor alternation depth, and can be performed in polynomial time.

The fragment $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G})$ contains formulas of temporal depth one using only $\mathbf{F}$ and $\mathbf{G}$ as temporal operators, and $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G},\mathbf{X}^*)$ contains formulas using only $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{X}$ as temporal operators and of temporal depth one, however we allow iterations of the $\mathbf{X}$ operator. Formally, $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G},\mathbf{X}^*)$ formulas are generated by the grammar

$$\begin{aligned}\varphi &::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi\\ \psi &::= \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X}^n\psi' \mid \mathbf{F}\,\psi' \mid \mathbf{G}\,\psi' \mid \psi'\\ \psi' &::= a_\pi \mid \neg\psi' \mid \psi' \vee \psi' \mid \psi' \wedge \psi'\end{aligned}$$

where n ranges over the natural numbers. The grammar for $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G})$ is obtained by removing $\mathbf{X}^n\psi'$ from the grammar above.

Also, we use standard notation for classes of formulas with restricted quantifier prefixes, e.g., $\forall^2\exists^*$ denotes the set of HyperLTL formulas in prenex normal form with two universal quantifiers followed by an arbitrary number of existential quantifiers, but no other quantifiers.

Finally, we encounter various complexity classes, classical ones from NP to N2ExpTime, as well as Tower (see, e.g., [24]). Intuitively, Tower is the set of problems that can be solved by a Turing machine that, on an input of size n, stops in time $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$, with the height of the tower of exponents bounded by $b(n)$, where b is a fixed elementary function. The reductions presented in this work are polynomial time reductions unless otherwise stated.

3 Satisfiability for Restricted Classes of Models

The satisfiability problem "Given a HyperLTL sentence φ, does φ have a nonempty model?" is undecidable, even when restricted to finite models [8]. Hence, one has to consider simpler problems to regain decidability. In this section, we simplify the problem by checking only for the existence of *simple* models, for the following three formalizations of simplicity, where the bound k is always part of the input:

- Models of cardinality at most k (Theorem 1).
- Models containing only ultimately periodic traces xy^ω with $|x| + |y| \leq k$ (Theorem 2).
- Models induced by finite-state systems with at most k states (Theorem 3).

In every case, we allow arbitrary HyperLTL formulas as input and encode k in binary.

With the following result, we determine the complexity of checking satisfiability with respect to models of bounded cardinality. The algorithm uses a technique introduced by Finkbeiner and Hahn [8, Theorem 3] that allows us to replace existential and universal quantification by disjunctions and conjunctions, if the model is finite. Similarly, the lower bound also follows from Finkbeiner and Hahn.

▶ **Theorem 1.** *The following problem is* ExpSpace*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model with at most k traces?*

Proof. For the ExpSpace upper bound, one can check, given φ and k, satisfiability of the sentence $\exists\pi_1 \ldots \exists\pi_k.\overline{\varphi}$ where $\overline{\varphi}$ is defined inductively as follows:

- $\overline{\varphi} = \varphi$ if φ is quantifier-free.
- $\overline{\forall\pi.\varphi} = \bigwedge_{i=1}^{k} \overline{\varphi}[\pi \leftarrow \pi_i]$.
- $\overline{\exists\pi.\varphi} = \bigvee_{i=1}^{k} \overline{\varphi}[\pi \leftarrow \pi_i]$.

Here, $\overline{\varphi}[\pi \leftarrow \pi_i]$ is obtained from $\overline{\varphi}$ by replacing every occurrence of π by π_i. This sentence states the existence of at most k traces satisfying φ by replacing every quantifier by an explicit conjunction or disjunction over the possible assignments.

The resulting sentence is of size at most $|\varphi|k^{|\varphi|} + k$, which is exponential in the size of the input and its satisfiability can be checked in polynomial space in the size of the resulting formula [8]. As a result, the problem is in ExpSpace as well.

Finkbeiner and Hahn showed that satisfiability is ExpSpace-complete for sentences of the form $\exists^*\forall^*$ [8]. This implies ExpSpace-hardness of our problem, as if such a sentence, say with k existential quantifiers, is satisfiable then it has a model with at most k traces. ◀

As the algorithm proceeds by a reduction to the satisfiability problem for $\exists^*$ formulas, which in turn is reduced to LTL satisfiability, one can show that a HyperLTL sentence φ has a model with k traces if and only if it has a model with k ultimately periodic traces.

Next, we consider another variant of the satisfiability problem, where we directly restrict the space of possible models to ultimately periodic ones of the form xy^ω with $|x| + |y| \leq k$.

As we encode k in binary, the length of those traces is exponential in the input and the cardinality of the model is bounded doubly-exponentially. This explains the increase in complexity in the following theorem in comparison to Theorem 1.

▶ **Theorem 2.** *The following problem is* N2ExpTime*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model whose elements are of the form xy^ω with $|x| + |y| \leq k$?*

As expected, the complexity of the satisfiability problem increases the more traces one has at hand to encode computations. In Theorem 1, we have exponentially many; in Theorem 2, we have doubly-exponentially many. In our last theorem, we consider infinite sets of traces that are finitely representable by finite-state systems. Here, satisfiability becomes intractable, yet still decidable, even when restricted to formulas of temporal depth one.

Formally, a *Kripke structure* $\mathcal{K} = (Q, \delta, Q_0, \lambda)$ consists of a finite set Q of states, a set $Q_0 \subseteq Q$ of initial states, a transition function $\delta\colon Q \to 2^Q \setminus \{\emptyset\}$, and a labelling function $\lambda\colon Q \to 2^{\mathrm{AP}}$. A *run* of $\mathcal{K}$ is an infinite sequence $q_0q_1q_2\cdots$ of states starting with $q_0 \in Q_0$ and such that $q_{j+1} \in \delta(q_j)$ for all $j \in \mathbb{N}$. A trace of $\mathcal{K}$ is the sequence of labels $\lambda(q_0)\lambda(q_1)\lambda(q_2)\cdots$ associated to a run $q_0q_1q_2\cdots$ of $\mathcal{K}$. The set of traces of $\mathcal{K}$ is denoted by $\mathrm{T}(\mathcal{K})$.

▶ **Theorem 3.** *The following problem is* Tower*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model $T(\mathcal{K})$ for some Kripke structure $\mathcal{K}$ with at most k states?*

Proof. Clarkson et al. presented a model-checking algorithm for HyperCTL* (and thus for HyperLTL, which is a fragment of HyperCTL*), and showed that its complexity is a tower of exponentials whose height is the alternation depth of the input sentence [3]. Thus, one can enumerate all Kripke structures with at most k states (up to isomorphism) and model-check them one by one in Tower. This yields the desired upper bound, as there are “only” exponentially many (in k) Kripke structures with k states.

The lower bound is obtained by a reduction from the universality problem for star-free regular expressions with complementation. The equivalence problem for those expressions is Tower-complete (under elementary reductions, which is standard for Tower-complete problems), even for two-letter alphabets [24, 26]. As those expressions are closed by complementation and union, the universality problem is Tower-complete as well.

Star-free expressions with complementation over $\{a, b\}$ are generated by the grammar

$$e ::= a \mid b \mid \varepsilon \mid \emptyset \mid e + e \mid ee \mid \neg e$$

and have the obvious semantics inducing a language over $\{a, b\}^*$, denoted by e as well.

Let e be such an expression. We construct a HyperLTL sentence φ_e and a Kripke structure $\mathcal{K}$ such that $T(\mathcal{K})$ is a model of φ_e if and only if e is universal. $\mathcal{K}$ does not depend on e and is shown in Figure 1. As all sets of variables in $\mathcal{K}$ are singletons, we indifferently use the notation a for the letter a and the singleton $\{a\}$. The set of traces induced by this Kripke structure is

$$T(\mathcal{K}) = l^\omega + l^*(a+b)^\omega + l^*(a+b)^*r^\omega + l^*\#r^\omega.$$

Given an expression e and a trace variable π, we inductively define a formula $\psi_{e,\pi}$ which expresses that when π is mapped by a trace assignment Π to a trace of $\mathcal{K}$ of the form l^nwr^ω with $w \in \{a, b\}^*$, then $w \in e$ if and only if $(T(\mathcal{K}), \Pi) \models \psi_{e,\pi}$.

- $\psi_{\emptyset,\pi} = a_\pi \wedge \neg a_\pi$: No trace assignment satisfies $\psi_{\emptyset,\pi}$, just as the language of $\emptyset$ does not contain any word.

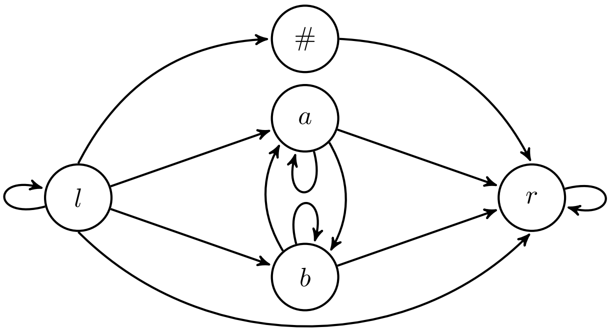

Figure 1 The Kripke structure $\mathcal{K}$ (all states are initial).

- $\psi_{\varepsilon,\pi} = \mathbf{G}\,(l_\pi \vee r_\pi)$: $(T(\mathcal{K}), \Pi)$ with $\Pi(\pi) = l^n w r^\omega$ satisfies $\psi_{\varepsilon,\pi}$ if and only if $w = \varepsilon$.
- $\psi_{a,\pi} = \exists \tau.(\mathbf{F}\,\#_\tau) \wedge \mathbf{F}\,(a_\pi) \wedge \mathbf{G}\,(l_\tau \Leftrightarrow l_\pi \wedge r_\tau \Leftrightarrow r_\pi)$: The traces of $\mathcal{K}$ with an occurrence of # are the traces of the form $l^*\#r^\omega$. Thus, $(T(\mathcal{K}), \Pi)$ with $\Pi(\pi) = l^n w r^\omega$ satisfies $\psi_{a,\pi}$ if and only if $l^n w r^\omega$ is a copy of such a trace with # replaced by a, i.e., if and only if $w = a$.
- $\psi_{b,\pi} = \exists \tau.(\mathbf{F}\,\#_\tau) \wedge \mathbf{F}\,(b_\pi) \wedge \mathbf{G}\,(l_\tau \Leftrightarrow l_\pi \wedge r_\tau \Leftrightarrow r_\pi)$: Similarly to $\psi_{a,\pi}$.
- $\psi_{e_1+e_2,\pi} = \psi_{e_1,\pi} \vee \psi_{e_2,\pi}$.
- $\psi_{e_1e_2,\pi} = \exists \pi_1.\exists \pi_2.\psi \wedge \psi'$ with

  $$\psi = \mathbf{F}\,r_{\pi_1} \wedge \mathbf{F}\,r_{\pi_2} \wedge \mathbf{G}\,(\neg\#_{\pi_1} \wedge \neg\#_{\pi_2}) \wedge \psi_{e_1,\pi_1} \wedge \psi_{e_2,\pi_2}$$

  expressing that π_1 and π_2 are of the form $l^{n_1} w_1 r^\omega$ and $l^{n_2} w_2 r^\omega$ with $w_1 \in e_1$ and $w_2 \in e_2$, and with

  $$\psi' = \mathbf{G}\,(l_{\pi_2} \Leftrightarrow \neg r_{\pi_1}) \wedge \mathbf{G}\,(a_\pi \Leftrightarrow (a_{\pi_1} \vee a_{\pi_2}) \wedge b_\pi \Leftrightarrow (b_{\pi_1} \vee b_{\pi_2}))$$

  expressing that $n_2 = n_1 + |w_1|$ and that $w = w_1 w_2$, where $\Pi(\pi) = l^n w r^\omega$. Thus, $(T(\mathcal{K}), \Pi)$ satisfies $\psi_{e_1e_2,\pi}$ if and only if there exist $w_1 \in e_1, w_2 \in e_2$ such that $w = w_1 w_2$.
- $\psi_{\neg e,\pi} = \neg\psi_{e,\pi}$.

Although this inductive definition does not necessarily give a formula in prenex normal form, one can easily check that no quantifier is in the scope of a temporal operator, thus the resulting formula can be turned into a HyperLTL formula.

To conclude, consider the sentence $\varphi_e = \forall \pi.\mathbf{G}\,\neg r_\pi \vee \mathbf{F}\,\#_\pi \vee \psi_{e,\pi}$, which can again be brought into prenex normal form. Further, note that no temporal operator is in the scope of another one, thus φ_e has temporal depth one. The set $T(\mathcal{K})$ is a model of φ_e if and only if all its traces are in $\{a, b, l\}^\omega$, in $l^*\#r^\omega$, or of the form $l^* w r^\omega$ with $w \in e$. This is the case if and only if all words $w \in \{a, b\}^*$ are in the language of e, i.e., if and only if e is universal. ◀

As the Kripke structure $\mathcal{K}$ in the lower bound proof above is fixed, we also obtain a novel hardness result for model-checking.

▶ **Corollary 4.** *HyperLTL model-checking a fixed Kripke structure with five states is* Tower-*complete, even for sentences of temporal depth one.*

Note that one could already infer the Tower-completeness of the model-checking problem by carefully examining the proof of Theorem 5 of [3] concerning HyperCTL* model-checking. The reduction from the satisfiability problem for QPTL presented there also works for HyperLTL, albeit with temporal depth larger than one.

Table 1 Complexity of HyperLTL satisfiability in terms of quantifier prefixes and temporal depth. An asterisk * denotes that the upper bound only holds for until-free formulas. All lower bounds in the second column already hold for temporal depth two.

	temporal depth one	arbitrary temporal depth
$\exists^*$ / $\forall^*$	NP-complete ([7]+[8])	PSPACE-complete ([8]+[25])
$\exists^*\forall^*$	NEXPTIME-complete (Thm. 12)	EXPSPACE-complete ([8])
$\exists^*\forall\exists^*$	in N2EXPTIME* (Thm. 11)	undecidable ([8])
$\forall^2\exists^*$	undecidable (Thm. 9)	undecidable

4 Satisfiability for Restricted Classes of Formulas

After studying the HyperLTL satisfiability problem for classes of restricted models, but arbitrary formulas, we now consider restrictions on formulas, but arbitrary models. Recall that Finkbeiner and Hahn presented a complete picture in terms of quantifier prefixes: Satisfiability is PSPACE-complete for the alternation-free fragments $\exists^*$ and $\forall^*$ as well as EXPSPACE-complete for $\exists^*\forall^*$. In all other cases, the problem is undecidable, i.e., as soon as there is a universal quantifier in front of an existential one.

In a sense, the decidable fragments are variants of LTL: Both alternation-free fragments can easily be reduced to LTL satisfiability while the $\exists^*\forall^*$ one is easily reducible to the $\exists^*$ fragment, with an exponential blowup. Thus, the decidable fragments barely exceed the realm of LTL.

In this section, we consider another dimension to measure the complexity of formulas, temporal depth, i.e., we restrict the nesting of temporal operators. The hope is that in this setting, we can obtain decidability for larger quantifier prefix classes. However, a slight adaptation of Finkbeiner and Hahn's undecidability result for $\forall\exists$, along with an application of Lemma 6 proven below, already shows undecidability for $\forall\exists$ formulas of temporal depth two and without untils.

Thus, we have to restrict our search to fragments of temporal depth one, which contain most of the information flow policies expressible in HyperLTL [3]. And indeed, we prove satisfiability decidable for $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ formulas. Thus, if the temporal depth is one and untils are excluded, then one can allow a universal quantifier in front of existential ones without losing decidability. This fragment includes, for example, the noninference property [21], as well as the second example presented in the introduction.

However, even allowing the smallest possible extension, i.e., adding a second universal quantifier, leads again to undecidability: HyperLTL satisfiability is undecidable for $\forall^2\exists^*$ formulas of temporal depth one using only $\mathbf{F}$ as temporal operator. Thus, satisfiability remains hard, even when severely restricting the temporal depth of formulas. Our results for temporal depth one are summarized in Table 1.

We begin this section by showing that every HyperLTL formula can be transformed in polynomial time into an equisatisfiable one with quantifier prefix $\forall^2\exists^*$ with temporal depth two. Thus, this fragment already captures the full complexity of the satisfiability problem. This transformation is later used in several proofs.

▶ **Theorem 5.** *For every HyperLTL sentence one can compute in polynomial time an equisatisfiable sentence of the form $\forall^2\exists^*$ with temporal depth at most two.*

We decompose the proof into three steps, formalized by the following three lemmas. We begin by reducing the temporal depth to at most two by adapting a construction of Demri and Schnoebelen, which associates to every LTL formula an equisatisfiable formula with temporal depth at most two [7].

▶ **Lemma 6.** *For every HyperLTL sentence $Q_1\pi_1 \dots Q_n\pi_n.\psi$ with quantifier-free ψ, one can compute in polynomial time an equisatisfiable sentence $Q_1\pi_1 \dots Q_n\pi_n.\exists\pi.\psi'$ with quantifier-free ψ' and temporal depth at most two.*

The idea is to add atomic propositions to witness the satisfaction of subformulas ψ' of ψ. We express the existence, for every n-tuple of traces $(t_1, \dots, t_n)$ of the model, of a *witness trace*. For all $j \in \mathbb{N}$, for all subformulas ψ' of ψ, the valuation $[(\pi_i \to t_i[j,\infty))_{1\leq i\leq n}]$ satisfies ψ' if and only if the associated atomic proposition is satisfied at position j of the witness trace.

Next, we turn the quantifier prefix into the form $\forall^*\exists^*$ without increasing the temporal depth.

▶ **Lemma 7.** *For every HyperLTL sentence φ, one can compute in polynomial time an equisatisfiable sentence φ' of the form $\forall^*\exists^*\psi$ with $\mathrm{td}(\varphi') = \max(\mathrm{td}(\varphi), 1)$.*

Here the key idea is to move existential quantifiers in the scope of universal ones after marking them with fresh atomic propositions: We can replace an $\exists\forall$ by a $\forall\exists$ if we require that the existentially quantified variable is now uniquely marked by a proposition (and therefore cannot depend on the universally quantified variable).

The construction presented in the proof of Lemma 7 may increase the number of universally quantified variables, but we can decrease that number to two without increasing the temporal or alternation depth. This step also completes the proof of Theorem 5.

▶ **Lemma 8.** *For every HyperLTL sentence φ of the form $\forall^*\exists^*\psi$ with quantifier-free ψ, one can compute in polynomial time an equisatisfiable sentence φ' of the form $\forall^2\exists^*\psi'$ where ψ' is quantifier-free and $\mathrm{td}(\varphi') = \max(\mathrm{td}(\varphi), 1)$.*

This can be achieved by merging several traces into one. To this end, we increase the set of atomic propositions by considering as new atomic propositions tuples of the previous atomic propositions, i.e., one trace now encodes a tuple of traces. However we cannot decrease the number of universal quantifiers below two this way, as we need two universal quantifiers to ensure that every possible combination of traces is represented in the model, i.e., any model of the resulting formula is the set of mergings of traces of another model.

Thus, $\forall^2\exists^*$ formulas with temporal depth two capture the complete complexity of the satisfiability problem for HyperLTL. As the latter problem is undecidable and as all reductions presented above are effective, we immediately obtain that satisfiability for $\forall^2\exists^*$ formulas with temporal depth two is also undecidable.

As alluded to above, an even stronger result can be obtained by strengthening the proof of Finkbeiner and Hahn for $\forall\exists$ formulas to only use temporal depth two.[1] Thus, only formulas of temporal depth one remain to be considered.

Before we start investigating this class let us quickly comment on why we disregard temporal depth zero: Every such sentence can easily be turned to an equisatisfiable instance of QBF, which is known to be solvable in polynomial space.

Thus, it only remains to consider formulas with arbitrary quantifier prefixes, but temporal depth one. Our main result of this section shows that even this problem is undecidable, even for HyperLTL$^1(\mathbf{F}, \mathbf{G})$ formulas with alternation depth one. Due to the restriction on the temporal depth, our encoding of a Minsky machine is more complicated than it would be with arbitrary temporal depth.

[1] Alternatively, one can also obtain a direct reduction from the Turing machine immortality problem [17] to satisfiability of $\forall\exists$ sentences of temporal depth two.

▶ **Theorem 9.** *The following problem is undecidable: Given a $\forall^2\exists^*$ HyperLTL$^1(\mathbf{F},\mathbf{G})$ sentence φ, is φ satisfiable?*

Proof. We reduce from the (non)-halting problem for 2-counter Minsky machines. Recall that such a machine can be seen as a tuple $\mathcal{M} = (Q, \Delta, q_0)$ where Q is a finite set of states, $q_0 \in Q$ an initial state, and $\Delta \subseteq Q \times \{1, 2\} \times OP \times Q$ a set of transition rules, where $OP = \{\texttt{++}, \texttt{--}, \texttt{=0?}\}$. A configuration of $\mathcal{M}$ is an element of $Q \times \mathbb{N} \times \mathbb{N}$. For all $n, n' \in \mathbb{N}$, $op \in OP$ we write $n \xrightarrow{op} n'$ if:

- op is ++ and $n' = n + 1$.
- op is -- and $n' = n - 1$ (note that this operation is only applicable if $n > 0$).
- op is =0? and $n' = n = 0$.

There is a transition from (q, n_1, n_2) to (q', n'_1, n'_2) if and only if there is an $i \in \{1, 2\}$ and $op \in OP$ with $(q, i, op, q') \in \Delta$, $n_{3-i} = n'_{3-i}$, and $n_i \xrightarrow{op} n'_i$. It is undecidable whether such a machine has an infinite computation $(q_0, 0, 0) \to (q_1, n^1_1, n^1_2) \to (q_2, n^2_1, n^2_2) \to \cdots$.

Let $\mathcal{M} = (Q, \Delta, q_0)$ be a 2-counter Minsky machine. We use $\mathrm{AP} = Q \cup \{1, 2\}$ as atomic propositions. Given $i \in \{1, 2\}$, we denote by $\bar{i}$ the other proposition. Consider the formula $\psi_1 = \forall\pi.\forall\pi'.\mathbf{G}\,(1_\pi \Rightarrow 1_{\pi'}) \vee \mathbf{G}\,(1_{\pi'} \Rightarrow 1_\pi)$. We define ψ_2 with $2 \in \mathrm{AP}$ analogously. In the following, we only consider sets of traces that satisfy $\psi_1 \wedge \psi_2$.

For each trace $t \in (2^{\mathrm{AP}})^\omega$ and $i \in \{1, 2\}$, we define the *i-set* of t as $S_i(t) = \{j \in \mathbb{N} \mid i \in t(j)\}$. Now fix $T \subseteq (2^{\mathrm{AP}})^\omega$ that satisfies $\psi_1 \wedge \psi_2$. We define the pre-order $\leq_i$ on T as follows: for all $t, t' \in T$, $t \leq_i t'$ if and only if $S_i(t) \subseteq S_i(t')$. It is straightforward to verify that $\leq_i$ is indeed reflexive and transitive. We write $t <_i t'$ if $S_i(t) \subsetneq S_i(t')$. As T satisfies $\psi_1 \wedge \psi_2$, the $\leq_i$ are total pre-orders on T. We also define for all $t \in T$ and $i \in \{1, 2\}$, the *rank* of t with respect to i as $\mathrm{rk}_i(t) = |\{S_i(t') \mid t' \in T \text{ and } t' <_i t\}|$, which may be infinite. Note that if $S_i(t) = \emptyset$ then $\mathrm{rk}_i(t) = 0$, and that if $S_i(t) = S_i(t')$ then $\mathrm{rk}_i(t) = \mathrm{rk}_i(t')$. Also, note that the rank depends on the fixed set T of traces under consideration.

Finally, as $\leq_i$ is a total pre-order, if we have $t <_i t'$, but there is no t'' with $t <_i t'' <_i t'$, then $\mathrm{rk}_i(t') = \mathrm{rk}_i(t)+1$. Note that this holds even when $\mathrm{rk}_i(t)$ is infinite, assuming $\infty+1 = \infty$.

We construct a HyperLTL$^1(\mathbf{F},\mathbf{G})$ formula φ that encodes the existence of an infinite computation $(q_0, 0, 0) \to (q_1, n^1_1, n^1_2) \to (q_2, n^2_1, n^2_2) \to \cdots$ of $\mathcal{M}$. In a model T of φ, a configuration (q, n_1, n_2) is encoded by a trace t with $t(0) \cap Q = \{q\}$ and for $i \in \{1, 2\}$, $rk_i(t) = n_i$. Then, φ states the existence of an initial trace t_0, representing the configuration $(q_0, 0, 0)$, as well as the existence of a successor t' encoding (q', n'_1, n'_2) for each trace t encoding (q, n_1, n_2), i.e., we require $(q, n_1, n_2) \to (q', n'_1, n'_2)$. The latter is witnessed by the existence of a transition (q, i, op, q') such that:

1. $t(0) \cap Q = \{q\}$ and $t'(0) \cap Q = \{q'\}$, i.e., t and t' indeed encode the states of their respective configurations correctly.
2. For all $j \in \mathbb{N}$, $\bar{i} \in t(j)$ if and only if $\bar{i} \in t'(j)$, i.e. $S_{\bar{i}}(t) = S_{\bar{i}}(t')$. Thus, as argued above, $\mathrm{rk}_{\bar{i}}(t) = \mathrm{rk}_{\bar{i}}(t')$, which implies $n_{\bar{i}} = n'_{\bar{i}}$.
3. If op is ++ then $t <_i t'$ and there does not exist any t'' such that $t <_i t'' <_i t'$, i.e., $\mathrm{rk}_i(t') = \mathrm{rk}_i(t) + 1$, as $\leq_i$ is a total pre-oder. Then, we have $n'_i = n_i + 1$.
4. If op is -- then $t >_i t'$ and there does not exist any t'' such that $t >_i t'' >_i t'$, i.e., $\mathrm{rk}_i(t') = \mathrm{rk}_i(t) - 1$, as $\leq_i$ is a total pre-oder. Then, we have $n'_i = n_i - 1$.
5. If op is =0? then for all $j \in \mathbb{N}$, $i \notin t(j)$ and $i \notin t'(j)$. Hence, $S_i(t) = S_i(t') = \emptyset$, i.e., $\mathrm{rk}_i(t) = \mathrm{rk}_i(t') = 0$, which implies $n_i = n'_i = 0$.

We encode those conditions in φ, which is the conjunction of the following three sentences and of $\psi_1 \wedge \psi_2$:

- $\varphi_1 = \forall\pi. \bigwedge_{q \neq q' \in Q} q_\pi \Rightarrow \neg q'_\pi$ expresses that a trace is associated to at most one state.
- $\varphi_2 = \exists\pi_0.(q_0)_{\pi_0} \wedge \mathbf{G}\,(\neg 1_{\pi_0} \wedge \neg 2_{\pi_0})$ expresses the existence of a trace representing the initial configuration $(q_0, 0, 0)$.

- $\varphi_3 = \forall\pi.\exists\pi'.\bigvee_{(q,i,op,q')\in\Delta} q_\pi \wedge q'_{\pi'} \wedge \varphi_{i,op} \wedge \mathbf{G}\,(\bar{i}_\pi \Leftrightarrow \bar{i}_{\pi'})$ expresses that all traces have a successor obtained by faithfully simulating a transition of the machine.

Here, we use the formulas

- $\varphi_{1,\texttt{++}} = \forall\pi''.\pi <_1 \pi' \wedge (\pi'' \leq_1 \pi \vee \pi' \leq_1 \pi'')$,
- $\varphi_{1,\texttt{--}} = \forall\pi''.\pi >_1 \pi' \wedge (\pi'' \geq_1 \pi \vee \pi' \geq_1 \pi'')$, and
- $\varphi_{1,\texttt{=0?}} = \mathbf{G}\,(\neg 1_\pi \wedge \neg 1_{\pi'})$,

where $\pi \leq_1 \pi' = \mathbf{G}\,(1_\pi \Rightarrow 1_{\pi'})$ and $\pi <_1 \pi' = \pi \leq_1 \pi' \wedge \mathbf{F}\,(\neg 1_\pi \wedge 1_{\pi'})$. Finally, we define the formulas $\leq_2$, $<_2$, and $\varphi_{2,op}$ analogously.

The sentence φ is not in prenex normal form. However, as no quantifier appears in the scope of a temporal operator, it can be put in that form. Further, it is not of the form $\forall^2\exists^*$, but we can apply Lemmas 7 and 8 to bring it into this form while preserving the temporal depth, which is already one. We claim that φ is satisfiable if and only if $\mathcal{M}$ has an infinite computation starting in $(q_0, 0, 0)$.

Suppose φ is satisfied by a model T. The subformulas φ_1 and φ_2 enforce that T contains a trace t_0 encoding the initial configuration $(q_0, 0, 0)$ of $\mathcal{M}$. Further, φ_3 expresses that every trace t encoding a configuration (q, n_1, n_2) has a successor t' encoding a configuration (q', n'_1, n'_2) with $(q, n_1, n_2) \to (q', n'_1, n'_2)$. Thus, there exists an infinite sequence $t_0, t_1, t_2, \ldots$ of traces encoding an infinite run of $\mathcal{M}$.

Conversely, suppose $\mathcal{M}$ has an infinite run $(q_0, 0, 0) \to (q_1, n_1^1, n_2^1) \to (q_2, n_1^2, n_2^2) \cdots$, then for all j let t_j be the trace whose projection to Q is $\{q_j\}\emptyset^\omega$, and whose projection to $\{i\}$ is $\{i\}^{n_i^j}\emptyset^\omega$ for $i \in \{1, 2\}$. One can then easily check that $\{t_j \mid j \in \mathbb{N}\}$ is a model of φ. ◀

Thus, two universal quantifiers before some existential ones and using only $\mathbf{F}$ and $\mathbf{G}$ without nesting yields undecidable satisfiability. Our next result shows that removing one of the two universal quantifiers allows us to recover decidability, even when allowing nested next operators and leading existential quantifiers.

As a first step in the proof, we show that the nested next operators can be eliminated without introducing additional universal quantifiers. This is true, as we are only interested in satisfiability.

▶ **Lemma 10.** *For every $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ sentence, one can construct in polynomial time an equisatisfiable $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G})$ sentence.*

Now, we are ready to prove our main decidability result in this section. Note that we do not claim a matching lower bound here. We comment on this gap in the conclusion.

▶ **Theorem 11.** *The following problem is in* N2EXPTIME*: Given a HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ sentence φ of the form $\exists^*\forall\exists^*$, is φ satisfiable?*

Proof. Let $\varphi = \exists\pi_1 \ldots \pi_n.\forall\pi.\exists\pi_{n+1} \ldots \exists\pi_{n+n'}.\psi$ be a HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ sentence with quantifier-free ψ. Due to Lemma 10, it is enough to consider the case where ψ is a Boolean combination of formulas of the form $\mathbf{F}\,\beta$ for a Boolean combination β of atomic propositions.

Note that such a formula can only specify the appearance or non-appearance of combinations of atomic propositions on the quantified traces, but not the order of these combinations. Hence, to every tuple $(t_1, \ldots, t_k)$ of traces $t_i \in (2^{\mathrm{AP}})^\omega$, we associate a finite set of tuples of valuations $V(t_1, \ldots, t_k) = \{(t_1(j), \ldots, t_k(j)) \mid j \in \mathbb{N}\} \subseteq (2^{\mathrm{AP}})^k$, i.e., the set all the tuples of valuations that appear eventually. The cardinality of $V(t_1, \ldots, t_k)$ is at most $2^{k|\mathrm{AP}|}$.

Let β be a Boolean combination of atomic propositions over trace variables $\pi_1, \ldots, \pi_k$. Then, a trace assignment $[(\pi_i \to t_i)_{1\leq i\leq k}]$ satisfies $\mathbf{F}\,\beta$ if and only if there exists $j \in \mathbb{N}$ such that β is satisfied at position j of $(t_1, \ldots, t_k)$, i.e., there exists $(v_1, \ldots, v_k) \in V(t_1, \ldots, t_k)$ such that $(v_1, \ldots, v_k)$ satisfies β (in the sense that any trace assignment Π such that $\Pi(\pi_i)(0) = v_i$ for all i satisfies β). Intuitively, we abstract a tuple of traces into a finite set of tuples of valuations, and then abstract a model as a set of such finite representations. Then, we show that satisfiability can be decided using such abstractions.

So, whether a given trace assignment $[(\pi_i \to t_i)_{1 \le i \le k}]$ satisfies a given Boolean combination ψ of formulas $\mathbf{F}\,\beta$ only depends on $V(t_1, \ldots, t_k)$, and given $V \subseteq (2^{\mathrm{AP}})^k$, one can check in polynomial time whether a trace assignment yielding V satisfies ψ. If it is the case, we say that V satisfies ψ.

To check the satisfiability of φ, we start by nondeterministically guessing a set $S \subseteq 2^{(2^{\mathrm{AP}})^{n+n'+1}}$ of sets of $(n + n' + 1)$-tuples of valuations. This set is supposed to represent a model of φ. The n first valuations represent the fixed values assigned to $\tau_1, \ldots, \tau_n$. The $(n + 1)$-th represents the valuation of the universally quantified variable. Thus, for every trace of the model there must exist a tuple in which that trace is represented at position $n + 1$. The valuations of positions $n + 2$ to $n + n'$ have to be such that φ is satisfied by all tuples.

Thus, we check the following requirements:

1. For all $V_1, V_2 \in S$, $\{(v_1, \ldots, v_n) \mid (v_1, \ldots, v_{n+n'+1}) \in V_1\}$ is equal to $\{(v_1, \ldots, v_n) \mid (v_1, \ldots, v_{n+n'+1}) \in V_2\}$: The set of values taken by the traces assigned to $\tau_1, \ldots, \tau_n$ cannot depend on the values of the other variables. Thus, we ensure that these values are fixed in the guessed model.
2. For all $V \in S$ and $1 \le i \le n + n' + 1$, there exists $V' \in S$ such that $\{(v_1, \ldots, v_n, v_i) \mid (v_1, \ldots, v_{n+n'+1}) \in V\} = \{(v_1, \ldots, v_{n+1}) \mid (v_1, \ldots, v_{n+n'+1}) \in V'\}$. All the values taken by the existentially quantified variables have to be taken by the universally quantified one as well.
3. For all $V \in S$, V satisfies ψ.

If all requirements are satisfied, we accept, otherwise we reject. This procedure requires nondeterministic doubly-exponential time as $|S| \le 2^{2^{|\mathrm{AP}|+n+n'+1}}$.

Suppose φ is satisfiable and fix a model T. There exist $t_1, \ldots, t_n \in T$ such that $(T, [(\tau_i \to t_i)_{1 \le i \le n}]) \models \forall \pi \exists \tau_{n+1} \ldots \exists \tau_{n+n'}.\psi$. Furthermore, for a fixed $t \in T$ there exist $t_{n+1}, \ldots, t_{n+n'} \in T$ such that $(T, [(\tau_i \to t_i)_{1 \le i \le n+n'}, \pi \to t]) \models \psi$. Let $V^*(t) = \{(t_1(j), \ldots, t_n(j), t(j), t_{n+1}(j), \ldots, t_{n+n'}(j)) \mid j \in \mathbb{N}\}$.

Now, one can easily check that Requirements 1, 2, and 3 are satisfied by $\{V^*(t) \mid t \in T\}$. Thus, the algorithm accepts φ.

Conversely, suppose the algorithm accepts φ. Then, there exists some S satisfying all three requirements above. We construct from S a model T of φ.

Let $t_1, \ldots, t_n$ be traces such that for all $V \in S$, $\{(v_1, \ldots, v_n) \mid (v_1, \ldots, v_{n+n'+1}) \in V\} = V(t_1, \ldots, t_n)$, and for all $(v_1, \ldots, v_{n+n'+1}) \in V$, $(v_1, \ldots, v_n) = (t_1(j), \ldots, t_n(j))$ for infinitely many j, i.e., each of the valuations appears infinitely often in the traces. Those traces can be constructed due to Requirement 1.

Let $T_0 = \{t_1, \ldots, t_n\}$. For all $\ell \in \mathbb{N}$ we construct T_ℓ by induction on $\ell \in \mathbb{N}$, while maintaining the following two invariants:

1. For all $t \in T_\ell$ there exists $V \in S$ such that $V(t_1, \ldots, t_n, t)$ is equal to $\{(v_1, \ldots, v_{n+1}) \mid (v_1, \ldots, v_{n+n'+1}) \in V\}$, and for all $(v_1, \ldots, v_{n+n'+1}) \in V$, $(v_1, \ldots, v_{n+1})$ is equal to $(t_1(j), \ldots, t_n(j), t(j))$ for infinitely many j, where the t_i are the traces in T_0.
2. If $\ell > 0$ then for every $t \in T_{\ell-1}$, there exist traces $t_{n+1}, \ldots, t_{n+n'} \in T_\ell$ such that $[(\tau_i \to t_i)_{1 \le i \le n+n'}, \pi \to t] \models \psi$.

By Requirement 2 and by construction, T_0 satisfies Invariant 1, and it clearly satisfies Invariant 2. Let $\ell \in \mathbb{N}$, suppose T_ℓ has been constructed, and that it satisfies Invariants 1 and 2. By Invariant 1, for all $t \in T_\ell$ we can construct traces $t_{n+1}, \ldots, t_{n+n'}$ such that $V(t_1, \ldots, t_n, t, t_{n+1}, \ldots, t_{n+n'}) \in S$ and for all $(v_1, \ldots, v_n, v, v_{n+1}, \ldots, v_{n+n'}) \in V(t_1, \ldots, t_n, t, t_{n+1}, \ldots, t_{n+n'})$, it is the case that $(v_1, \ldots, v_n, v, v_{n+1}, \ldots, v_{n+n'})$ is equal to $(t_1(j), \ldots, t_n(j), t(j), t_{n+1}(j), \ldots, t_{n+n'}(j))$ for infinitely many j (as all the $(v_1, \ldots, v_n, v)$ appear infinitely many times in $(t_1, \ldots, t_n, t)$ by Invariant 1). Let $I(t) = \{t_{n+1}, \ldots, t_{n+n'}\}$. Let $T_{\ell+1} = \bigcup_{t \in T} I(t)$, which satisfies Invariant 1 by Requirement 2. It also satisfies Invariant 2 by definition. Furthermore, by Requirement 3, $V(t_1, \ldots, t_n, t, t_{n+1}, \ldots, t_{n+n'})$ satisfies ψ.

Finally, let $T = \bigcup_{\ell \in \mathbb{N}} T_\ell$ and let $t \in T$. Then, there exists an ℓ such that $t \in T_\ell$. Thus, there also exist $t_{n+1}, \ldots, t_{n+n'} \in T_{\ell+1}$ such that $[(\tau_i \to t_i)_{1 \leq i \leq n+n'}, \pi \to t]$ satisfies ψ. Therefore, T satisfies φ. ◀

Recall that satisfiability of $\exists^*\forall^*$ formulas is EXPSPACE-complete [8]. The proof of Finkbeiner and Hahn can be slightly adapted to produce a formula of temporal depth two: their approach states the existence of a trace representing a sequence of configurations of an exponential-space bounded Turing machine. The only difficulty that can arise in expressing the correctness of the run described by that trace is relating a position of one of the configurations to the neighbouring positions in the next configuration (in order to simulate the movement of the head). One may then require to combine an until and a next in order to express this requirement, in the scope of an always expressing that it holds for every position. This nesting can be removed by adding a fresh proposition p that is satisfied on all positions of the first configuration, on none of the second one, and so on, i.e., its truth value alternates between the configurations. One can then express the previous requirement with a single until in the scope of an always, yielding temporal depth two.

Our next result shows that one obtains better complexity when restricting the temporal depth of formulas to one.

▶ **Theorem 12.** *The following problem is* NEXPTIME*-complete: Given an $\exists^*\forall^*$ HyperLTL sentence φ with temporal depth one, is φ satisfiable?*

We adapt the proof of Finkbeiner and Hahn for EXPSPACE-completeness of the problem with arbitrary temporal depth [8], i.e., we turn the HyperLTL formula into an exponentially larger equisatisfiable LTL one (cp. the proof of Theorem 1). The decrease in complexity is a consequence of the switch from PSPACE to NP of the complexity of LTL satisfiability when restricting temporal depth to one [7].

We conclude by considering the satisfiability problem for HyperLTL$^1(\mathbf{F}, \mathbf{G})$ with arbitrary quantifier prefixes, but restricted to models induced by finite-state systems. The undecidability of satisfiability for arbitrary formulas over finite-state systems can be easily inferred from the proof of undecidability of satisfiability of Finkbeiner and Hahn, as the formulas they construct, if satisfiable, have a finite and ultimately periodic model, which is therefore representable by a finite-state system. For formulas of HyperLTL$^1(\mathbf{F}, \mathbf{G})$, we leave decidability open, but prove intractability.

▶ **Theorem 13.** *The following problem is* TOWER*-hard: Given a HyperLTL$^1(\mathbf{F}, \mathbf{G})$ sentence φ, does φ have a model $T(\mathcal{K})$ for some Kripke structure $\mathcal{K}$?*

Let us conclude by remarking that the satisfiability problem for HyperLTL$^1(\mathbf{F}, \mathbf{G})$ over Kripke structures is different from the general one, i.e., there are satisfiable formulas which are not satisfied by the set of traces of any Kripke structure. Consider for instance the sentence $\varphi = \forall\pi.\exists\pi'.\mathbf{G}\,(a_\pi \Rightarrow a_{\pi'}) \wedge \mathbf{F}\,(\neg a_\pi \wedge a_{\pi'})$, which is satisfied by $\{a\}^*\emptyset^\omega$.

Suppose there exists a Kripke structure $\mathcal{K}$ with a set of traces satisfying this sentence. We define inductively an increasing sequence of finite trace prefixes p_n for $n \in \mathbb{N}$ as $p_0 = \varepsilon$ and $p_{n+1} = p_n\{a\}$ if $p_n\{a\}$ is a prefix of a trace of $\mathcal{K}$, and $p_{n+1} = p_n\emptyset$ otherwise. Let t be the limit of the sequence $(p_n)_{n\in\mathbb{N}}$, i.e., the unique trace with prefix p_n for every n. As the p_n are prefixes of traces of $\mathcal{K}$, t itself is a trace of $\mathcal{K}$. As $\mathcal{K}$ satisfies φ, there exists t' such that for all j, if $a \in t(j)$ then $a \in t'(j)$ and there exists j^* such that $a \in t'(j^*)$ and $a \notin t(j^*)$. In particular, there exists a minimal such j^*. Then $p_{j^*+1} = p_{j^*}\emptyset$, but $p_{j^*}\{a\}$ is a prefix of t'. This contradicts the choice of p_{j^*+1}, as we prefer to extend by $\{a\}$ instead of $\emptyset$. Thus, the satisfiable sentence φ is not satisfiable by the set of traces of a finite Kripke structure.

5 Conclusion

We have shown that HyperLTL satisfiability can be decidable, either if one restricts the space of models one is interested in to sufficiently simple ones, or if one restricts the alternation and temporal depth of the formulas under consideration. In particular, we have investigated the formulas of temporal depth one without untils. An interesting open problem is to extend the decidability result presented in Theorem 11 to formulas with untils. Also, we claimed no lower bound on the problem solved in Theorem 11. We claim there is an ExpSpace lower bound obtained by encoding exponential space Turing machines, but the exact complexity of the problem is left open. Another interesting problem left open is the decidability of HyperLTL$^1(\mathbf{F}, \mathbf{G})$ over Kripke structures. We have presented a Tower lower bound in Theorem 13, but it is open whether the problem is indeed decidable.

In general, restricting the space of models turns out to be more fruitful than to restrict the formulas under consideration, as satisfiability is undecidable for extremely simple formulas (simplicity being measured in alternation depth and temporal depth). An interesting challenge pertains to finding other measures of simplicity that yield larger decidable fragments.

References

1 Shreya Agrawal and Borzoo Bonakdarpour. Runtime Verification of k-Safety Hyperproperties in HyperLTL. In *CSF 2016*, pages 239–252. IEEE Computer Society, 2016. doi:10.1109/CSF.2016.24.

2 Borzoo Bonakdarpour and Bernd Finkbeiner. Runtime Verification for HyperLTL. In Yliès Falcone and César Sánchez, editors, *RV 2016*, volume 10012 of *LNCS*, pages 41–45. Springer, 2016. doi:10.1007/978-3-319-46982-9_4.

3 Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. Temporal Logics for Hyperproperties. In Martín Abadi and Steve Kremer, editors, *POST 2014*, volume 8414 of *LNCS*, pages 265–284. Springer, 2014. doi:10.1007/978-3-642-54792-8_15.

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

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

6 Norine Coenen, Bernd Finkbeiner, César Sánchez, and Leander Tentrup. Verifying hyperliveness. In Isil Dillig and Serdar Tasiran, editors, *CAV 2019*, volume 11561 of *LNCS*, pages 121–139. Springer, 2019. doi:10.1007/978-3-030-25540-4_7.

7 Stéphane Demri and Philippe Schnoebelen. The Complexity of Propositional Linear Temporal Logics in Simple Cases. *Inf. Comput.*, 174(1):84–103, 2002. doi:10.1006/inco.2001.3094.

8 Bernd Finkbeiner and Christopher Hahn. Deciding Hyperproperties. In Josée Desharnais and Radha Jagadeesan, editors, *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.

9 Bernd Finkbeiner, Christopher Hahn, and Tobias Hans. MGHyper: Checking Satisfiability of HyperLTL Formulas Beyond the $\exists^*\forall^*$ Fragment. In Shuvendu K. Lahiri and Chao Wang, editors, *ATVA 2018*, volume 11138 of *LNCS*, pages 521–527. Springer, 2018. doi:10.1007/978-3-030-01090-4_31.

10 Bernd Finkbeiner, Christopher Hahn, Philip Lukert, Marvin Stenger, and Leander Tentrup. Synthesizing Reactive Systems from Hyperproperties. In Hana Chockler and Georg Weissenbacher, editors, *CAV 2018 (Part I)*, volume 10981 of *LNCS*, pages 289–306. Springer, 2018. doi:10.1007/978-3-319-96145-3_16.

11 Bernd Finkbeiner, Christopher Hahn, and Marvin Stenger. EAHyper: Satisfiability, implication, and equivalence checking of hyperproperties. In Rupak Majumdar and Viktor Kuncak, editors, *CAV 2017 (Part II)*, volume 10427 of *LNCS*, pages 564–570. Springer, 2017. doi:10.1007/978-3-319-63390-9_29.

12 Bernd Finkbeiner, Christopher Hahn, Marvin Stenger, and Leander Tentrup. RVHyper: A runtime verification tool for temporal hyperproperties. In Dirk Beyer and Marieke Huisman, editors, *TACAS 2018 (Part II)*, volume 10806 of *LNCS*, pages 194–200. Springer, 2018. doi:10.1007/978-3-319-89963-3_11.

13 Bernd Finkbeiner, Markus N. Rabe, and César Sánchez. Algorithms for Model Checking HyperLTL and HyperCTL*. In Daniel Kroening and Corina S. Pasareanu, editors, *CAV 2015 (Part I)*, volume 9206 of *LNCS*, pages 30–48. Springer, 2015. doi:10.1007/978-3-319-21690-4_3.

14 Bernd Finkbeiner and Martin Zimmermann. The First-Order Logic of Hyperproperties. In Heribert Vollmer and Brigitte Vallée, editors, *STACS 2017*, volume 66 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.30.

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

16 Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, and Victor Vianu. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic*, 7(2):213–236, 2001. doi:10.2307/2687775.

17 Philip K. Hooper. The undecidability of the Turing machine immortality problem. *J. Symb. Log.*, 31(2):219–234, 1966. doi:10.2307/2269811.

18 Robert P. Kurshan. Transfer of Model Checking to Industrial Practice. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking.*, pages 763–793. Springer, 2018. doi:10.1007/978-3-319-10575-8_23.

19 Daryl McCullough. Noninterference and the composability of security properties. In *1988 IEEE Symposium on Security and Privacy*, pages 177–186. IEEE Computer Society, 1988. doi:10.1109/SECPRI.1988.8110.

20 Daryl McCullough. A Hookup Theorem for Multilevel Security. *IEEE Trans. Software Eng.*, 16(6):563–568, 1990. doi:10.1109/32.55085.

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

22 Jonathan K. Millen. Unwinding Forward Correctability. *Journal of Computer Security*, 3(1):35–54, 1995. doi:10.3233/JCS-1994/1995-3104.

23 Amir Pnueli. The Temporal Logic of Programs. In *FOCS 1977*, pages 46–57, 1977.

24 Sylvain Schmitz. Complexity Hierarchies beyond Elementary. *TOCT*, 8(1):3:1–3:36, 2016. doi:10.1145/2858784.

25 A. Prasad Sistla and Edmund M. Clarke. The Complexity of Propositional Linear Temporal Logics. *Journal of the ACM*, 32(3):733–749, 1985. doi:10.1145/3828.3837.

26 Larry J. Stockmeyer and Albert R. Meyer. Word Problems Requiring Exponential Time: Preliminary Report. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd,

Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *STOC 1973*, pages 1–9. ACM, 1973. doi:10.1145/800125.804029.

27 Steve Zdancewic and Andrew C. Myers. Observational Determinism for Concurrent Program Security. In *CSFW 2003*, page 29. IEEE Computer Society, 2003. doi:10.1109/CSFW.2003.1212703.