The Complexity of Second-Order HyperLTL

Hadar Frenkel
Bar-Ilan University, Ramat Gan, Israel

Martin Zimmermann
Aalborg University, Denmark

Abstract

We determine the complexity of second-order HyperLTL satisfiability, finite-state satisfiability, and model-checking: All three are equivalent to truth in third-order arithmetic.

We also consider two fragments of second-order HyperLTL that have been introduced with the aim to facilitate effective model-checking by restricting the sets one can quantify over. The first one restricts second-order quantification to smallest/largest sets that satisfy a guard while the second one restricts second-order quantification further to least fixed points of (first-order) HyperLTL definable functions. All three problems for the first fragment are still equivalent to truth in third-order arithmetic while satisfiability for the second fragment is Σ_1^1-complete, i.e., only as hard as for (first-order) HyperLTL and therefore much less complex. Finally, finite-state satisfiability and model-checking are in Σ_2^2 and are Σ_1^1-hard, and thus also less complex than for full second-order HyperLTL.

2012 ACM Subject Classification Theory of computation → Verification by model checking; Theory of computation → Logic and verification

Keywords and phrases HyperLTL, Satisfiability, Model-checking

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.10

Related Version *Full Version*: `https://arxiv.org/abs/2311.15675` [21]

Funding *Martin Zimmermann*: Supported by DIREC – Digital Research Centre Denmark.

Acknowledgements This work was initiated by a discussion at Dagstuhl Seminar 23391 "The Futures of Reactive Synthesis" and some results were obtained at Dagstuhl Seminar 24111 "Logics for Dependence and Independence: Expressivity and Complexity". We are grateful to Gaëtan Regaud for finding and fixing a bug in the proof of Theorem 18 and to the reviewers for their detailed and valuable feedback, which improved the paper considerably.

1 Introduction

The introduction of hyperlogics [11] for the specification and verification of hyperproperties [12] – properties that relate multiple system executions, has been one of the major success stories of formal verification during the last decade. Logics like HyperLTL and HyperCTL* [11], the extensions of LTL [32] and CTL* [14] (respectively) with trace quantification, are natural specification languages for information-flow and security properties, have a decidable model-checking problem [17], and hence found many applications in program verification.

However, while expressive enough to express common information-flow properties, they are unable to express other important hyperproperties, e.g., common knowledge in multi-agent systems and asynchronous hyperproperties (witnessed by a plethora of asynchronous extensions of HyperLTL, e.g., [1, 2, 3, 6, 9, 10, 23, 26, 27, 28]). These examples all have in common that they are *second-order* properties, i.e., they naturally require quantification over *sets* of traces, while HyperLTL (and HyperCTL*) only allows quantification over traces.

33rd EACSL Annual Conference on Computer Science Logic (CSL 2025).
Editors: Jörg Endrullis and Sylvain Schmitz; Article No. 10; pp. 10:1–10:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In light of this situation, Beutner et al. [4] introduced the logic Hyper^{2}LTL, which extends HyperLTL with second-order quantification, i.e., quantification over sets of traces. They show that the resulting logic, Hyper^{2}LTL, is indeed able to capture common knowledge, asynchronous extensions of HyperLTL, and many other applications.

Consider, e.g., common knowledge in multi-agent systems where each agent i only observes some parts of the system. The agent *knows* that a statement φ holds if it holds on all traces that are *indistinguishable* in the agent's view. We write $\pi \sim_i \pi'$ if the traces π and π' are indistinguishable for agent i. A property φ is common knowledge among all agents if all agents know φ, all agents know that all agents know φ, and so on, i.e., one takes the infinite closure of knowledge among all agents. This infinite closure cannot be expressed using first-order quantification over traces [8], like the one used in HyperLTL. The second-order quantification suggested by Beutner et al. allows us to express common knowledge, as demonstrated by the formula φ_{ck}, which states that φ is common knowledge on all traces of the system (we use a simplified syntax for readability):

$$\varphi_{ck} = \forall \pi.\, \exists X.\, \pi \in X \wedge \Big(\forall \pi' \in X.\, \forall \pi''.\, \big(\bigvee\nolimits_{i=1}^{n} \pi' \sim_i \pi''\big) \rightarrow \pi'' \in X\Big) \wedge \forall \pi' \in X.\, \varphi(\pi')$$

The formula φ_{ck} expresses that for every trace t (instantiating π), there exists a set T (an instantiation of the second-order variable X) such that t is in T, T is closed under the observations of all agents (if t' is in T and t'' is indistinguishable from t' for some agent i, then also t'' is in T), and all traces in T satisfy φ.

However, Beutner et al. also note that this expressiveness comes at a steep price: model-checking Hyper^{2}LTL is highly undecidable, i.e., Σ_1^1-hard. Thus, their main result is a partial model-checking algorithm for a fragment of Hyper^{2}LTL where second-order quantification degenerates to least fixed point computations of HyperLTL definable functions. Their algorithm over- and underapproximates these fixed points and then invokes a HyperLTL model-checking algorithm on these approximations. A prototype implementation of the algorithm is able to model-check properties capturing common knowledge, asynchronous hyperproperties, and distributed computing.

However, one question has been left open: Just how complex is Hyper^{2}LTL verification?

Complexity Classes for Undecidable Problems. The complexity of undecidable problems is typically captured in terms of the arithmetical and analytical hierarchy, where decision problems (encoded as subsets of $\mathbb{N}$) are classified based on their definability by formulas of higher-order arithmetic, namely by the type of objects one can quantify over and by the number of alternations of such quantifiers. We refer to Roger's textbook [35] for fully formal definitions and refer to Figure 1 for a visualization.

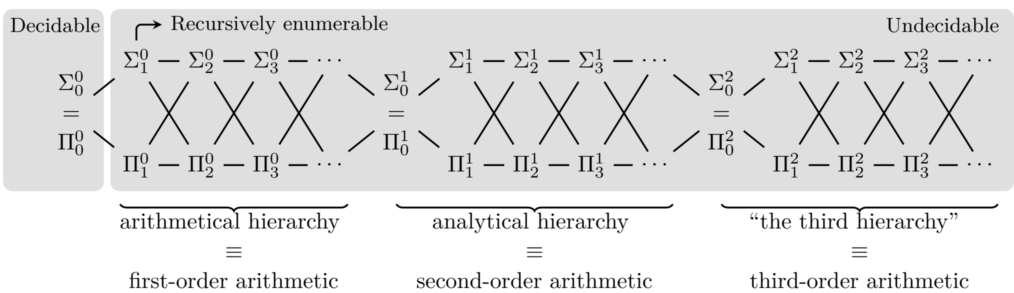

Figure 1 The arithmetical hierarchy, the analytical hierarchy, and beyond.

The class Σ^0_1 contains the sets of natural numbers of the form

$$\{x \in \mathbb{N} \mid \exists x_0. \cdots \exists x_k. \ \psi(x, x_0, \dots, x_k)\}$$

where quantifiers range over natural numbers and ψ is a quantifier-free arithmetic formula. Note that this is exactly the class of recursively enumerable sets. The notation Σ^0_1 signifies that there is a single block of existential quantifiers (the subscript 1) ranging over natural numbers (type 0 objects, explaining the superscript 0). Analogously, Σ^1_1 is induced by arithmetic formulas with existential quantification of type 1 objects (sets of natural numbers) and arbitrary (universal and existential) quantification of type 0 objects. So, Σ^0_1 is part of the first level of the arithmetical hierarchy while Σ^1_1 is part of the first level of the analytical hierarchy. In general, level Σ^0_n (level Π^0_n) of the arithmetical hierarchy is induced by formulas with at most $n-1$ alternations between existential and universal type 0 quantifiers, starting with an existential (universal) quantifier. Similar hierarchies can be defined for arithmetic of any fixed order by limiting the alternations of the highest-order quantifiers and allowing arbitrary lower-order quantification. In this work, the highest order we are concerned with is three, i.e., quantification over sets of sets of natural numbers.

HyperLTL satisfiability is Σ^1_1-complete [19], HyperLTL finite-state satisfiability is Σ^0_1-complete [16, 20], and, as mentioned above, Hyper^{2}LTL model-checking is Σ^1_1-hard [4], but, prior to this current work, no upper bounds were known for Hyper^{2}LTL.

Another yardstick is truth for order k arithmetic, i.e., the question whether a given sentence of order k arithmetic evaluates to true. In the following, we are in particular interested in the case $k = 3$, i.e., we consider formulas with arbitrary quantification over type 0 objects, type 1 objects, and type 2 objects (sets of sets of natural numbers). Note that these formulas span the whole third hierarchy, as we allow arbitrary nesting of existential and universal third-order quantification.

Our Contributions. In this work, we determine the exact complexity of Hyper^{2}LTL satisfiability, finite-state satisfiability, and model-checking, for the full logic and the two fragments introduced by Beutner et al. [4], as well as for two variations of the semantics.

An important stepping stone for us is the investigation of the cardinality of models of Hyper^{2}LTL. It is known that every satisfiable HyperLTL sentence has a countable model, and that some have no finite models [18]. This restricts the order of arithmetic that can be simulated in HyperLTL and explains in particular the Σ^1_1-completeness of HyperLTL satisfiability [19]. We show that (unsurprisingly) second-order quantification allows to write formulas that only have uncountable models by generalizing the lower bound construction of HyperLTL to Hyper^{2}LTL. Note that the cardinality of the continuum is a trivial upper bound on the size of models, as they are sets of traces.

With this tool at hand, we are able to show that Hyper^{2}LTL satisfiability is equivalent to truth in third-order arithmetic, i.e., much harder than HyperLTL satisfiability. This increase in complexity is not surprising, as second-order quantification can be expected to increase the complexity considerably. But what might be surprising at first glance is that the problem is not Σ^2_1-complete, i.e., at the same position of the third hierarchy that HyperLTL satisfiability occupies in one full hierarchy below (see Figure 1). However, arbitrary second-order trace quantification corresponds to arbitrary quantification over type 2 objects, which allows to capture the full third hierarchy. Furthermore, we also show that Hyper^{2}LTL finite-state satisfiability is equivalent to truth in third-order arithmetic, and therefore as hard as general satisfiability. This should be contrasted with the situation for HyperLTL described above, where finite-state satisfiability is Σ^0_1-complete (i.e., recursively enumerable) and thus much simpler than general satisfiability, which is Σ^1_1-complete.

Finally, our techniques for Hyper^2LTL satisfiability also shed light on the exact complexity of Hyper^2LTL model-checking, which we show to be equivalent to truth in third-order arithmetic as well, i.e., all three problems we consider have the same complexity. In particular, this increases the lower bound on Hyper^2LTL model-checking from Σ_1^1 to truth in third-order arithmetic. Again, this has be contrasted with the situation for HyperLTL, where model-checking is decidable, albeit TOWER-complete [33, 31].

So, quantification over arbitrary sets of traces makes verification very hard. However, Beutner et al. [4] noticed that many of the applications of Hyper^2LTL described above do not require full second-order quantification, but can be expressed with restricted forms of second-order quantification. To capture this, they first restrict second-order quantification to smallest/largest sets satisfying a guard (obtaining the fragment $\text{Hyper}^2\text{LTL}_{\text{mm}}$)[1] and then further restrict those to least fixed points induced by HyperLTL definable operators (obtaining the fragment lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$). By construction, these least fixed points are unique, i.e., second-order quantification degenerates to least fixed point computation.

As an example, consider again φ_{ck} above. The internal constraint

$$\forall \pi' \in X. \forall \pi''. \left(\bigvee\nolimits_{i=1}^{n} \pi' \sim_i \pi'' \right) \rightarrow \pi'' \in X$$

defines a condition on what traces have to be in the set X, and how they are added gradually to X, a behavior that can be captured by a fixed point computation for the (monotone) operator induced by the formula above. Since the last part $\forall \pi' \in X.\ \varphi(\pi')$ of φ_{ck} universally quantifies over all traces in X, and since X is existentially quantified, it is enough to consider the minimal set that satisfies the internal constraint: if *some* set satisfies a universal condition, then so does the minimal set. This minimal set is exactly the least fixed point of the operator induced by the formula above. Similar behavior is exhibited by many other applications of the logic, which gives the motivation to explore the fragment lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$.

Nevertheless, we show that $\text{Hyper}^2\text{LTL}_{\text{mm}}$ retains the same complexity as Hyper^2LTL, i.e., all three problems are still equivalent to truth in third-order arithmetic: Just restricting to guarded second-order quantification does not decrease the complexity.

For all results mentioned so far, it is irrelevant whether we allow second-order quantifiers to range over sets of traces that may contain traces that are not in the model (standard semantics) or whether we restrict these quantifiers to subsets of the model (closed-world semantics). But if we consider lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ satisfiability under closed-world semantics, the complexity finally decreases to Σ_1^1-completeness. Stated differently, one can add least fixed points of HyperLTL definable operators to HyperLTL without increasing the complexity of the satisfiability problem. Finally, for lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ finite-state satisfiability and model-checking, we prove Σ_2^2-membership and Σ_1^1 lower bounds for both semantics, thereby confining the complexity to the second level of the third hierarchy.

Table 1 lists our results and compares them to LTL and HyperLTL. Recall that Beutner et al. showed that lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ yields (partial) model checking and monitoring algorithms [4, 5]. Our results confirm the usability of the lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ fragment also from a theoretical point of view, as all problems relevant for verification have significantly lower complexity (albeit, still highly undecidable).

Proofs omitted due to space restrictions can be found in the full version [21].

[1] In [4] this fragment is termed $\text{Hyper}^2\text{LTL}_{\text{fp}}$. For clarity, since it is not fixed point based, but uses minimality/maximality constraints, we use the subscript "mm" instead of "fp".

Table 1 List of our results (in bold) and comparison to related logics. "T3A-equivalent" stands for "equivalent to truth in third-order arithmetic". Entries marked with an asterisk only hold for closed-world semantics, all others hold for both semantics.

Logic	Satisfiability	Finite-state satisfiability	Model-checking
LTL	PSPACE-complete	PSPACE-complete	PSPACE-complete
HyperLTL	Σ^1_1-complete	Σ^0_1-complete	TOWER-complete
Hyper^{2}LTL	**T3A-equivalent**	**T3A-equivalent**	**T3A-equivalent**
Hyper^{2}LTL$_{\mathrm{mm}}$	**T3A-equivalent**	**T3A-equivalent**	**T3A-equivalent**
lfp-Hyper^{2}LTL$_{\mathrm{mm}}$	**Σ^1_1-complete***	**Σ^1_1-hard/in Σ^2_2**	**Σ^1_1-hard/in Σ^2_2**

2 Preliminaries

We denote the nonnegative integers by $\mathbb{N}$. An alphabet is a nonempty finite set. The set of infinite words over an alphabet Σ is denoted by Σ^ω. Let AP be a nonempty finite set of atomic propositions. A trace over AP is an infinite word over the alphabet 2^{AP}. Given a subset $\mathrm{AP}' \subseteq \mathrm{AP}$, the AP'-projection of a trace $t(0)t(1)t(2)\cdots$ over AP is the trace $(t(0) \cap \mathrm{AP}')(t(1) \cap \mathrm{AP}')(t(2) \cap \mathrm{AP}')\cdots$ over AP'.

A transition system $\mathfrak{T} = (V, E, I, \lambda)$ consists of a finite nonempty set V of vertices, a set $E \subseteq V \times V$ of (directed) edges, a set $I \subseteq V$ of initial vertices, and a labeling $\lambda\colon V \to 2^{\mathrm{AP}}$ of the vertices by sets of atomic propositions. We assume that every vertex has at least one outgoing edge. A path ρ through $\mathfrak{T}$ is an infinite sequence $\rho(0)\rho(1)\rho(2)\cdots$ of vertices with $\rho(0) \in I$ and $(\rho(n), \rho(n+1)) \in E$ for every $n \geq 0$. The trace of ρ is defined as $\lambda(\rho) = \lambda(\rho(0))\lambda(\rho(1))\lambda(\rho(2))\cdots$. The set of traces of $\mathfrak{T}$ is $\mathrm{Tr}(\mathfrak{T}) = \{\lambda(\rho) \mid \rho \text{ is a path through } \mathfrak{T}\}$.

Hyper^{2}LTL. Let $\mathcal{V}_1$ be a set of first-order trace variables (i.e., ranging over traces) and $\mathcal{V}_2$ be a set of second-order trace variables (i.e., ranging over sets of traces) such that $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$. We typically use π (possibly with decorations) to denote first-order variables and X, Y, Z (possibly with decorations) to denote second-order variables. Also, we assume the existence of two distinguished second-order variables $X_a, X_d \in \mathcal{V}_2$ such that X_a refers to the set $(2^{\mathrm{AP}})^\omega$ of all traces, and X_d refers to the universe of discourse (the set of traces the formula is evaluated over).

The formulas of Hyper^{2}LTL are given by the grammar

$$\varphi ::= \exists X.\ \varphi \mid \forall X.\ \varphi \mid \exists \pi \in X.\ \varphi \mid \forall \pi \in X.\ \varphi \mid \psi \qquad \psi ::= \mathtt{p}_\pi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

where $\mathtt{p}$ ranges over AP, π ranges over $\mathcal{V}_1$, X ranges over $\mathcal{V}_2$, and $\mathbf{X}$ (next) and $\mathbf{U}$ (until) are temporal operators. Conjunction ($\wedge$), exclusive disjunction ($\oplus$), implication ($\rightarrow$), and equivalence ($\leftrightarrow$) are defined as usual, and the temporal operators eventually ($\mathbf{F}$) and always ($\mathbf{G}$) are derived as $\mathbf{F}\,\psi = \neg\psi\,\mathbf{U}\,\psi$ and $\mathbf{G}\,\psi = \neg\,\mathbf{F}\,\neg\psi$. We measure the size of a formula by its number of distinct subformulas.

The semantics of Hyper^{2}LTL is defined with respect to a variable assignment, i.e., a partial mapping $\Pi\colon \mathcal{V}_1 \cup \mathcal{V}_2 \to (2^{\mathrm{AP}})^\omega \cup 2^{(2^{\mathrm{AP}})^\omega}$ such that

- if $\Pi(\pi)$ for $\pi \in \mathcal{V}_1$ is defined, then $\Pi(\pi) \in (2^{\mathrm{AP}})^\omega$ and
- if $\Pi(X)$ for $X \in \mathcal{V}_2$ is defined, then $\Pi(X) \in 2^{(2^{\mathrm{AP}})^\omega}$.

Given a variable assignment Π, a variable $\pi \in \mathcal{V}_1$, and a trace t, we denote by $\Pi[\pi \mapsto t]$ the assignment that coincides with Π on all variables but π, which is mapped to t. Similarly, for a variable $X \in \mathcal{V}_2$, and a set T of traces, $\Pi[X \mapsto T]$ is the assignment that coincides with Π everywhere but X, which is mapped to T. Furthermore, $\Pi[j, \infty)$ denotes the variable assignment mapping every $\pi \in \mathcal{V}_1$ in Π's domain to $\Pi(\pi)(j)\Pi(\pi)(j+1)\Pi(\pi)(j+2)\cdots$, the suffix of $\Pi(\pi)$ starting at position j (the assignment of variables $X \in \mathcal{V}_2$ is not updated).

For a variable assignment Π we define

- $\Pi \models \mathtt{p}_\pi$ if $\mathtt{p} \in \Pi(\pi)(0)$,
- $\Pi \models \neg\psi$ if $\Pi \not\models \psi$,
- $\Pi \models \psi_1 \vee \psi_2$ if $\Pi \models \psi_1$ or $\Pi \models \psi_2$,
- $\Pi \models \mathbf{X}\,\psi$ if $\Pi[1,\infty) \models \psi$,
- $\Pi \models \psi_1 \,\mathbf{U}\, \psi_2$ if there is a $j \geq 0$ such that $\Pi[j,\infty) \models \psi_2$ and for all $0 \leq j' < j$ we have $\Pi[j',\infty) \models \psi_1$,
- $\Pi \models \exists \pi \in X.\ \varphi$ if there exists a trace $t \in \Pi(X)$ such that $\Pi[\pi \mapsto t] \models \varphi$,
- $\Pi \models \forall \pi \in X.\ \varphi$ if for all traces $t \in \Pi(X)$ we have $\Pi[\pi \mapsto t] \models \varphi$,
- $\Pi \models \exists X.\ \varphi$ if there exists a set $T \subseteq (2^{\mathrm{AP}})^\omega$ such that $\Pi[X \mapsto T] \models \varphi$, and
- $\Pi \models \forall X.\ \varphi$ if for all sets $T \subseteq (2^{\mathrm{AP}})^\omega$ we have $\Pi[X \mapsto T] \models \varphi$.

Throughout the paper, we use the following shorthands to simplify our formulas:

- We write $\pi =_{\mathrm{AP}'} \pi'$ for a set $\mathrm{AP}' \subseteq \mathrm{AP}$ for the formula $\mathbf{G} \bigwedge_{\mathtt{p} \in \mathrm{AP}'} (\mathtt{p}_\pi \leftrightarrow \mathtt{p}_{\pi'})$ expressing that the AP'-projection of π and the AP'-projection of π' are equal.
- We write $\pi \triangleright X$ for the formula $\exists \pi' \in X.\ \pi =_{\mathrm{AP}} \pi'$ expressing that the trace π is in X. Note that this shorthand cannot be used under the scope of temporal operators, as we require formulas to be in prenex normal form.

A sentence is a formula in which only the variables X_a, X_d can be free. The variable assignment with empty domain is denoted by $\Pi_\emptyset$. We say that a set T of traces satisfies a Hyper^{2}LTL sentence φ, written $T \models \varphi$, if $\Pi_\emptyset[X_a \mapsto (2^{\mathrm{AP}})^\omega, X_d \mapsto T] \models \varphi$, i.e., if we assign the set of all traces to X_a and the set T to the universe of discourse X_d. In this case, we say that T is a model of φ. A transition system $\mathfrak{T}$ satisfies φ, written $\mathfrak{T} \models \varphi$, if $\mathrm{Tr}(\mathfrak{T}) \models \varphi$.

Although Hyper^{2}LTL sentences are required to be in prenex normal form, Hyper^{2}LTL sentences are closed under Boolean combinations, which can easily be seen by transforming such a sentence into an equivalent one in prenex normal form (which might require renaming of variables). Thus, in examples and proofs we will often use Boolean combinations of Hyper^{2}LTL sentences.

▶ Remark 1. HyperLTL is the fragment of Hyper^{2}LTL obtained by disallowing second-order quantification and only allowing first-order quantification of the form $\exists \pi \in X_d$ and $\forall \pi \in X_d$, i.e., one can only quantify over traces from the universe of discourse. Hence, we typically simplify our notation to $\exists \pi$ and $\forall \pi$ in HyperLTL formulas.

Closed-World Semantics. Second-order quantification in Hyper^{2}LTL as defined by Beutner et al. [4] (and introduced above) ranges over arbitrary sets of traces (not necessarily from the universe of discourse) and first-order quantification ranges over elements in such sets, i.e., (possibly) again over arbitrary traces. To disallow this, we introduce *closed-world* semantics for Hyper^{2}LTL, only considering formulas that do not use the variable X_a. We change the semantics of set quantifiers as follows, where the closed-world semantics of atomic propositions, Boolean connectives, temporal operators, and trace quantifiers is defined as before:

- $\Pi \models_{\mathrm{cw}} \exists X.\ \varphi$ if there exists a set $T \subseteq \Pi(X_d)$ such that $\Pi[X \mapsto T] \models \varphi$, and
- $\Pi \models_{\mathrm{cw}} \forall X.\ \varphi$ if for all sets $T \subseteq \Pi(X_d)$ we have $\Pi[X \mapsto T] \models \varphi$.

We say that $T \subseteq (2^{\mathrm{AP}})^\omega$ satisfies φ under closed-world semantics, if $\Pi_\emptyset[X_d \mapsto T] \models_{\mathrm{cw}} \varphi$. Hence, under closed-world semantics, second-order quantifiers only range over subsets of the universe of discourse. Consequently, first-order quantifiers also range over traces from the universe of discourse.

▶ **Lemma 2.** *Every* Hyper^{2}LTL *sentence φ can be translated in polynomial time (in $|\varphi|$) into a* Hyper^{2}LTL *sentence φ' such that for all sets T of traces we have that $T \models_{\mathrm{cw}} \varphi$ if and only if $T \models \varphi'$ (under standard semantics).*

Thus, all complexity upper bounds we derive for standard semantics also hold for closed-world semantics and all lower bounds for closed-world semantics hold for standard semantics.

▶ Remark 3. Let φ be an X_a-free Hyper^{2}LTL sentence over AP. We have $(2^{\mathrm{AP}})^\omega \models \varphi$ (under standard semantics) if and only if $(2^{\mathrm{AP}})^\omega \models_{\mathrm{cw}} \varphi$, as the second-order quantifiers range in both cases over subsets of $(2^{\mathrm{AP}})^\omega$, which implies that the trace quantifiers in both cases range over traces from $(2^{\mathrm{AP}})^\omega$.

Arithmetic. To capture the complexity of undecidable problems, we consider formulas of arithmetic, i.e., predicate logic with signature $(+, \cdot, <, \in)$, evaluated over the structure $(\mathbb{N}, +, \cdot, <, \in)$. A type 0 object is a natural number in $\mathbb{N}$, a type 1 object is a subset of $\mathbb{N}$, and a type 2 object is a set of subsets of $\mathbb{N}$.

Our benchmark is third-order arithmetic, i.e., predicate logic with quantification over type 0, type 1, and type 2 objects. In the following, we use lower-case roman letters (possibly with decorations) for first-order variables, upper-case roman letters (possibly with decorations) for second-order variables, and upper-case calligraphic roman letters (possibly with decorations) for third-order variables. Note that every fixed natural number is definable in first-order arithmetic, so we freely use them as syntactic sugar. Truth of third-order arithmetic is the following problem: given a sentence φ of third-order arithmetic, does $(\mathbb{N}, +, \cdot, <, \in)$ satisfy φ?

Arithmetic formulas with a single free first-order variable define sets of natural numbers. We are interested in the classes

- Σ^1_1 containing sets of the form $\{x \in \mathbb{N} \mid \exists X_1 \subseteq \mathbb{N}.\ \cdots \exists X_k \subseteq \mathbb{N}.\ \psi(x, X_1, \ldots, X_k)\}$, where ψ is a formula of arithmetic with arbitrary quantification over type 0 objects (but no second-order quantifiers), and
- Σ^2_2 containing sets of the following form, where ψ is a formula of arithmetic with arbitrary quantification over type 0 and type 1 objects (but no third-order quantifiers): $\{x \in \mathbb{N} \mid \exists \mathcal{X}_1 \subseteq 2^{\mathbb{N}}.\ \cdots \exists \mathcal{X}_k \subseteq 2^{\mathbb{N}}. \forall \mathcal{Y}_1 \subseteq 2^{\mathbb{N}}.\ \cdots \forall \mathcal{Y}_{k'} \subseteq 2^{\mathbb{N}}.\ \psi(x, \mathcal{X}_1, \ldots, \mathcal{X}_k, \mathcal{Y}_1, \ldots, \mathcal{Y}_{k'})\}$.

3 The Cardinality of Hyper^{2}LTL Models

In this section, we investigate the cardinality of models of satisfiable Hyper^{2}LTL sentences, i.e., the number of traces in the model.

We begin by stating a (trivial) upper bound, which follows from the fact that models are sets of traces. Here, $\mathfrak{c}$ denotes the cardinality of the continuum (equivalently, the cardinality of $2^{\mathbb{N}}$ and of $(2^{\mathrm{AP}})^\omega$ for any finite nonempty AP).

▶ **Proposition 4.** *Every satisfiable* Hyper^{2}LTL *sentence has a model of cardinality at most $\mathfrak{c}$.*

In this section, we show that this trivial upper bound is tight.

▶ Remark 5. There is a very simple, albeit equally unsatisfactory way to obtain the desired lower bound: Consider $\forall \pi \in X_a.\ \pi \triangleright X_d$ expressing that every trace in the set of all traces is also in the universe of discourse, i.e., $(2^{\mathrm{AP}})^\omega$ is its only model over AP. However, this crucially relies on the fact that X_a is, by definition, interpreted as the set of all traces. In fact, the formula does not even use second-order quantification.

We show how to construct a sentence that has only uncountable models, and which retains that property under closed-world semantics (which in particular means it cannot use X_a). This should be compared with HyperLTL, where every satisfiable sentence has a countable model [18]: Unsurprisingly, the addition of (even closed-world) second-order quantification increases the cardinality of minimal models, even without cheating.

▶ **Example 6.** We begin by recalling a construction of Finkbeiner and Zimmermann giving a satisfiable HyperLTL sentence ψ that has no finite models [18]. The sentence intuitively posits the existence of a unique trace for every natural number n. Our lower bound for Hyper^{2}LTL builds upon that construction.

Fix AP $= \{\mathtt{x}\}$ and consider the conjunction $\psi = \psi_1 \wedge \psi_2 \wedge \psi_3$ of the following three formulas:

1. $\psi_1 = \forall\pi.\ \neg\mathtt{x}_\pi\,\mathbf{U}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg\mathtt{x}_\pi)$: every trace in a model is of the form $\emptyset^n\{\mathtt{x}\}\emptyset^\omega$ for some $n \in \mathbb{N}$, i.e., every model is a subset of $\{\emptyset^n\{\mathtt{x}\}\emptyset^\omega \mid n \in \mathbb{N}\}$.
2. $\psi_2 = \exists\pi.\ \mathtt{x}_\pi$: the trace $\emptyset^0\{\mathtt{x}\}\emptyset^\omega$ is in every model.
3. $\psi_3 = \forall\pi.\ \exists\pi'.\ \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathtt{x}_{\pi'})$: if $\emptyset^n\{\mathtt{x}\}\emptyset^\omega$ is in a model for some $n \in \mathbb{N}$, then also $\emptyset^{n+1}\{\mathtt{x}\}\emptyset^\omega$.

Then, ψ has exactly one model (over AP), namely $\{\emptyset^n\{\mathtt{x}\}\emptyset^\omega \mid n \in \mathbb{N}\}$.

A trace of the form $\emptyset^n\{\mathtt{x}\}\emptyset^\omega$ encodes the natural number n and ψ expresses that every model contains the encodings of all natural numbers and nothing else. But we can of course also encode sets of natural numbers with traces as follows: a trace t over a set of atomic propositions containing $\mathtt{x}$ encodes the set $\{n \in \mathbb{N} \mid \mathtt{x} \in t(n)\}$. In the following, we show that second-order quantification allows us to express the existence of the encodings of all subsets of natural numbers by requiring that for every subset $S \subseteq \mathbb{N}$ (quantified as the set $\{\emptyset^n\{\mathtt{x}\}\emptyset^\omega \mid n \in S\}$ of traces) there is a trace t encoding S, which means $\mathtt{x}$ is in $t(n)$ if and only if S contains a trace in which $\mathtt{x}$ holds at position n. This equivalence can be expressed in Hyper^{2}LTL. For technical reasons, we do not capture the equivalence directly but instead use encodings of both the natural numbers that are in S and the natural numbers that are not in S.

▶ **Theorem 7.** *There is a satisfiable X_a-free* Hyper^{2}LTL *sentence that only has models of cardinality $\mathfrak{c}$ (both under standard and closed-world semantics).*

Proof. We first prove that there is a satisfiable X_a-free Hyper^{2}LTL sentence $\varphi_{allSets}$ whose unique model (under standard semantics) has cardinality $\mathfrak{c}$. To this end, we fix AP $= \{\mathtt{+}, \mathtt{-}, \mathtt{s}, \mathtt{x}\}$ and consider the conjunction $\varphi_{allSets} = \varphi_0 \wedge \cdots \wedge \varphi_4$ of the following formulas:

- $\varphi_0 = \forall\pi \in X_d.\ \bigvee_{\mathtt{p}\in\{\mathtt{+},\mathtt{-},\mathtt{s}\}} \mathbf{G}(\mathtt{p}_\pi \wedge \bigwedge_{\mathtt{p}'\in\{\mathtt{+},\mathtt{-},\mathtt{s}\}\setminus\{\mathtt{p}\}} \neg\mathtt{p}'_\pi)$: In each trace of a model, one of the propositions in $\{\mathtt{+}, \mathtt{-}, \mathtt{s}\}$ holds at every position and the other two propositions in $\{\mathtt{+}, \mathtt{-}, \mathtt{s}\}$ hold at none of the positions. Consequently, we speak in the following about type $\mathtt{p}$ traces for $\mathtt{p} \in \{\mathtt{+}, \mathtt{-}, \mathtt{s}\}$.
- $\varphi_1 = \forall\pi \in X_d.\ (\mathtt{+}_\pi \vee \mathtt{-}_\pi) \rightarrow \neg\mathtt{x}_\pi\,\mathbf{U}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg\mathtt{x}_\pi)$: Type $\mathtt{p}$ traces for $\mathtt{p} \in \{\mathtt{+}, \mathtt{-}\}$ in the model have the form $\{\mathtt{p}\}^n\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$ for some $n \in \mathbb{N}$.
- $\varphi_2 = \bigwedge_{\mathtt{p}\in\{\mathtt{+},\mathtt{-}\}} \exists\pi \in X_d.\ \mathtt{p}_\pi \wedge \mathtt{x}_\pi$: for both $\mathtt{p} \in \{\mathtt{+}, \mathtt{-}\}$, the type $\mathtt{p}$ trace $\{\mathtt{p}\}^0\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$ is in every model.
- $\varphi_3 = \bigwedge_{\mathtt{p}\in\{\mathtt{+},\mathtt{-}\}} \forall\pi \in X_d.\ \exists\pi' \in X_d.\ \mathtt{p}_\pi \rightarrow (\mathtt{p}_{\pi'} \wedge \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathtt{x}_{\pi'}))$: for both $\mathtt{p} \in \{\mathtt{+}, \mathtt{-}\}$, if the type $\mathtt{p}$ trace $\{\mathtt{p}\}^n\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$ is in a model for some $n \in \mathbb{N}$, then also $\{\mathtt{p}\}^{n+1}\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$.

The formulas $\varphi_1, \varphi_2, \varphi_3$ are similar to the formulas ψ_1, ψ_2, ψ_3 from Example 6. So, every model of $\varphi_0 \wedge \cdots \wedge \varphi_3$ contains $\{\{\mathtt{+}\}^n\{\mathtt{x}, \mathtt{+}\}\{\mathtt{+}\}^\omega \mid n \in \mathbb{N}\}$ and $\{\{\mathtt{-}\}^n\{\mathtt{x}, \mathtt{-}\}\{\mathtt{-}\}^\omega \mid n \in \mathbb{N}\}$ as subsets, and no other type $\mathtt{+}$ or type $\mathtt{-}$ traces.

Now, consider a set T of traces over AP (recall that second-order quantification ranges over arbitrary sets, not only over subsets of the universe of discourse). We say that T is contradiction-free if there is no $n \in \mathbb{N}$ such that $\{\texttt{+}\}^n\{\texttt{x},\texttt{+}\}\{\texttt{+}\}^\omega \in T$ and $\{\texttt{-}\}^n\{\texttt{x},\texttt{-}\}\{\texttt{-}\}^\omega \in T$. Furthermore, a trace t over AP is consistent with a contradiction-free T if

(C1) $\{\texttt{+}\}^n\{\texttt{x},\texttt{+}\}\{\texttt{+}\}^\omega \in T$ implies $\texttt{x} \in t(n)$ and

(C2) $\{\texttt{-}\}^n\{\texttt{x},\texttt{-}\}\{\texttt{-}\}^\omega \in T$ implies $\texttt{x} \notin t(n)$.

Note that T does not necessarily specify the truth value of $\texttt{x}$ in every position of t, i.e., in those positions $n \in \mathbb{N}$ where neither $\{\texttt{+}\}^n\{\texttt{x},\texttt{+}\}\{\texttt{+}\}^\omega$ nor $\{\texttt{-}\}^n\{\texttt{x},\texttt{-}\}\{\texttt{-}\}^\omega$ are in T. Nevertheless, for every trace t over $\{\texttt{x}\}$ there is a contradiction-free T such that the $\{\texttt{x}\}$-projection of every trace t' over AP that is consistent with T is equal to t. Thus, each of the uncountably many traces over $\{\texttt{x}\}$ is induced by some subset of the model.

- Hence, we define φ_4 as the formula

$$\forall X.\ \overbrace{[\forall\pi \in X.\ \forall\pi' \in X.\ (\texttt{+}_\pi \wedge \texttt{-}_{\pi'}) \rightarrow \neg\mathbf{F}(\texttt{x}_\pi \wedge \texttt{x}_{\pi'})]}^{X \text{ is contradiction-free}} \rightarrow$$
$$\exists\pi'' \in X_d.\ \forall\pi''' \in X.\ \texttt{s}_{\pi''} \wedge \underbrace{(\texttt{+}_{\pi'''} \rightarrow \mathbf{F}(\texttt{x}_{\pi'''} \wedge \texttt{x}_{\pi''}))}_{\text{(C1)}} \wedge \underbrace{(\texttt{-}_{\pi'''} \rightarrow \mathbf{F}(\texttt{x}_{\pi'''} \wedge \neg\texttt{x}_{\pi''}))}_{\text{(C2)}},$$

  expressing that for every contradiction-free set of traces T, there is a type $\texttt{s}$ trace t'' in the model (note that π'' is required to be in X_d) that is consistent with T.

While $\varphi_{allSets}$ is not in prenex normal form, it can easily be turned into an equivalent formula in prenex normal form (at the cost of readability).

Now, the set

$$T_{allSets} = \{\{\texttt{+}\}^n\{\texttt{x},\texttt{+}\}\{\texttt{+}\}^\omega \mid n \in \mathbb{N}\} \cup \{\{\texttt{-}\}^n\{\texttt{x},\texttt{-}\}\{\texttt{-}\}^\omega \mid n \in \mathbb{N}\} \cup$$
$$\{(t(0) \cup \{\texttt{s}\})(t(1) \cup \{\texttt{s}\})(t(2) \cup \{\texttt{s}\}) \cdots \mid t \in (2^{\{\texttt{x}\}})^\omega\}$$

of traces satisfies $\varphi_{allSets}$. On the other hand, every model of $\varphi_{allSets}$ must indeed contain $T_{allSets}$ as a subset, as $\varphi_{allSets}$ requires the existence of all of its traces in the model. Finally, due to φ_0 and φ_1, a model (over AP) cannot contain any traces that are not in $T_{allSets}$, i.e., $T_{allSets}$ is the unique model of $\varphi_{allSets}$.

To conclude, we just remark that

$$\{(t(0) \cup \{\texttt{s}\})(t(1) \cup \{\texttt{s}\})(t(2) \cup \{\texttt{s}\}) \cdots \mid t \in (2^{\{\texttt{x}\}})^\omega\} \subseteq T_{allSets}$$

has indeed cardinality $\mathfrak{c}$, as $(2^{\{\texttt{x}\}})^\omega$ has cardinality $\mathfrak{c}$.

Finally, let us consider closed-world semantics. We can restrict the second-order quantifier in φ_4 (the only one in $\varphi_{allSets}$) to subsets of the universe of discourse, as the set $T = \{\{\texttt{+}\}^n\{\texttt{x},\texttt{+}\}\{\texttt{+}\}^\omega \mid n \in \mathbb{N}\} \cup \{\{\texttt{-}\}^n\{\texttt{x},\texttt{-}\}\{\texttt{-}\}^\omega \mid n \in \mathbb{N}\}$ of traces (which is a subset of every model) is already *rich* enough to encode every subset of $\mathbb{N}$ by an appropriate contradiction-free subset of T. Thus, $\varphi_{allSets}$ has the unique model $T_{allSets}$ even under closed-world semantics. ◀

4 The Complexity of Hyper^{2}LTL Satisfiability

A Hyper^{2}LTL sentence is satisfiable if it has a model. The Hyper^{2}LTL satisfiability problem asks, given a Hyper^{2}LTL sentence φ, whether φ is satisfiable. In this section, we determine tight bounds on the complexity of Hyper^{2}LTL satisfiability and some of its variants.

Recall that in Section 3, we encoded sets of natural numbers as traces over a set of propositions containing $\texttt{x}$ and encoded natural numbers as singleton sets. The proof of

Theorem 7 relies on constructing a sentence that requires each of its models to encode every subset of $\mathbb{N}$ by a trace in the model. Hence, sets of traces can encode sets of sets of natural numbers, i.e., type 2 objects.

Another important ingredient in the following proof is the implementation of addition and multiplication in HyperLTL. Let $\mathrm{AP}_{arith} = \{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$ and let $T_{(+,\cdot)}$ be the set of all traces $t \in (2^{\mathrm{AP}_{arith}})^\omega$ such that:

- there are unique $n_1, n_2, n_3 \in \mathbb{N}$ with $\mathtt{arg1} \in t(n_1)$, $\mathtt{arg2} \in t(n_2)$, and $\mathtt{res} \in t(n_3)$, and
- either $\mathtt{add} \in t(n)$ and $\mathtt{mult} \notin t(n)$ for all n, and $n_1 + n_2 = n_3$, or $\mathtt{mult} \in t(n)$ and $\mathtt{add} \notin t(n)$ for all n, and $n_1 \cdot n_2 = n_3$.

▶ **Proposition 8** (Theorem 5.5 of [20]). *There is a satisfiable* HyperLTL *sentence $\varphi_{(+,\cdot)}$ such that the AP_{arith}-projection of every model of $\varphi_{(+,\cdot)}$ is $T_{(+,\cdot)}$.*

Combining the capability of quantifying over type 0, type 1, and type 2 objects and the encoding of addition and multiplication, we show that Hyper^{2}LTL and truth in third-order arithmetic have the same complexity.

▶ **Theorem 9.** *The* Hyper^{2}LTL *satisfiability problem is polynomial-time equivalent to truth in third-order arithmetic. The lower bound holds even for X_a-free sentences.*

Proof. We begin with the lower bound by reducing truth in third-order arithmetic to Hyper^{2}LTL satisfiability: we present a polynomial-time translation from sentences φ of third-order arithmetic to Hyper^{2}LTL sentences φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if φ' is satisfiable.

Given a third-order sentence φ, we define

$$\varphi' = \exists X_{allSets}.\ \exists X_{arith}.\ (\varphi_{allSets}[X_d/X_{allSets}] \wedge \varphi'_{(+,\cdot)} \wedge hyp(\varphi))$$

where

- $\varphi_{allSets}[X_d/X_{allSets}]$ is the Hyper^{2}LTL sentence from the proof of Theorem 7 where every occurrence of X_d is replaced by $X_{allSets}$ and thus enforces every subset of $\mathbb{N}$ to be encoded in the interpretation of $X_{allSets}$ (as introduced in the proof of Theorem 7),
- $\varphi'_{(+,\cdot)}$ is the Hyper^{2}LTL formula obtained from the HyperLTL formula $\varphi_{(+,\cdot)}$ by replacing each quantifier $\exists\pi$ ($\forall\pi$, respectively) by $\exists\pi \in X_{arith}$ ($\forall\pi \in X_{arith}$, respectively) and thus enforces that X_{arith} is interpreted by a set whose AP_{arith}-projection is $T_{(+,\cdot)}$, and

where $hyp(\varphi)$ is defined inductively as follows:

- For third-order variables $\mathcal{Y}$,

$$hyp(\exists\mathcal{Y}.\ \psi) = \exists X_{\mathcal{Y}}.\ (\forall\pi \in X_{\mathcal{Y}}.\ \exists\pi' \in X_{allSets}.\ (\pi =_{\{\mathtt{+},\mathtt{-},\mathtt{s},\mathtt{x}\}} \pi') \wedge \mathtt{s}_\pi) \wedge hyp(\psi).$$

- For third-order variables $\mathcal{Y}$,

$$hyp(\forall\mathcal{Y}.\ \psi) = \forall X_{\mathcal{Y}}.\ (\forall\pi \in X_{\mathcal{Y}}.\ \exists\pi' \in X_{allSets}.\ (\pi =_{\{\mathtt{+},\mathtt{-},\mathtt{s},\mathtt{x}\}} \pi') \wedge \mathtt{s}_\pi) \rightarrow hyp(\psi).$$

- For second-order variables Y, $hyp(\exists Y.\ \psi) = \exists\pi_Y \in X_{allSets}.\ \mathtt{s}_{\pi_Y} \wedge hyp(\psi)$.
- For second-order variables Y, $hyp(\forall Y.\ \psi) = \forall\pi_Y \in X_{allSets}.\ \mathtt{s}_{\pi_Y} \rightarrow hyp(\psi)$.
- For first-order variables y,

$$hyp(\exists y.\ \psi) = \exists\pi_y \in X_{allSets}.\ \mathtt{s}_{\pi_y} \wedge [(\neg\mathtt{x}_{\pi_y})\,\mathbf{U}(\mathtt{x}_{\pi_y} \wedge \mathbf{X}\,\mathbf{G}\,\neg\mathtt{x}_{\pi_y})] \wedge hyp(\psi).$$

- For first-order variables y,

$$hyp(\forall y.\ \psi) = \forall\pi_y \in X_{allSets}.\ (\mathtt{s}_{\pi_y} \wedge [(\neg\mathtt{x}_{\pi_y})\,\mathbf{U}(\mathtt{x}_{\pi_y} \wedge \mathbf{X}\,\mathbf{G}\,\neg\mathtt{x}_{\pi_y})]) \rightarrow hyp(\psi).$$

- $hyp(\psi_1 \vee \psi_2) = hyp(\psi_1) \vee hyp(\psi_2)$.
- $hyp(\neg\psi) = \neg hyp(\psi)$.
- For second-order variables Y and third-order variables $\mathcal{Y}$,

  $$hyp(Y \in \mathcal{Y}) = \exists \pi \in X_{\mathcal{Y}}.\ \pi_Y =_{\{\mathtt{x}\}} \pi.$$

- For first-order variables y and second-order variables Y, $hyp(y \in Y) = \mathbf{F}(\mathtt{x}_{\pi_y} \wedge \mathtt{x}_{\pi_Y})$.
- For first-order variables y, y', $hyp(y < y') = \mathbf{F}(\mathtt{x}_{\pi_y} \wedge \mathbf{X}\,\mathbf{F}\,\mathtt{x}_{\pi_{y'}})$.
- For first-order variables y_1, y_2, y,

  $$hyp(y_1 + y_2 = y) = \exists \pi \in X_{arith}.\ \mathtt{add}_\pi \wedge \mathbf{F}(\mathtt{arg1}_\pi \wedge \mathtt{x}_{\pi_{y_1}}) \wedge \mathbf{F}(\mathtt{arg2}_\pi \wedge \mathtt{x}_{\pi_{y_2}}) \wedge \mathbf{F}(\mathtt{res}_\pi \wedge \mathtt{x}_{\pi_y}).$$

- For first-order variables y_1, y_2, y,

  $$hyp(y_1 \cdot y_2 = y) = \exists \pi \in X_{arith}.\ \mathtt{mult}_\pi \wedge \mathbf{F}(\mathtt{arg1}_\pi \wedge \mathtt{x}_{\pi_{y_1}}) \wedge \mathbf{F}(\mathtt{arg2}_\pi \wedge \mathtt{x}_{\pi_{y_2}}) \wedge \mathbf{F}(\mathtt{res}_\pi \wedge \mathtt{x}_{\pi_y}).$$

While φ' is not in prenex normal form, it can easily be brought into prenex normal form, as there are no quantifiers under the scope of a temporal operator.

As we are evaluating φ' w.r.t. standard semantics and the variable X_d (interpreted with the model) does not occur in φ', satisfaction of φ' is independent of the model, i.e., for all sets T, T' of traces, $T \models \varphi'$ if and only if $T' \models \varphi'$. So, let us fix some set T of traces. An induction shows that $(\mathbb{N}, +, \cdot, <, \in)$ satisfies φ if and only if T satisfies φ'. Altogether we obtain the desired equivalence between $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ and φ' being satisfiable.

For the upper bound, we conversely reduce Hyper^{2}LTL satisfiability to truth in third-order arithmetic: we present a polynomial-time translation from Hyper^{2}LTL sentences φ to sentences φ' of third-order arithmetic such that φ is satisfiable if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \varphi'$. Here, we assume AP to be fixed, so that we can use $|\mathrm{AP}|$ as a constant in our formulas (which is definable in arithmetic).

Let $pair\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ denote Cantor's pairing function defined as $pair(i, j) = \frac{1}{2}(i+j)(i+j+1)+j$, which is a bijection. Furthermore, fix some bijection $e\colon \mathrm{AP} \to \{0, 1, \ldots, |\mathrm{AP}|-1\}$. Then, we encode a trace $t \in (2^{\mathrm{AP}})^\omega$ by the set $S_t = \{pair(j, e(\mathtt{p})) \mid j \in \mathbb{N} \text{ and } \mathtt{p} \in t(j)\} \subseteq \mathbb{N}$. As $pair$ is a bijection, we have that $t \neq t'$ implies $S_t \neq S_{t'}$. While not every subset of $\mathbb{N}$ encodes some trace t, the first-order formula

$$\varphi_{isTrace}(Y) = \forall x.\ \forall y.\ y \geq |\mathrm{AP}| \to pair(x, y) \notin Y$$

checks if a set does encode a trace. Here, we use $pair$ as syntactic sugar, which is possible as the definition of $pair$ only uses addition and multiplication.

As (certain) sets of natural numbers encode traces, sets of (certain) sets of natural numbers encode sets of traces. This is sufficient to reduce Hyper^{2}LTL to third-order arithmetic, which allows the quantification over sets of sets of natural numbers. Before we present the translation, we need to introduce some more auxiliary formulas:

- Let $\mathcal{Y}$ be a third-order variable (i.e., $\mathcal{Y}$ ranges over sets of sets of natural numbers). Then, the formula

  $$\varphi_{onlyTraces}(\mathcal{Y}) = \forall Y.\ Y \in \mathcal{Y} \to \varphi_{isTrace}(Y)$$

  checks if a set of sets of natural numbers only contains sets encoding a trace.
- Further, the formula

  $$\varphi_{allTraces}(\mathcal{Y}) = \varphi_{onlyTraces}(\mathcal{Y}) \wedge \forall Y.\ \varphi_{isTrace}(Y) \to Y \in \mathcal{Y}$$

  checks if a set of sets of natural numbers contains exactly the sets encoding a trace.

Now, we are ready to define our encoding of Hyper^{2}LTL in third-order arithmetic. Given a Hyper^{2}LTL sentence φ, let

$$\varphi' = \exists \mathcal{Y}_a.\ \exists \mathcal{Y}_d.\ \varphi_{allTraces}(\mathcal{Y}_a) \wedge \varphi_{onlyTraces}(\mathcal{Y}_d) \wedge (ar(\varphi))(0)$$

where $ar(\varphi)$ is defined inductively as presented below. Note that φ' requires $\mathcal{Y}_a$ to contain exactly the encodings of all traces (i.e., it corresponds to the distinguished Hyper^{2}LTL variable X_a in the following translation) and $\mathcal{Y}_d$ is an existentially quantified set of trace encodings (i.e., it corresponds to the distinguished Hyper^{2}LTL variable X_d in the following translation).

In the inductive definition of $ar(\varphi)$, we will employ a free first-order variable i to denote the position at which the formula is to be evaluated to capture the semantics of the temporal operators. As seen above, in φ', this free variable is set to zero in correspondence with the Hyper^{2}LTL semantics.

- $ar(\exists X.\ \psi) = \exists \mathcal{Y}_X.\ \varphi_{onlyTraces}(\mathcal{Y}_X) \wedge ar(\psi)$. Here, the free variable of $ar(\exists X.\ \psi)$ is the free variable of $ar(\psi)$.
- $ar(\forall X.\ \psi) = \forall \mathcal{Y}_X.\ \varphi_{onlyTraces}(\mathcal{Y}_X) \rightarrow ar(\psi)$. Here, the free variable of $ar(\forall X.\ \psi)$ is the free variable of $ar(\psi)$.
- $ar(\exists \pi \in X.\ \psi) = \exists Y_\pi.\ Y_\pi \in \mathcal{Y}_X \wedge ar(\psi)$. Here, the free variable of $ar(\exists \pi \in X.\ \psi)$ is the free variable of $ar(\psi)$.
- $ar(\forall \pi \in X.\ \psi) = \forall Y_\pi.\ Y_\pi \in \mathcal{Y}_X \rightarrow ar(\psi)$. Here, the free variable of $ar(\forall \pi \in X.\ \psi)$ is the free variable of $ar(\psi)$.
- $ar(\psi_1 \vee \psi_2) = ar(\psi_1) \vee ar(\psi_2)$. Here, we require that the free variables of $ar(\psi_1)$ and $ar(\psi_2)$ are the same (which can always be achieved by variable renaming), which is then also the free variable of $ar(\psi_1 \vee \psi_2)$.
- $ar(\neg\psi) = \neg ar(\psi)$. Here, the free variable of $ar(\neg\psi)$ is the free variable of $\neg ar(\psi)$.
- $ar(\mathbf{X}\,\psi) = \exists i'(i' = i+1) \wedge ar(\psi)$, where i' is the free variable of $ar(\psi)$ and i is the free variable of $ar(\mathbf{X}\,\psi)$.
- $ar(\psi_1\,\mathbf{U}\,\psi_2) = \exists i_2.\ i_2 \geq i \wedge ar(\psi_2) \wedge \forall i_1.\ (i \leq i_1 \wedge i_1 < i_2) \rightarrow ar(\psi_1)$, where i_j is the free variable of $ar(\psi_j)$, and i is the free variable of $ar(\psi_1\,\mathbf{U}\,\psi_2)$.
- $ar(\mathtt{p}_\pi) = pair(i, e(\mathtt{p})) \in Y_\pi$, i.e., i is the free variable of $ar(\mathtt{p}_\pi)$.

Now, an induction shows that $\Pi_\emptyset[X_a \rightarrow (2^{\mathrm{AP}})^\omega, X_d \mapsto T] \models \varphi$ if and only if $(\mathbb{N}, +, \cdot, <, \in)$ satisfies $(ar(\varphi))(0)$ when the variable $\mathcal{Y}_a$ is interpreted by the encoding of $(2^{\mathrm{AP}})^\omega$ and $\mathcal{Y}_d$ is interpreted by the encoding of T. Hence, φ is indeed satisfiable if and only if $(\mathbb{N}, +, \cdot, <, \in)$ satisfies φ'. ◀

In the lower bound proof above, we have turned a sentence φ of third-order arithmetic into a Hyper^{2}LTL sentence φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if φ' is satisfiable. In fact, we have constructed φ' such that if it is satisfiable, then every set of traces satisfies it, in particular $(2^{\mathrm{AP}})^\omega$. Recall that Remark 3 states that $(2^{\mathrm{AP}})^\omega$ satisfies φ' under standard semantics if and only if $(2^{\mathrm{AP}})^\omega$ satisfies φ' under closed-world semantics. Thus, altogether we obtain that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if φ' is satisfiable under closed-world semantics, i.e, the lower bound holds even under closed-world semantics. Together with Lemma 2, this settles the complexity of Hyper^{2}LTL satisfiability under closed-world semantics.

▶ **Corollary 10.** *The* Hyper^{2}LTL *satisfiability problem under closed-world semantics is polynomial-time equivalent to truth in third-order arithmetic.*

The Hyper^{2}LTL finite-state satisfiability problem asks, given a Hyper^{2}LTL sentence φ, whether there is a finite transition system satisfying φ. Note that we do not ask for a finite set T of traces satisfying φ. In fact, the set of traces of the finite transition system may still be infinite or even uncountable. Nevertheless, the problem is potentially simpler, as there are only countably many finite transition systems (and their sets of traces are much simpler). However, we show that the finite-state satisfiability problem is as hard as the general satisfiability problem, as Hyper^{2}LTL allows the quantification over arbitrary (sets of) traces, i.e., restricting the universe of discourse to the traces of a finite transition system does not restrict second-order quantification at all (as the set of all traces is represented by a finite transition system). This has to be contrasted with the finite-state satisfiability problem for HyperLTL (defined analogously), which is Σ^0_1-complete (a.k.a. recursively enumerable), as HyperLTL model-checking of finite transition systems is decidable [11].

▶ **Theorem 11.** *The* Hyper^{2}LTL *finite-state satisfiability problem is polynomial-time equivalent to truth in third-order arithmetic. The lower bound holds even for X_a-free sentences.*

Proof. For the lower bound under standard semantics, we reduce truth in third-order arithmetic to Hyper^{2}LTL finite-state satisfiability: we present a polynomial-time translation from sentences φ of third-order arithmetic to Hyper^{2}LTL sentences φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if φ' is satisfied by a finite transition system.

So, let φ be a sentence of third-order arithmetic. Recall that in the proof of Theorem 9, we have shown how to construct from φ the Hyper^{2}LTL sentence φ' such that the following three statements are equivalent:

- $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$.
- φ' is satisfiable.
- φ' is satisfied all sets T of traces (and in particular by some finite-state transition system).

Thus, the lower bound follows from Theorem 9.

For the upper bound, we conversely reduce Hyper^{2}LTL finite-state satisfiability to truth in third-order arithmetic: we present a polynomial-time translation from Hyper^{2}LTL sentences φ to sentences φ'' of third-order arithmetic such that φ is satisfied by a finite transition system if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \varphi''$.

Recall that in the proof of Theorem 9, we have constructed a sentence

$$\varphi' = \exists \mathcal{Y}_a.\ \exists \mathcal{Y}_d.\ \varphi_{allTraces}(\mathcal{Y}_a) \wedge \varphi_{onlyTraces}(\mathcal{Y}_d) \wedge (ar(\varphi))(0)$$

of third-order arithmetic where $\mathcal{Y}_a$ represents the distinguished Hyper^{2}LTL variable X_a, $\mathcal{Y}_d$ represents the distinguished Hyper^{2}LTL variable X_d, and where $ar(\varphi)$ is the encoding of φ in Hyper^{2}LTL.

To encode the general satisfiability problem it was sufficient to express that $\mathcal{Y}_d$ only contains traces. Here, we now require that $\mathcal{Y}_d$ contains exactly the traces of some finite transition system, which can easily be expressed in second-order arithmetic[2] as follows.

We begin with a formula $\varphi_{isTS}(n, E, I, \ell)$ expressing that the second-order variables E, I, and ℓ encode a transition system with set $\{0, 1, \ldots, n-1\}$ of vertices. Our encoding will make extensive use of the pairing function introduced in the proof of Theorem 9. Formally, we define $\varphi_{isTS}(n, E, I, \ell)$ as the conjunction of the following formulas (where all quantifiers are first-order and we use *pair* as syntactic sugar):

- $n > 0$: the transition system is nonempty.
- $\forall y.\ y \in E \rightarrow \exists v.\ \exists v'.\ (v < n \wedge v' < n \wedge y = pair(v, v'))$: edges are pairs of vertices.
- $\forall v.\ v < n \rightarrow \exists v'.\ (v' < n \wedge pair(v, v') \in E)$: every vertex has a successor.
- $\forall v.\ v \in I \rightarrow v < n$: the set of initial vertices is a subset of the set of all vertices.

[2] With a little more effort, and a little less readability, first-order suffices for this task, as finite transition systems can be encoded by natural numbers.

- $\forall y.\ y \in \ell \rightarrow \exists v.\ \exists p.\ (v < n \wedge p < |\mathrm{AP}| \wedge y = \mathit{pair}(v,p))$: the labeling of v by p is encoded by the pair (v,p). Here, we again assume AP to be fixed and therefore can use $|\mathrm{AP}|$ as a constant.

Next, we define $\varphi_{\mathit{isPath}}(P, n, E, I)$, expressing that the second-order variable P encodes a path through the transition system encoded by n, E, and I, as the conjunction of the following formulas:

- $\forall j.\ \exists v.\ (v < n \wedge \mathit{pair}(j,v) \in P \wedge \neg\exists v'.\ (v' \neq v \wedge \mathit{pair}(j,v') \in P))$: the fact that at position j the path visits vertex v is encoded by the pair (j,v). Exactly one vertex is visited at each position.
- $\exists v.\ v \in I \wedge \mathit{pair}(0,v) \in P$: the path starts in an initial vertex.
- $\forall j.\ \exists v.\ \exists v'.\ \mathit{pair}(j,v) \in P \wedge \mathit{pair}(j+1,v') \in P \wedge \mathit{pair}(v,v') \in E$: successive vertices in the path are indeed connected by an edge.

Finally, we define $\varphi_{\mathit{traceOf}}(T, P, \ell)$, expressing that the second-order variable T encodes the trace (using the encoding from the proof of Theorem 9) of the path encoded by the second-order variable P, as the following formula:

- $\forall j.\ \forall p.\ \mathit{pair}(j,p) \in T \leftrightarrow (\exists v.\ \mathit{pair}(j,v) \in P \wedge \mathit{pair}(v,p) \in \ell)$: a proposition holds in the trace at position j if and only if it is in the labeling of the j-th vertex of the path.

Now, we define the sentence φ'' as

$$\begin{aligned}
&\exists \mathcal{Y}_a.\ \exists \mathcal{Y}_d.\ \varphi_{\mathit{allTraces}}(\mathcal{Y}_a) \wedge \varphi_{\mathit{onlyTraces}}(\mathcal{Y}_d) \wedge \\
&\qquad \Big[\underbrace{\exists n.\ \exists E.\ \exists I.\ \exists \ell.\ \varphi_{\mathit{isTS}}(n, E, I, \ell)}_{\text{there exists a transition system } \mathfrak{T}} \wedge \\
&\qquad \underbrace{(\forall T.\ T \in \mathcal{Y}_d \rightarrow \exists P.\ (\varphi_{\mathit{isPath}}(P, n, E, I) \wedge \varphi_{\mathit{traceOf}}(T, P, \ell)))}_{\mathcal{Y}_d \text{ contains only traces of paths through } \mathfrak{T}} \wedge \\
&\qquad \underbrace{(\forall P.\ (\varphi_{\mathit{isPath}}(P, n, E, I) \rightarrow \exists T.\ T \in \mathcal{Y}_d \wedge \varphi_{\mathit{traceOf}}(T, P, \ell)))}_{\mathcal{Y}_d \text{ contains all traces of paths through } \mathfrak{T}.}\Big] \wedge (ar(\varphi))(0),
\end{aligned}$$

which holds in $(\mathbb{N}, +, \cdot, <, \in)$ if and only if φ is satisfied by a finite transition system. ◀

Again, the lower bound proof can easily be extended to the case of closed-world semantics, using the same arguments as in the case of general satisfiability.

▶ **Corollary 12.** *The* Hyper^{2}LTL *finite-state satisfiability problem under closed-world semantics is polynomial-time equivalent to truth in third-order arithmetic.*

5 The Complexity of Hyper^{2}LTL Model-Checking

The Hyper^{2}LTL model-checking problem asks, given a finite transition system $\mathfrak{T}$ and a Hyper^{2}LTL sentence φ, whether $\mathfrak{T} \models \varphi$. Beutner et al. [4] have shown that Hyper^{2}LTL model-checking is Σ_1^1-hard, but there is no known upper bound in the literature. We improve the lower bound considerably, i.e., also to truth in third-order arithmetic, and show that this bound is tight. This is the first upper bound on the problem's complexity.

▶ **Theorem 13.** *The* Hyper^{2}LTL *model-checking problem is polynomial-time equivalent to truth in third-order arithmetic. The lower bound already holds for X_a-free sentences.*

Proof. For the lower bound, we reduce truth in third-order arithmetic to the Hyper^{2}LTL model-checking problem: we present a polynomial-time translation from sentences φ of third-order arithmetic to pairs $(\mathfrak{T}, \varphi')$ of a finite transition system $\mathfrak{T}$ and a Hyper^{2}LTL sentence φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if $\mathfrak{T} \models \varphi'$.

In the proof of Theorem 9 we have, given a sentence φ of third-order arithmetic, constructed a Hyper^{2}LTL sentence φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if every set T of traces satisfies φ' (i.e., satisfaction is independent of the model). Thus, we obtain the lower bound by mapping φ to φ' and $\mathfrak{T}^*$, where $\mathfrak{T}^*$ is some fixed transition system.

For the upper bound, we reduce the Hyper^{2}LTL model-checking problem to truth in third-order arithmetic: we present a polynomial-time translation from pairs $(\mathfrak{T}, \varphi)$ of a finite transition system and a Hyper^{2}LTL sentence φ to sentences φ' of third-order arithmetic such that $\mathfrak{T} \models \varphi$ if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \varphi'$.

In the proof of Theorem 11, we have constructed, from a Hyper^{2}LTL sentence φ, a sentence φ' of third-order arithmetic that expresses the existence of a finite transition system that satisfies φ. We obtain the desired upper bound by modifying φ' to replace the existential quantification of the transition system by hardcoding $\mathfrak{T}$ instead. ◀

Again, the lower bound proof can easily be extended to closed-world semantics, using the same arguments as in the case of satisfiability.

▶ **Corollary 14.** *The* Hyper^{2}LTL *model-checking problem under closed-world semantics is polynomial-time equivalent to truth in third-order arithmetic.*

6 Hyper^{2}LTL$_{\text{mm}}$

As we have seen, unrestricted second-order quantification makes Hyper^{2}LTL very expressive and therefore highly undecidable. But restricted forms of second-order quantification are sufficient for many application areas. Beutner et al. [4] introduced Hyper^{2}LTL$_{\text{mm}}$, a fragment[3] of Hyper^{2}LTL in which second-order quantification ranges over smallest/largest sets that satisfy a given guard. For example, the formula $\exists(X, \curlyvee, \varphi_1).\ \varphi_2$ expresses that there is a set T of traces that satisfies both φ_1 and φ_2, and T is *a* smallest set that satisfies φ_1 (i.e., φ_1 is the guard). This fragment is expressive enough to express common knowledge, asynchronous hyperproperties, and causality in reactive systems [4].

The formulas of Hyper^{2}LTL$_{\text{mm}}$ are given by the grammar

$$\varphi ::= \exists(X, \curlyvee\!\!\!\curlywedge, \varphi).\ \varphi \mid \forall(X, \curlyvee\!\!\!\curlywedge, \varphi).\ \varphi \mid \exists\pi \in X.\ \varphi \mid \forall\pi \in X.\ \varphi \mid \psi$$
$$\psi ::= \mathtt{p}_\pi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

where $\mathtt{p}$ ranges over AP, π ranges over $\mathcal{V}_1$, X ranges over $\mathcal{V}_2$, and $\curlyvee\!\!\!\curlywedge \in \{\curlyvee, \curlywedge\}$, i.e., the only modification concerns the syntax of second-order quantification.

Accordingly, the semantics of Hyper^{2}LTL$_{\text{mm}}$ is similar to that of Hyper^{2}LTL but for the second-order quantifiers, for which we define (for $\curlyvee\!\!\!\curlywedge \in \{\curlyvee, \curlywedge\}$):

- $\Pi \models \exists(X, \curlyvee\!\!\!\curlywedge, \varphi_1).\ \varphi_2$ if there exists a set $T \in \mathrm{sol}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ such that $\Pi[X \mapsto T] \models \varphi_2$
- $\Pi \models \forall(X, \curlyvee\!\!\!\curlywedge, \varphi_1).\ \varphi_2$ if for all sets $T \in \mathrm{sol}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ we have $\Pi[X \mapsto T] \models \varphi_2$

[3] In [4] this fragment is termed Hyper^{2}LTL$_{\text{fp}}$.

Here, $\mathrm{sol}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ is the set of all minimal/maximal models of the formula φ_1, which is defined as follows:

$$\mathrm{sol}(\Pi, (X, \curlyvee, \varphi_1)) = \{T \subseteq (2^{\mathrm{AP}})^\omega \mid \Pi[X \mapsto T] \models \varphi_1 \text{ and } \Pi[X \mapsto T'] \not\models \varphi_1 \text{ for all } T' \subsetneq T\}$$
$$\mathrm{sol}(\Pi, (X, \curlywedge, \varphi_1)) = \{T \subseteq (2^{\mathrm{AP}})^\omega \mid \Pi[X \mapsto T] \models \varphi_1 \text{ and } \Pi[X \mapsto T'] \not\models \varphi_1 \text{ for all } T' \supsetneq T\}$$

Note that $\mathrm{sol}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ may be empty, may be a singleton, or may contain multiple sets, which then are pairwise incomparable.

Let us also define closed-world semantics for Hyper^{2}LTL$_{\mathrm{mm}}$. Here, we again disallow the use of the variable X_a and change the semantics of set quantification to

- $\Pi \models_{\mathrm{cw}} \exists(X, \curlyvee\!\!\!\curlywedge, \varphi_1).\ \varphi_2$ if there exists a set $T \in \mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ such that $\Pi[X \mapsto T] \models \varphi_2$, and
- $\Pi \models_{\mathrm{cw}} \forall(X, \curlyvee\!\!\!\curlywedge, \varphi_1).\ \varphi_2$ if for all sets $T \in \mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ we have $\Pi[X \mapsto T] \models \varphi_2$,

where $\mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlyvee, \varphi_1))$ and $\mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlywedge, \varphi_1))$ are defined as follows:

$$\begin{aligned}\mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlyvee, \varphi_1)) = \{T \subseteq \Pi(X_d) \mid{}& \Pi[X \mapsto T] \models_{\mathrm{cw}} \varphi_1 \\ &\text{and } \Pi[X \mapsto T'] \not\models_{\mathrm{cw}} \varphi_1 \text{ for all } T' \subsetneq T\} \\ \mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlywedge, \varphi_1)) = \{T \subseteq \Pi(X_d) \mid{}& \Pi[X \mapsto T] \models_{\mathrm{cw}} \varphi_1 \\ &\text{and } \Pi[X \mapsto T'] \not\models_{\mathrm{cw}} \varphi_1 \text{ for all } \Pi(X_d) \supseteq T' \supsetneq T\}.\end{aligned}$$

Note that $\mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlyvee\!\!\!\curlywedge, \varphi_1))$ may still be empty, may be a singleton, or may contain multiple sets, but all sets in it are now incomparable subsets of $\Pi(X_d)$.

A Hyper^{2}LTL$_{\mathrm{mm}}$ formula is a sentence if it does not have any free variables except for X_a and X_d (also in the guards). Models are defined as for Hyper^{2}LTL.

▶ **Proposition 15** (Proposition 1 of [4]). *Every* Hyper^{2}LTL$_{\mathrm{mm}}$ *sentence φ can be translated in polynomial time (in $|\varphi|$) into a* Hyper^{2}LTL *sentence φ' such that for all sets T of traces we have that $T \models \varphi$ if and only if $T \models \varphi'$.*[4]

The same claim is also true for closed-world semantics, using the same proof.

▶ Remark 16. Every Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ can be translated in polynomial time (in $|\varphi|$) into a Hyper^{2}LTL sentence φ' such that for all sets T of traces we have that $T \models_{\mathrm{cw}} \varphi$ if and only if $T \models_{\mathrm{cw}} \varphi'$.

Thus, every complexity upper bound for Hyper^{2}LTL also holds for Hyper^{2}LTL$_{\mathrm{mm}}$ and every lower bound for Hyper^{2}LTL$_{\mathrm{mm}}$ also holds for Hyper^{2}LTL. In the following, we show that lower bounds can also be transferred in the other direction, i.e., from Hyper^{2}LTL to Hyper^{2}LTL$_{\mathrm{mm}}$. Thus, contrary to the design goal of Hyper^{2}LTL$_{\mathrm{mm}}$, it is in general not more feasible than full Hyper^{2}LTL.

We begin again by studying the cardinality of models of Hyper^{2}LTL$_{\mathrm{mm}}$ sentences, which will be the key technical tool for our complexity results. Again, as such formulas are evaluated over sets of traces, whose cardinality is bounded by $\mathfrak{c}$, there is a trivial upper bound. Our main result is that this bound is tight even for the restricted setting of Hyper^{2}LTL$_{\mathrm{mm}}$. The proof is similar to the one of Theorem 7, we just have to modify φ_4 so that the universal second-order quantifier only ranges over maximal contradiction-free sets.

[4] The polynomial-time claim is not made in [4], but follows from the construction when using appropriate data structures for formulas.

▶ **Theorem 17.** *There is a satisfiable X_a-free* $\text{Hyper}^2\text{LTL}_{\text{mm}}$ *sentence that only has models of cardinality $\mathfrak{c}$ (under standard and closed-world semantics).*

Now, let us describe how we settle the complexity of $\text{Hyper}^2\text{LTL}_{\text{mm}}$ satisfiability and model-checking: Recall that Hyper^2LTL allows set quantification over arbitrary sets of traces while $\text{Hyper}^2\text{LTL}_{\text{mm}}$ restricts quantification to minimal/maximal sets of traces that satisfy a guard formula. By using a sentence $\varphi_{\mathfrak{c}}$ as guard that has only models of cardinality $\mathfrak{c}$, the minimal sets satisfying the guard have cardinality $\mathfrak{c}$. Thus, we can obtain every possible set over propositions not used by $\varphi_{\mathfrak{c}}$ as the projection of a subset of a minimal set satisfying the guard $\varphi_{\mathfrak{c}}$. Thus, quantification of arbitrary sets of traces can be mimicked by quantification of minimal and maximal sets satisfying a guard.

▶ **Theorem 18.** $\text{Hyper}^2\text{LTL}_{\text{mm}}$ *satisfiability, finite-state satisfiability, and model-checking are polynomial-time equivalent to truth in third-order arithmetic. The lower bounds hold even for X_a-free sentences.*

Let us conclude by mentioning that Theorem 18 can again be extended to $\text{Hyper}^2\text{LTL}_{\text{mm}}$ under closed-world semantics, using the same arguments as for full Hyper^2LTL.

▶ **Corollary 19.** $\text{Hyper}^2\text{LTL}_{\text{mm}}$ *satisfiability, finite-state satisfiability, and model-checking under closed-world semantics are polynomial-time equivalent to truth in third-order arithmetic.*

7 The Least Fixed Point Fragment of $\text{Hyper}^2\text{LTL}_{\text{mm}}$

We have seen that even restricting second-order quantification to smallest/largest sets that satisfy a guard formula is essentially as expressive as full Hyper^2LTL, and thus as difficult. However, Beutner et al. [4] note that applications like common knowledge and asynchronous hyperproperties do not even require quantification over smallest/largest sets satisfying a guard, they "only" require quantification over least fixed points of HyperLTL definable functions. This finally yields a fragment with (considerably) lower complexity: we show that satisfiability under closed-world semantics is Σ_1^1-complete while finite-state satisfiability and model-checking are in Σ_2^2 and Σ_1^1-hard (under both semantics). For satisfiability under closed-world semantics, this matches the complexity of HyperLTL satisfiability.

A $\text{Hyper}^2\text{LTL}_{\text{mm}}$ sentence using only minimality constraints has the form

$$\varphi = \gamma_1.\ Q_1(Y_1, \curlyvee, \varphi_1^{\text{con}}).\ \gamma_2.\ Q_2(Y_2, \curlyvee, \varphi_2^{\text{con}}).\ \ldots \gamma_k.\ Q_k(Y_k, \curlyvee, \varphi_k^{\text{con}}).\ \gamma_{k+1}.\ \psi$$

satisfying the following properties:

- Each γ_j is a block $\gamma_j = Q_{\ell_{j-1}+1}\pi_{\ell_{j-1}+1} \in X_{\ell_{j-1}+1} \cdots Q_{\ell_j}\pi_{\ell_j} \in X_{\ell_j}$ of trace quantifiers (with $\ell_0 = 0$). As φ is a sentence, this implies that we have $\{X_{\ell_j+1}, \ldots, X_{\ell_j}\} \subseteq \{X_a, X_d, Y_1, \ldots, Y_{j-1}\}$.
- The free variables of ψ_j^{con} are among the trace variables quantified in the $\gamma_{j'}$ and $X_a, X_d, Y_1, \ldots, Y_j$.
- ψ is a quantifier-free formula. Again, as φ is a sentence, the free variables of ψ are among the trace variables quantified in the γ_j.

Now, φ is an lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ sentence[5], if additionally each φ_j^{con} has the form

$$\varphi_j^{\text{con}} = \dot\pi_1 \triangleright Y_j \wedge \cdots \wedge \dot\pi_n \triangleright Y_j \wedge \forall \ddot\pi_1 \in Z_1.\ \ldots \forall \ddot\pi_{n'} \in Z_{n'}.\ \psi_j^{\text{step}} \rightarrow \ddot\pi_m \triangleright Y_j$$

[5] Our definition here differs slightly from the one of [4] in that we allow to express the existence of some traces in the fixed point (via the subformulas $\dot\pi_i \triangleright Y_j$). All examples and applications of [4] are also of this form.

for some $n \geq 0$, $n' \geq 1$, where $1 \leq m \leq n'$, and where we have

- $\{\dot\pi_1, \ldots, \dot\pi_n\} \subseteq \{\pi_1, \ldots, \pi_{\ell_j}\}$,
- $\{Z_1, \ldots, Z_{n'}\} \subseteq \{X_a, X_d, Y_1, \ldots, Y_j\}$, and
- ψ_j^{step} is quantifier-free with free variables among $\ddot\pi_1, \ldots, \ddot\pi_{n'}, \pi_1, \ldots, \pi_{\ell_j}$.

As always, φ_j^{con} can be brought into the required prenex normal form.

Let us give some intuition for the definition. To this end, fix some $j \in \{1, 2, \ldots, k\}$ and a variable assignment Π whose domain contains at least all variables quantified before Y_j, i.e., all $Y_{j'}$ and all variables in the $\gamma_{j'}$ for $j' < j$, as well as X_a and X_d. Then,

$$\varphi_j^{\text{con}} = \dot\pi_1 \in Y_j \wedge \cdots \wedge \dot\pi_n \in Y_j \wedge \big(\forall \ddot\pi_1 \in Z_1.\ \ldots \forall \ddot\pi_{n'} \in Z_{n'}.\ \psi_j^{\text{step}} \rightarrow \ddot\pi_m \triangleright Y_j\big)$$

induces the monotonic function $f_{\Pi,j} \colon 2^{(2^{\text{AP}})^\omega} \rightarrow 2^{(2^{\text{AP}})^\omega}$ defined as

$$f_{\Pi,j}(S) = S \cup \{\Pi(\dot\pi_1), \ldots, \Pi(\dot\pi_n)\} \cup \{\Pi'(\ddot\pi_m) \mid \Pi' = \Pi[\ddot\pi_1 \mapsto t_1, \ldots, \ddot\pi_{n'} \mapsto t_{n'}]$$
$$\text{for } t_i \in \Pi(Z_i) \text{ if } Z_i \neq Y_j \text{ and } t_i \in S \text{ if } Z_i = Y_j \text{ s.t. } \Pi' \models \psi_j^{\text{step}}\}.$$

We define $S_0 = \emptyset$, $S_{\ell+1} = f_{\Pi,j}(S_\ell)$, and

$$\text{lfp}(\Pi, j) = \bigcup\nolimits_{\ell \in \mathbb{N}} S_\ell,$$

which is the least fixed point of $f_{\Pi,j}$. Due to the minimality constraint on Y_j in φ, $\text{lfp}(\Pi, j)$ is the unique set in $\text{sol}(\Pi, (Y_j, \curlyvee, \varphi_j^{\text{con}}))$. Hence, an induction shows that $\text{lfp}(\Pi, j)$ only depends on the values $\Pi(\pi)$ for trace variables π quantified before Y_j as well as the values $\Pi(X_d)$ and $\Pi(X_a)$, but not on the values $\Pi(Y_{j'})$ for $j' < j$ (as they are unique).

Thus, as $\text{sol}(\Pi, (Y_j, \curlyvee, \varphi_j^{\text{con}}))$ is a singleton, it is irrelevant whether Q_j is an existential or a universal quantifier. Instead of interpreting second-order quantification as existential or universal, here one should understand it as a deterministic least fixed point computation: choices for the trace variables and the two distinguished second-order variables uniquely determine the set of traces that a second-order quantifier assigns to a second-order variable.

▶ Remark 20. Note that the traces that are added to a fixed point assigned to Y_j either come from another $Y_{j'}$ with $j' < j$, from the model (via X_d), or from the set of all traces (via X_a). Thus, for X_a-free formulas, all second-order quantifiers range over (unique) subsets of the model, i.e., there is no need for an explicit definition of closed-world semantics. The analogue of closed-world semantics for lfp-Hyper^{2}LTL$_{\text{mm}}$ is to restrict oneself to X_a-free sentences.

In the remainder of this section, we study the complexity of lfp-Hyper^{2}LTL$_{\text{mm}}$. For satisfiability, the key step is again to study the size of models of satisfiable sentences. For X_a-free lfp-Hyper^{2}LTL$_{\text{mm}}$, as for HyperLTL, we are able to show that each satisfiable sentence has a countable model. The following result is proven by generalizing the proof for the analogous result for HyperLTL [18] showing that every model T of a HyperLTL sentence φ contains a countable $R \subseteq T$ that is closed under the application of Skolem functions. This implies that R is also a model of φ.

▶ **Lemma 21.** *Every satisfiable X_a-free* lfp-Hyper^{2}LTL$_{\text{mm}}$ *sentence has a countable model.*

Proof Sketch. Let $\varphi = \gamma_1 Q_1(Y_1, \curlyvee, \varphi_1^{\text{con}}).\ \gamma_2 Q_2(Y_2, \curlyvee, \varphi_2^{\text{con}}).\ \ldots \gamma_k Q_2(Y_k, \curlyvee, \varphi_k^{\text{con}}).\ \gamma_{k+1}.\ \psi$ be a satisfiable lfp-Hyper^{2}LTL$_{\text{mm}}$ sentence where

$$\varphi_j^{\text{con}} = \dot\pi_1 \triangleright Y_j \wedge \cdots \wedge \dot\pi_n \triangleright Y_j \wedge \forall \ddot\pi_1 \in Z_1.\ \ldots \forall \ddot\pi_{n'} \in Z_{n'}.\ \psi_j^{\text{step}} \rightarrow \ddot\pi_m \triangleright Y_j.$$

We assume w.l.o.g. that each trace variable is quantified at most once in φ. This implies that for each trace variable π quantified in some γ_j or in some φ_j^{con}, there is a unique second-order variable X_π such that π ranges over X_π.

Membership of traces in least fixed points assigned to the variables Y_j can be characterized by trees labeled by traces that make the inductive construction of the stages of the least fixed points explicit. Intuitively, consider the formula φ_j^{con} above inducing the unique least fixed point $\mathrm{lfp}(\Pi, j)$ that Y_j ranges over. It expresses that a trace t is in the fixed point either because it is of the form $\Pi(\dot{\pi}_i)$ for some $i \in \{1, \ldots, n\}$ where $\dot{\pi}_i$ is a trace variable quantified before the quantification of Y_j, or t is in the fixed point because there are traces $t_1, \ldots, t_{n'}$ such that assigning them to the $\ddot{\pi}_i$ satisfies ψ_j^{step} and $t = t_m$. Thus, the traces $t_1, \ldots, t_{n'}$ witness that t is in the fixed point. However, each t_i must be selected from $\Pi(Z_i)$, which, if $Z_i = Y_{j'}$ for some j', again needs witnesses. Thus, a witness is in general a tree whose vertices are labeled by traces and indexes in $\{1, 2,' ldots, k\}$ indicating in which fixed point the trace is in.

As φ is satisfiable, there exists a set T of traces such that $T \models \varphi$. We show that there is a countable $R \subseteq T$ with $R \models \varphi$. Intuitively, we show that the smallest set R that is closed under the application of the Skolem functions and that contains the traces labeling witness trees (for the fixed points computed w.r.t. T) for the traces in R has the desired properties.

The full proof requires additional notation, e.g., a formalization of the notion of witness trees, and can be found in the full version [21]. ◀

Before we continue with our complexity results, let us briefly mention that the formula from Remark 5 on Page 7 shows that the restriction to X_a-free sentences is essential to obtain the upper bound above.

With this upper bound, we can express the existence of (w.l.o.g.) countable models of a given X_a-free sentence φ via arithmetic formulas that only use existential quantification of type 1 objects (sets of natural numbers), which are rich enough to express countable sets T of traces and objects (e.g., Skolem functions and more) witnessing that T satisfies φ. This places satisfiability in Σ_1^1 while the matching lower bound already holds for HyperLTL [19].

▶ **Theorem 22.** $\mathrm{lfp}\text{-}\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ *satisfiability for X_a-free sentences is Σ_1^1-complete.*

Proof Sketch. The Σ_1^1 lower bound already holds for HyperLTL satisfiability [19], as HyperLTL is a fragment of X_a-free $\mathrm{lfp}\text{-}\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ (see Remark 1). Hence, we focus in the following on the upper bound, which is a generalization of the corresponding upper bound for HyperLTL [19].

Let φ be an X_a-free $\mathrm{lfp}\text{-}\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ sentence. From Lemma 21, φ is satisfiable if and only if it has a countable model T. Thus, to prove that the $\mathrm{lfp}\text{-}\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ satisfiability problem for X_a-free sentences is in Σ_1^1, we express the existence of a countable set T of traces and a witness that T is indeed a model of φ.

As we want to show a Σ_1^1 upper bound, we have to express the existence of a countable model by a sentence of arithmetic with existential quantification over sets of natural numbers and existential and universal quantification over natural numbers. A bit more in detail, since we only have to work with countable sets (as second-order quantifiers in φ range over subsets of the countable model), we can use natural numbers to "name" traces. Thus, a countable set of traces is a mapping from $\mathbb{N} \times \mathbb{N}$ (names and positions) to 2^{AP}, which can be encoded by a set of natural numbers. Then, we can encode the existence of the following type 1 objects:

- Variable assignments, such that membership of their assigned traces into respective fixed point sets can be captured in first-order arithmetic.
- Functions for the existentially quantified first-order variables of φ, which can be verified to be Skolem functions (in first-order arithmetic).
- Functions expressing the satisfaction of subformulas of φ.

Furthermore, first-order arithmetic can express that the variable assignments indeed map set variables to least fixed points.

Altogether, this allows us to capture the satisfiability of lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ in Σ_1^1. ◀

Finally, we consider finite-state satisfiability and model-checking. Note that we have to deal with uncountable sets of traces in both problems, as the sets of traces of finite transition systems may be uncountable. The lower bounds are proven by reductions from a variant of the recurrent tiling problem [24] while the upper bounds are obtained by expressing least fixed points in second-order arithmetic.

▶ **Theorem 23.** lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *finite-state satisfiability and model-checking are both in* Σ_2^2 *and* Σ_1^1*-hard, where the lower bounds already hold for* X_a*-free sentences.*

8 Related Work

As mentioned in Section 1, the complexity problems for HyperLTL were thoroughly studied [16, 19, 20]. For Hyper^{2}LTL, Beutner et al. mainly focused on the algorithmic aspects by providing model checking [4] and monitoring [5] algorithms, and did not study the respective complexity problems in depth.

Logics related to Hyper^{2}LTL are asynchronous and epistemic logics. Much research has been done regarding epistemic properties [13, 15, 29, 36] and their relations to hyperproperties [8]. However, most of this work concerns expressiveness and decidability results (e.g., [7]), and not complexity analysis for the undecidable fragments. This is similar for asynchronous hyperlogics [1, 2, 3, 6, 9, 10, 23, 26, 27, 28], where most work concerns decidability results and expressive power, but not complexity analysis.

Another related logic is TeamLTL [28], a hyperlogic for the specification of dependence and independence. Lück [30] studied similar problems to those we study in this paper and showed that, in general, satisfiability and model checking of TeamLTL with Boolean negation is equivalent to truth in third-order arithmetic. Kontinen and Sandström [25] generalize this result and show that any logic between TeamLTL with Boolean negation and second-order logic inherits the same complexity results. Kontinen et al. [26] study set semantics for asynchronous TeamLTL, and provide positive complexity and decidability results. Gutsfeld et al. [22] study an extension of TeamLTL to express refined notions of asynchronicity and analyze the expressiveness and complexity of their logic, proving it also highly undecidable. While TeamLTL is closely related to Hyper^{2}LTL, the exact relation between them is still unknown.

9 Conclusion

We have investigated and settled the complexity of satisfiability, finite-state satisfiability, and model-checking for Hyper^{2}LTL and Hyper^{2}LTL$_{\mathrm{mm}}$ and (almost) settled it for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$. For the former two, all three problems are equivalent to truth in third-order arithmetic, and therefore (not surprisingly) much harder than the corresponding problems for HyperLTL, which are "only" Σ_1^1-complete, Σ_1^0-complete, and Tower-complete, respectively. This shows that the addition of second-order quantification increases the already high complexity of HyperLTL significantly. However, for the fragment lfp-Hyper^{2}LTL$_{\mathrm{mm}}$, in which second-order quantification degenerates to least fixed point computations, the complexity is much lower: satisfiability under closed-world semantics is Σ^1_1-complete and finite-state satisfiability as well as model-checking are in Σ^2_2.

Recently, Regaud and Zimmermann [34] have solved several problems left open in this work, e.g., they settled the complexity of $\text{Hyper}^2\text{LTL}_{\text{mm}}$ with only minimality constraints or only maximality constraints, the complexity of lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ under standard semantics, and closed the gaps in our results for lfp-$\text{Hyper}^2\text{LTL}_{\text{mm}}$ finite-state satisfiability and model-checking. Furthermore, they settled the complexity of all three decision problems we consider here for HyperQPTL [33].

References

1 Ezio Bartocci, Thomas A. Henzinger, Dejan Nickovic, and Ana Oliveira da Costa. Hypernode automata. In Guillermo A. Pérez and Jean-François Raskin, editors, *CONCUR 2023*, volume 279 of *LIPIcs*, pages 21:1–21:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CONCUR.2023.21.

2 Jan Baumeister, Norine Coenen, Borzoo Bonakdarpour, Bernd Finkbeiner, and César Sánchez. A temporal logic for asynchronous hyperproperties. In Alexandra Silva and K. Rustan M. Leino, editors, *CAV 2021, Part I*, volume 12759 of *LNCS*, pages 694–717. Springer, 2021. doi:10.1007/978-3-030-81685-8_33.

3 Raven Beutner and Bernd Finkbeiner. HyperATL*: A logic for hyperproperties in multi-agent systems. *Log. Methods Comput. Sci.*, 19(2), 2023. doi:10.46298/LMCS-19(2:13)2023.

4 Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. Second-order hyperproperties. In Constantin Enea and Akash Lal, editors, *CAV 2023, Part II*, volume 13965 of *LNCS*, pages 309–332. Springer, 2023. doi:10.1007/978-3-031-37703-7_15.

5 Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. Monitoring second-order hyperproperties. In Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum, editors, *AAMAS 2024*, pages 180–188. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2024. doi:10.5555/3635637.3662865.

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

7 Laura Bozzelli, Bastien Maubert, and Aniello Murano. On the complexity of model checking knowledge and time. *ACM Trans. Comput. Log.*, 25(1):8:1–8:42, 2024. doi:10.1145/3637212.

8 Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Unifying hyper and epistemic temporal logics. In Andrew M. Pitts, editor, *FoSSaCS 2015*, volume 9034 of *LNCS*, pages 167–182. Springer, 2015. doi:10.1007/978-3-662-46678-0_11.

9 Laura Bozzelli, Adriano Peron, and César Sánchez. Asynchronous extensions of HyperLTL. In *LICS 2021*, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470583.

10 Laura Bozzelli, Adriano Peron, and César Sánchez. Expressiveness and decidability of temporal logics for asynchronous hyperproperties. In Bartek Klin, Slawomir Lasota, and Anca Muscholl, editors, *CONCUR 2022*, volume 243 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CONCUR.2022.27.

11 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.

12 Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *J. Comput. Secur.*, 18(6):1157–1210, 2010. doi:10.3233/JCS-2009-0393.

13 Catalin Dima. Revisiting satisfiability and model-checking for CTLK with synchrony and perfect recall. In Michael Fisher, Fariba Sadri, and Michael Thielscher, editors, *CLIMA IX*, volume 5405 of *LNCS*, pages 117–131. Springer, 2008. doi:10.1007/978-3-642-02734-5_8.

14 E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986. doi:10.1145/4904.4999.

15 Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995. doi:10.7551/MITPRESS/5803.001.0001.

16 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.

17 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.

18 Bernd Finkbeiner and Martin Zimmermann. The First-Order Logic of Hyperproperties. In *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.

19 Marie Fortin, Louwe B. Kuijer, Patrick Totzke, and Martin Zimmermann. HyperLTL satisfiability is Σ_1^1-complete, HyperCTL* satisfiability is Σ_1^2-complete. In Filippo Bonchi and Simon J. Puglisi, editors, *MFCS 2021*, volume 202 of *LIPIcs*, pages 47:1–47:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.MFCS.2021.47.

20 Marie Fortin, Louwe B. Kuijer, Patrick Totzke, and Martin Zimmermann. HyperLTL satisfiability is highly undecidable, HyperCTL* is even harder. *arXiv*, 2303.16699, 2023. Journal version of [19]. Under submission. doi:10.48550/arXiv.2303.16699.

21 Hadar Frenkel and Martin Zimmermann. The complexity of second-order HyperLTL. *arXiv*, 2311.15675, 2023. doi:10.48550/arXiv.2311.15675.

22 Jens Oliver Gutsfeld, Arne Meier, Christoph Ohrem, and Jonni Virtema. Temporal team semantics revisited. In Christel Baier and Dana Fisman, editors, *LICS 2022*, pages 44:1–44:13. ACM, 2022. doi:10.1145/3531130.3533360.

23 Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph Ohrem. Automata and fixpoints for asynchronous hyperproperties. *Proc. ACM Program. Lang.*, 5(POPL):1–29, 2021. doi:10.1145/3434319.

24 David Harel. Recurring Dominoes: Making the Highly Undecidable Highly Understandable. *North-Holland Mathematical Studies*, 102:51–71, 1985. doi:10.1016/S0304-0208(08)73075-5.

25 Juha Kontinen and Max Sandström. On the expressive power of TeamLTL and first-order team logic over hyperproperties. In Alexandra Silva, Renata Wassermann, and Ruy J. G. B. de Queiroz, editors, *WoLLIC 2021*, volume 13038 of *LNCS*, pages 302–318. Springer, 2021. doi:10.1007/978-3-030-88853-4_19.

26 Juha Kontinen, Max Sandström, and Jonni Virtema. Set semantics for asynchronous TeamLTL: Expressivity and complexity. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *MFCS 2023*, volume 272 of *LIPIcs*, pages 60:1–60:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.MFCS.2023.60.

27 Juha Kontinen, Max Sandström, and Jonni Virtema. A remark on the expressivity of asynchronous TeamLTL and HyperLTL. In Arne Meier and Magdalena Ortiz, editors, *FoIKS 2024*, volume 14589 of *LNCS*, pages 275–286. Springer, 2024. doi:10.1007/978-3-031-56940-1_15.

28 Andreas Krebs, Arne Meier, Jonni Virtema, and Martin Zimmermann. Team semantics for the specification and verification of hyperproperties. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *MFCS 2018*, volume 117 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.10.

29 Alessio Lomuscio and Franco Raimondi. The complexity of model checking concurrent programs against CTLK specifications. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss,

and Peter Stone, editors, *AAMAS 2006*, pages 548–550. ACM, 2006. doi:10.1145/1160633.1160733.

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

31 Corto Mascle and Martin Zimmermann. The keys to decidable HyperLTL satisfiability: Small models or very simple formulas. In Maribel Fernández and Anca Muscholl, editors, *CSL 2020*, volume 152 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CSL.2020.29.

32 Amir Pnueli. The temporal logic of programs. In *FOCS 1977*, pages 46–57. IEEE, October 1977. doi:10.1109/SFCS.1977.32.

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

34 Gaëtan Regaud and Martin Zimmermann. The complexity of fragments of second-order HyperLTL, 2025. Under preparation.

35 Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, MA, USA, 1987.

36 Ron van der Meyden and Nikolay V. Shilov. Model checking knowledge and time in systems with perfect recall (extended abstract). In C. Pandu Rangan, Venkatesh Raman, and Ramaswamy Ramanujam, editors, *FSTTCS 1999*, volume 1738 of *LNCS*, pages 432–445. Springer, 1999. doi:10.1007/3-540-46691-6_35.