THE COMPLEXITY OF SECOND-ORDER HYPERLTL

HADAR FRENKEL [a], GAËTAN REGAUD [b], AND MARTIN ZIMMERMANN [c]

[a] Bar-Ilan University, Ramat Gan, Israel
e-mail address: hadar.frenkel@biu.ac.il

[b] ENS Rennes, Rennes, France
e-mail address: gaetan.regaud@ens-rennes.fr

[c] Aalborg University, Aalborg, Denmark
e-mail address: mzi@cs.aau.dk

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_1^2-complete, and finite-state satisfiability and model-checking are equivalent to truth in second-order arithmetic.

Finally, we also introduce closed-world semantics for second-order HyperLTL, where set quantification ranges only over subsets of the model, while set quantification in standard semantics ranges over arbitrary sets of traces. Here, satisfiability for the least fixed point fragment becomes Σ_1^1-complete, but all other results are unaffected.

1. Introduction

The introduction of hyperlogics [CFK+14] for the specification and verification of hyperproperties [CS10] – 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* [CFK+14], the extensions of LTL [Pnu77] and CTL* [EH86] (respectively) with trace quantification, are natural specification languages for information-flow and security properties, have a decidable model-checking problem [FRS15], 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., [BHNdC23, BCB+21, BF23, BBST24, BPS21,

A preliminary version of this article has been published in the proceedings of CSL 2025 [FZ25]. Here, all problems left open in the conference version have been solved, as reported in the technical report [RZ25b].

© H. Frenkel, G. Regaud, and M. Zimmermann
Creative Commons

BPS22, GMO21, KSV23, KSV24, KMVZ18]). 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.

In light of this situation, Beutner et al. [BFFM23] 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 [BMP15], 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):

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

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 t'' is also 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. 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?

1.1. **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 [Rog87] for fully formal definitions and refer to Figure 1 for a visualization.

The class Σ_1^0 contains the sets of natural numbers of the form

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

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_1^0 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_1^0 is part of the first level of the arithmetical hierarchy while Σ_1^1 is part of the first level of the analytical

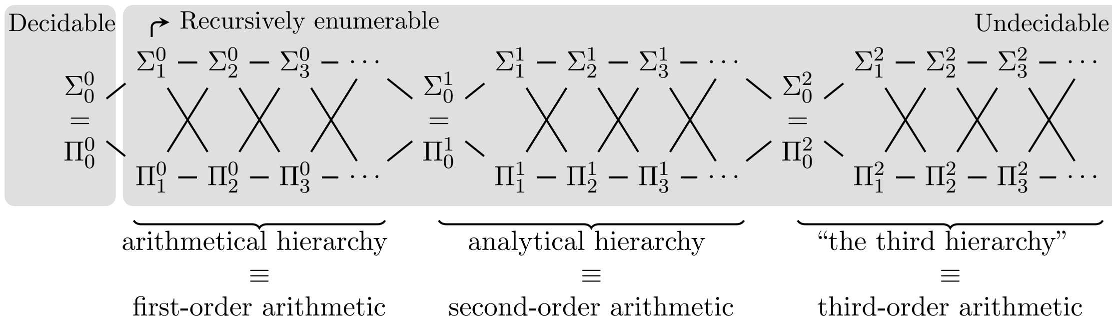


Figure 1: The arithmetical hierarchy, the analytical hierarchy, and beyond.

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 [FKTZ21], HyperLTL finite-state satisfiability is Σ^0_1-complete [FH16, FKTZ25], and, as mentioned above, Hyper^{2}LTL model-checking is Σ^1_1-hard [BFFM23], but, prior to this 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.

1.2. **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. [BFFM23], as well as for two variants 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 [FZ17]. This restricts the order of arithmetic that can be simulated in HyperLTL and explains in particular the Σ^1_1-completeness of HyperLTL satisfiability [FKTZ21]. 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 the analytic hierarchy (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^{2}LTL satisfiability also shed light on the exact complexity of Hyper^{2}LTL 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^{2}LTL 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 [Rab16, MZ20].

So, quantification over arbitrary sets of traces makes verification very hard. However, Beutner et al. [BFFM23] noticed that many of the applications of Hyper^{2}LTL 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 Hyper^{2}LTL$_{\mathrm{mm}}$)[1] and then further restrict those to least fixed points induced by HyperLTL definable operators (obtaining the fragment lfp-Hyper^{2}LTL$_{\mathrm{mm}}$). 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

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

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-Hyper^{2}LTL$_{\mathrm{mm}}$.

Nevertheless, we show that Hyper^{2}LTL$_{\mathrm{mm}}$ retains the same complexity as Hyper^{2}LTL, 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. Furthermore, we show that this is even the case when we allow only minimality constraints (Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$) and if we allow only maximality constraints (Hyper^{2}LTL$^{\curlywedge}_{\mathrm{mm}}$).

But if we consider lfp-Hyper^{2}LTL$_{\mathrm{mm}}$, the complexity finally decreases: we show that satisfiability is Σ^2_1-complete while finite-state satisfiability and model-checking are both equivalent to truth in second-order arithmetic.

Furthermore, we introduce and study a new semantics for Hyper^{2}LTL and its fragments: In standard semantics as introduced by Beutner et al. [BFFM23], second-order quantifiers range over arbitrary sets of traces that may contain traces that are not in the model. In closed-world semantics introduced here, second-order quantifiers range only over subsets of the model.

We show that for all but one case, the three problems have the same complexity under closed-world semantics as they have under standard semantics. The sole exception

[1]In [BFFM23] this fragment is termed Hyper^{2}LTL$_{\mathrm{fp}}$. For clarity, since it is not fixed point based, but uses minimality/maximality constraints, we use the subscript “mm” instead of “fp”.

is satisfiability for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ under closed-world semantics, which we prove Σ^1_1-complete, i.e., as hard as HyperLTL satisfiability. Stated differently, one can add least fixed points of HyperLTL definable operators to HyperLTL without increasing the complexity of the satisfiability problem.

Table 1 lists our results and compares them to the related logics LTL, HyperLTL, and HyperQPTL (see Section 8 for more details on these results). Recall that Beutner et al. showed that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ yields (partial) model checking and monitoring algorithms [BFFM23, BFFM24]. Our results confirm the usability of the lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ fragment also from a theoretical point of view, as all problems relevant for verification have significantly lower complexity (albeit, still highly undecidable).

Table 1: List of our results (in bold and red) and comparison to related logics [FH16, FKTZ25, FZ25, RZ24]. "T2A-equiv." stands for "equivalent to truth in second-order arithmetic", and "T3A-equiv." stands for "equivalent to truth in third-order arithmetic". Unless explicitly specified, results hold for both semantics.

Logic	Satisfiability	Finite-state satisfiability	Model-checking
LTL	PSpace-compl. [SC85]	PSpace-compl. [SC85]	PSpace-compl. [SC85]
HyperLTL	Σ^1_1-compl. [FKTZ21]	Σ^0_1-compl. [FH16, FKTZ25]	Tower-compl. [Rab16, MZ20]
Hyper^{2}LTL	**T3A-equiv.** (Thm.4.2)	**T3A-equiv.** (Thm.4.4)	**T3A-equiv.** (Thm.5.1)
Hyper^{2}LTL$_{\mathrm{mm}}$	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)
Hyper^{2}LTL$^{\curlywedge}_{\mathrm{mm}}$	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)
Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)	**T3A-equiv.** (Thm.6.3)
lfp-Hyper^{2}LTL$_{\mathrm{mm}}$	**Σ^2_1-compl.** (std) (Thm.7.7) **Σ^1_1-compl.** (cw) (Thm.7.6)	**T2A-equiv.** (Thm.7.9)	**T2A-equiv.** (Thm.7.9)
HyperQPTL	Σ^2_1-compl. [RZ24]	Σ^0_1-compl. [Rab16]	Tower-compl. [Rab16]
HyperQPTL$^+$	T3A-equiv. [RZ24]	T3A-equiv. [RZ24]	T3A-equiv. [RZ24]

1.3. **Outline.** In Section 2, we introduce Hyper^{2}LTL, the problems we are interested in, and how to use arithmetic to classify the complexity of undecidable problems. In Section 3, we show that there are Hyper^{2}LTL sentences that only have uncountable models. The proof of this result is then used in Section 4 to show that Hyper^{2}LTL satisfiability and finite-state satisfiability are equivalent to truth in third-order arithmetic. Then, in Section 5, we show that similar techniques allow us to show that Hyper^{2}LTL model-checking is equivalent to truth in third-order arithmetic.

Then, in Section 6, we show that all three problems for Hyper^{2}LTL can be reduced to the analogous problems for Hyper^{2}LTL$_{\mathrm{mm}}$ where all second-order quantifiers are guarded. Hence, all three problems for Hyper^{2}LTL$_{\mathrm{mm}}$ are equivalent to truth in third-order arithmetic as well.

Finally, in Section 7, we consider the fragment lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ where second-order quantification is restricted to computation of least fixed points. Here, all three problems are simpler, but still highly undecidable.

We conclude with discussing related work in Section 8 and with mentioning some questions for future work in Section 9.

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'. The AP'-projection of $T \subseteq (2^{\mathrm{AP}})^\omega$ is defined as the set of AP-projections of traces in T. Now, let AP and AP' be two disjoint sets, let t be a trace over AP, and let t' be a trace over AP'. Then, we define $t^\frown t'$ as the pointwise union of t and t', i.e., $t^\frown t'$ is the trace over $\mathrm{AP} \cup \mathrm{AP}'$ defined as $(t(0) \cup t'(0))(t(1) \cup t'(1))(t(2) \cup t'(2))\cdots$.

A finite 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 $\mathrm{Tr}(\mathfrak{T}) = \{\lambda(\rho) \mid \rho \text{ is a path through } \mathfrak{T}\}$.

2.1. **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

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

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.[2]

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 $\Pi[X \mapsto T] \models \varphi$.

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.

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

- We write $\pi_1 =_{\mathrm{AP}'} \pi_2$ for a set $\mathrm{AP}' \subseteq \mathrm{AP}$ for the formula $\mathbf{G} \bigwedge_{\mathtt{p} \in \mathrm{AP}'} (\mathtt{p}_{\pi_1} \leftrightarrow \mathtt{p}_{\pi_2})$ expressing that the AP'-projection of π_1 and the AP'-projection of π_2 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.

Remark 2.1. 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.

2.2. **Closed-World Semantics.** Second-order quantification in Hyper^{2}LTL as defined by Beutner et al. [BFFM23] (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:

[2] Note that the assignment of variables $X \in \mathcal{V}_2$ is not updated, as this is not necessary for our application: $\Pi[j,\infty)$ is used to define the semantics of the temporal operators and at that point the assignments to second-order variables is irrelevant, as we only consider formulas in prenex normal form.

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

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 2.2. *Every X_a-free* 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).*

Proof. Second-order quantification over subsets of the universe of discourse can easily be mimicked by guarding classical quantifiers ranging over arbitrary sets. Here, we rely on the formula $\psi_{\subseteq X_d}(X) = \forall \pi \in X.\ \pi \triangleright X_d$, which expresses that every trace in X is also in X_d.

Now, given an X_a-free Hyper^{2}LTL sentence φ, let φ' be the Hyper^{2}LTL sentence obtained by recursively replacing

- each existential second-order quantifier $\exists X.\ \psi$ in φ by $\exists X.\ \psi_{\subseteq X_d}(X) \wedge \psi$,
- each universal second-order quantifier $\forall X.\ \psi$ in φ by $\forall X.\ \psi_{\subseteq X_d}(X) \rightarrow \psi$,

and then bringing the resulting sentence into prenex normal form, which can be done as no quantifier is under the scope of a temporal operator. Then, we have $T \models_{\mathrm{cw}} \varphi$ 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 2.3. 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 $(2^{\mathrm{AP}})^\omega$.

2.3. **Problem Statement.**

We are interested in the complexity of the following three problems for fragments of Hyper^{2}LTL for both semantics:

- Satisfiability: Given a sentence φ, does it have a model, i.e., is there a set T of traces such that $T \models \varphi$?
- Finite-state satisfiability: Given a sentence φ, is it satisfied by a finite transition system, i.e., is there a finite transition system $\mathfrak{T}$ such that $\mathfrak{T} \models \varphi$?
- Model-checking: Given a sentence φ and a finite transition system $\mathfrak{T}$, do we have $\mathfrak{T} \models \varphi$?

2.4. **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 $\mathbb{N}$. In the following, we use lower-case roman letters x, y (possibly with decorations) for first-order variables, upper-case roman letters X, Y (possibly with decorations) for second-order variables, and upper-case calligraphic roman letters $\mathcal{X}, \mathcal{Y}$ (possibly with decorations) for third-order variables.

First-order arithmetic allows to quantify over type 0 objects, second-order arithmetic allows to quantify over type 0 and type 1 objects, and third-order arithmetic allows to quantify over type 0, type 1, and type 2 objects. Note that every fixed natural number is definable in first-order arithmetic, so we freely use them as syntactic sugar. Similarly, equality can be eliminated if necessary, as it can be expressed using $<$.

Truth in second-order arithmetic is the following problem: given a sentence φ of second-order arithmetic, does $(\mathbb{N}, +, \cdot, <, \in)$ satisfy φ? Truth in third-order arithmetic is defined analogously. Furthermore, 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 other quantifiers), and
- Σ_1^2 containing sets of the form
$$\{x \in \mathbb{N} \mid \exists \mathcal{X}_1 \subseteq 2^{\mathbb{N}}.\ \cdots \exists \mathcal{X}_k \subseteq 2^{\mathbb{N}}.\ \psi(x, \mathcal{X}_1, \ldots, \mathcal{X}_k)\},$$
where ψ is a formula of arithmetic with arbitrary quantification over type 0 and type 1 objects (but no other quantifiers).

Let A be an alphabet. We say that a language $L \subseteq A^*$ is Σ_1^1-complete if

- $\{e(w) \mid w \in L\}$ is in Σ_1^1 (where $e\colon A^* \to \mathbb{N}$ is a computable encoding of words over A by natural numbers), and
- for every $S \in \Sigma_1^1$ there is a computable function function $f\colon \mathbb{N} \to A^*$ such that $n \in S$ if and only if $f(n) \in L$.

We define Σ_1^2-completeness analogously. Note that all encodings e and functions f we use are polynomial-time computable.

Thus, while, e.g., Σ_1^1 captures the complexity of formulas with a specific pattern of quantifier alternation, truth in, e.g., second-order arithmetic allows for arbitrary second-order formulas.

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. This is an important step in classifying the complexity of Hyper^{2}LTL.

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 3.1. *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 3.2. 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 [FZ17]: Unsurprisingly, the addition of (even closed-world) second-order quantification increases the cardinality of minimal models, even without cheating.

Example 3.3. We begin by recalling a construction of Finkbeiner and Zimmermann giving a satisfiable HyperLTL sentence ψ that has no finite models [FZ17]. 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 $\mathrm{AP} = \{\mathtt{x}\}$ 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 $S' = \{\emptyset^n\{\mathtt{x}\}\emptyset^\omega \mid n \in S\}$ 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 3.4. *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 $\varphi_{\mathit{allSets}}$ whose unique model (under standard semantics) has cardinality $\mathfrak{c}$. To this end, we fix $\mathrm{AP} = \{+, -, \mathtt{s}, \mathtt{x}\}$ and consider the conjunction $\varphi_{\mathit{allSets}} = \varphi_0 \wedge \cdots \wedge \varphi_4$ of the following formulas:

- $\varphi_0 = \forall \pi \in X_d.\ \bigvee_{\mathtt{p} \in \{+,-,\mathtt{s}\}} \mathbf{G}(\mathtt{p}_\pi \wedge \bigwedge_{\mathtt{p}' \in \{+,-,\mathtt{s}\} \setminus \{\mathtt{p}\}} \neg \mathtt{p}'_\pi)$: In each trace of a model, one of the propositions in $\{+, -, \mathtt{s}\}$ holds at every position and the other two propositions in $\{+, -, \mathtt{s}\}$ hold at none of the positions. Consequently, we speak in the following about type $\mathtt{p}$ traces for $\mathtt{p} \in \{+, -, \mathtt{s}\}$.
- $\varphi_1 = \forall \pi \in X_d.\ (+_\pi \vee -_\pi) \rightarrow \neg \mathtt{x}_\pi \,\mathbf{U}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathbf{G} \neg \mathtt{x}_\pi)$: Type $\mathtt{p}$ traces for $\mathtt{p} \in \{+, -\}$ in the model have the form $\{\mathtt{p}\}^n\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$ for some $n \in \mathbb{N}$.
- $\varphi_2 = \bigwedge_{\mathtt{p} \in \{+,-\}} \exists \pi \in X_d.\ \mathtt{p}_\pi \wedge \mathtt{x}_\pi$: for both $\mathtt{p} \in \{+, -\}$, the type $\mathtt{p}$ trace $\{\mathtt{p}\}^0\{\mathtt{x}, \mathtt{p}\}\{\mathtt{p}\}^\omega$ is in every model.
- $\varphi_3 = \bigwedge_{\mathtt{p} \in \{+,-\}} \forall \pi \in X_d.\ \exists \pi' \in X_d.\ \mathtt{p}_\pi \rightarrow (\mathtt{p}_{\pi'} \wedge \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\, \mathtt{x}_{\pi'}))$: for both $\mathtt{p} \in \{+, -\}$, 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 3.3. So, every model of $\varphi_0 \wedge \cdots \wedge \varphi_3$ contains $\{\{+\}^n\{\mathtt{x},+\}\{+\}^\omega \mid n \in \mathbb{N}\}$ and $\{\{-\}^n\{\mathtt{x},-\}\{-\}^\omega \mid n \in \mathbb{N}\}$ as subsets, and no other type $+$ or type $-$ 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 it only contains traces of the form $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ or $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ and if there is no $n \in \mathbb{N}$ such that $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega \in T$ and $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega \in T$. Furthermore, a trace t over AP is consistent with a contradiction-free T if

(C1): $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega \in T$ implies $\mathtt{x} \in t(n)$ and
(C2): $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega \in T$ implies $\mathtt{x} \notin t(n)$.

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

- Hence, we define φ_4 as the formula

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

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

While $\varphi_{\mathit{allSets}}$ 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

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

of traces satisfies $\varphi_{\mathit{allSets}}$. On the other hand, every model of $\varphi_{\mathit{allSets}}$ must indeed contain T_{allSets} as a subset, as $\varphi_{\mathit{allSets}}$ 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 $\varphi_{\mathit{allSets}}$.

To conclude, we just remark that

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

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

Finally, let us consider closed-world semantics. The second-order quantifier in φ_4 (the only one in $\varphi_{\mathit{allSets}}$) is already restricted to subsets of the universe of discourse. Thus, $\varphi_{\mathit{allSets}}$ has the unique model T_{allSets} even under closed-world semantics. $\square$

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 $\mathtt{x}$ and encoded natural numbers as singleton sets. The proof of Theorem 3.4 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 [FKTZ25]. 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 4.1 (Lemma 5.5 of [FKTZ25]). *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 is at least as hard as truth in third-order arithmetic. A matching upper bound will be obtained by showing that one can encode (using arithmetic) traces as sets of natural numbers and thus sets of traces as sets of sets of natural numbers.

Theorem 4.2. *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

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

where

- $\varphi_{allSets}[X_d/X_{allSets}]$ is the Hyper^{2}LTL sentence from the proof of Theorem 3.4 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 3.4),
- $\varphi'_{(+,\cdot)}$ 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 $\mathit{hyp}(\varphi)$ is defined inductively as follows:

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

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

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

- For first-order variables y,

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

- $hyp(\psi_1 \vee \psi_2) = hyp(\psi_1) \vee hyp(\psi_2)$.
- $hyp(\neg\psi) = \neg hyp(\psi)$.
- 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 first-order arithmetic).

Let $pair\colon \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ 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 $e\colon \mathrm{AP} \rightarrow \{0, 1, \ldots, |\mathrm{AP}|-1\}$. 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

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

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
$$\varphi_{onlyTraces}(\mathcal{Y}) = \forall Y.\ Y \in \mathcal{Y} \rightarrow \varphi_{isTrace}(Y)$$
checks if a set of sets of natural numbers only contains sets encoding a trace.
- Further, the formula
$$\varphi_{allTraces}(\mathcal{Y}) = \varphi_{onlyTraces}(\mathcal{Y}) \wedge \forall Y.\ \varphi_{isTrace}(Y) \rightarrow Y \in \mathcal{Y}$$
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 $ar(\psi)$.
- $ar(\mathbf{X}\,\psi) = \exists i'(i' = i + 1) \wedge ar(\psi)$, 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 2.3 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 2.2, this settles the complexity of Hyper^{2}LTL satisfiability under closed-world semantics.

Corollary 4.3. *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_1^0-complete (a.k.a. recursively enumerable), as HyperLTL model-checking of finite transition systems is decidable [CFK$^+$14].

Theorem 4.4. *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 4.2, 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 by all sets T of traces (and in particular by some finite-state transition system).

Thus, the lower bound follows from Theorem 4.2.

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 4.2, 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[3] 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 $\{0, 1, \dots, n-1\}$ of vertices. Our encoding will make extensive use of the pairing function introduced in the proof of Theorem 4.2. 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.
- $\forall y.\ y \in E \rightarrow \exists v.\ \exists v'.\ (v < n \wedge v' < n \wedge y = \mathit{pair}(v, v'))$: edges are pairs of vertices.
- $\forall v.\ v < n \rightarrow \exists v'.\ (v' < n \wedge \mathit{pair}(v, v') \in E)$: 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.
- $\forall y.\ y \in \ell \rightarrow \exists v.\ \exists p.\ (v < n \wedge p < |\mathrm{AP}| \wedge y = \mathit{pair}(v, p))$: the labeling of v by the proposition encoded by p is encoded by the pair (v, p). Here, as in the proof of Theorem 4.2, we assume AP to be fixed and therefore can use $|\mathrm{AP}|$ as a constant, and we identify propositions by numbers in $\{0, 1, \dots, |\mathrm{AP}| - 1\}$.

Next, we define $\varphi_{isPath}(P, n, E, I)$, 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 $\varphi_{traceOf}(T, P, \ell)$, expressing that the second-order variable T encodes the trace (using the encoding from the proof of Theorem 4.2) 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

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

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

[3]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.

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

Let us also just remark that the proof of Theorem 4.4 can easily be adapted to show that other natural variations of the satisfiability problem are also polynomial-time equivalent to truth in third-order arithmetic, e.g., satisfiability by countable transition systems, satisfiability by finitely branching transition systems, etc. In fact, as long as a class $\mathcal{C}$ of transition systems is axiomatizable in third-order arithmetic, the Hyper^{2}LTL satisfiability problem restricted to transition systems in $\mathcal{C}$ is reducible to truth in third-order arithmetic. Similarly, truth in third-order arithmetic is reducible to Hyper^{2}LTL satisfiability for every nonempty class of models (w.r.t. standard semantics) and to every class of models containing at least one model whose language contains $T_{allSets} \cup T_{arith}$ (w.r.t. closed-world semantics).

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. [BFFM23] have shown that Hyper^{2}LTL model-checking is Σ_1^1-hard. 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 5.1. *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 4.2 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 4.4, 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 5.2. *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$_{\mathrm{mm}}$

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. [BFFM23] introduced Hyper^{2}LTL$_{\mathrm{mm}}$, a fragment[4] 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). Note that the guards themselves may again contain (guarded) second-order quantifiers. This fragment is expressive enough to express common knowledge, asynchronous hyperproperties, and causality in reactive systems [BFFM23], but it can also reason directly about maximal (or minimal) sets that satisfy some property. As a concrete example, consider the HyperLTL formula

$$\varphi_{gni} = \forall \pi_1. \forall \pi_2. \exists \pi. (\pi_1 =_{\text{high-in}} \pi) \wedge (\pi_2 =_{\text{low-in}} \pi)$$

expressing generalized non-interference [CFK$^+$14]. In case a system does not satisfy it, we might be interested in finding a maximal subset of the system that does satisfy φ_{gni}, while preserving some correctness property φ. We can do so using the formula $\exists(X, \curlywedge, \varphi_{gni}).\ \varphi$ with closed-world semantics (see below the formal definitions).

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

$$\begin{aligned}
\varphi &::= \exists(X, \curlyvee\mkern-14mu\curlywedge, \varphi).\ \varphi \mid \forall(X, \curlyvee\mkern-14mu\curlywedge, \varphi).\ \varphi \mid \exists \pi \in X.\ \varphi \mid \forall \pi \in X.\ \varphi \mid \psi \\
\psi &::= \mathtt{p}_\pi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\, \mathbf{U}\, \psi
\end{aligned}$$

where $\mathtt{p}$ ranges over AP, π ranges over $\mathcal{V}_1$, X ranges over $\mathcal{V}_2$, and $\curlyvee\mkern-14mu\curlywedge \in \{\curlyvee, \curlywedge\}$, i.e., the only modification concerns the syntax of second-order quantification.

We consider two fragments of Hyper^{2}LTL$_{\mathrm{mm}}$ obtained by only allowing quantification over maximal sets and only allowing quantification over minimal sets, respectively:

- Hyper^{2}LTL$_{\mathrm{mm}}^{\curlywedge}$ is the fragment of Hyper^{2}LTL$_{\mathrm{mm}}$ obtained by disallowing second-order quantifiers of the form $\exists(X, \curlyvee, \varphi)$ and $\forall(X, \curlyvee, \varphi)$.
- Hyper^{2}LTL$_{\mathrm{mm}}^{\curlyvee}$ is the fragment of Hyper^{2}LTL$_{\mathrm{mm}}$ obtained by disallowing second-order quantifiers of the form $\exists(X, \curlywedge, \varphi)$ and $\forall(X, \curlywedge, \varphi)$.

The semantics of Hyper^{2}LTL$_{\mathrm{mm}}$ is similar to that of Hyper^{2}LTL but for the second-order quantifiers, for which we define (for $\curlyvee\mkern-14mu\curlywedge \in \{\curlyvee, \curlywedge\}$):

- $\Pi \models \exists(X, \curlyvee\mkern-14mu\curlywedge, \varphi_1).\ \varphi_2$ if there exists a set $T \in \mathrm{sol}(\Pi, (X, \curlyvee\mkern-14mu\curlywedge, \varphi_1))$ such that $\Pi[X \mapsto T] \models \varphi_2$.
- $\Pi \models \forall(X, \curlyvee\mkern-14mu\curlywedge, \varphi_1).\ \varphi_2$ if for all sets $T \in \mathrm{sol}(\Pi, (X, \curlyvee\mkern-14mu\curlywedge, \varphi_1))$ we have $\Pi[X \mapsto T] \models \varphi_2$.

Here, $\mathrm{sol}(\Pi, (X, \curlyvee\mkern-14mu\curlywedge, \varphi_1))$ is the set of all minimal/maximal models of the formula φ_1, which is defined as

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

and

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

Note that $\mathrm{sol}(\Pi, (X, \curlyvee\mkern-14mu\curlywedge, \varphi_1))$ 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

[4] In [BFFM23] this fragment is termed Hyper^{2}LTL$_{\mathrm{fp}}$.

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

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

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

Note that $\mathrm{sol}_{\mathrm{cw}}(\Pi, (X, \curlywedge\!\!\!\!\curlyvee, \varphi_1))$ 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 $\mathrm{Hyper^2LTL_{mm}}$ 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 $\mathrm{Hyper^2LTL}$.

Proposition 6.1 (Proposition 1 of [BFFM23])**.** *Every* $\mathrm{Hyper^2LTL_{mm}}$ *sentence* φ *can be translated in polynomial time (in* $|\varphi|$*) into a* $\mathrm{Hyper^2LTL}$ *sentence* φ' *such that for all sets* T *of traces we have that* $T \models \varphi$ *if and only if* $T \models \varphi'$*.*[5]

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

Remark 6.2. Every $\mathrm{Hyper^2LTL_{mm}}$ sentence φ can be translated in polynomial time (in $|\varphi|$) into a $\mathrm{Hyper^2LTL}$ 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 $\mathrm{Hyper^2LTL}$ also holds for $\mathrm{Hyper^2LTL_{mm}}$ and every lower bound for $\mathrm{Hyper^2LTL_{mm}}$ also holds for $\mathrm{Hyper^2LTL}$.

6.1. **Satisfiability, Finite-State Satisfiability, and Model-Checking.** In this subsection, we settle the complexity of satisfiability, finite-state satisfiability, and model-checking for the fragments $\mathrm{Hyper^2LTL}^{\curlywedge}_{\mathrm{mm}}$ and $\mathrm{Hyper^2LTL}^{\curlyvee}_{\mathrm{mm}}$ using only second-order quantification over maximal respectively minimal sets satisfying a given guard. We will show, for both semantics, that all three problems have the same complexity as the corresponding problems for full $\mathrm{Hyper^2LTL}$, i.e., they are equivalent to truth in third-order arithmetic. Thus, contrary to the design goal of $\mathrm{Hyper^2LTL_{mm}}$, it is in general not more feasible than full $\mathrm{Hyper^2LTL}$.

Due to Proposition 6.1 and Remark 6.2, the upper bounds already hold for full $\mathrm{Hyper^2LTL}$, hence the remainder of this subsection is concerned with lower bounds. We show that one can translate each $\mathrm{Hyper^2LTL}$ sentence φ (over AP) into a $\mathrm{Hyper^2LTL}^{\curlywedge}_{\mathrm{mm}}$ sentence $\varphi^{\curlywedge}$ and into a $\mathrm{Hyper^2LTL}^{\curlyvee}_{\mathrm{mm}}$ sentence $\varphi^{\curlyvee}$ (both over some $\mathrm{AP}' \supsetneq \mathrm{AP}$) and each $T \subseteq (2^{\mathrm{AP}})^\omega$ into a $T' \subseteq (2^{\mathrm{AP}'})^\omega$ such that $T \models_{\mathrm{cw}} \varphi$ if and only if $T' \models_{\mathrm{cw}} \varphi^{\curlywedge}$ and $T \models_{\mathrm{cw}} \varphi$ if and only if $T' \models_{\mathrm{cw}} \varphi^{\curlyvee}$. This translation allows us to reduce $\mathrm{Hyper^2LTL}$ satisfiability, finite-state satisfiability, and model-checking to their $\mathrm{Hyper^2LTL}^{\curlywedge}_{\mathrm{mm}}$ and $\mathrm{Hyper^2LTL}^{\curlyvee}_{\mathrm{mm}}$ counterparts. As closed-world semantics can be reduced to standard semantics, this suffices to prove the result for both semantics.

Intuitively, $\varphi^{\curlywedge}$ and $\varphi^{\curlyvee}$ mimic the quantification over arbitrary sets in φ by quantification over maximal and minimal sets that satisfy a guard φ_1 that is only satisfied by uncountable sets. In the proof of Theorem 3.4, we have constructed a $\mathrm{Hyper^2LTL}$ formula φ_1 with this

[5] The polynomial-time claim is not made in [BFFM23], but follows from the construction when using appropriate data structures for formulas.

property. Here, we show that similar formulas can also be written in $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlywedge}$ and $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlyvee}$. With these formulas as guards (which use fresh propositions in $\mathrm{AP}' \setminus \mathrm{AP}$), we mimic arbitrary set quantification via quantification of sets of traces over AP' that are uncountable (enforced by the guards) and then consider their AP-projections. This approach works for all sets but the empty set, as projecting an uncountable set cannot result in the empty set. For this reason, we additionally mark some traces in the uncountable set and only project the marked ones, but discard the unmarked ones. Thus, by marking no trace, the projection is the empty set.

Theorem 6.3.

(1) $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlywedge}$ *satisfiability, finite-state satisfiability, and model-checking (both under standard semantics and under closed-world semantics) are polynomial-time equivalent to truth in third-order arithmetic. The lower bounds for standard semantics already hold for X_a-free sentences.*
(2) $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlyvee}$ *satisfiability, finite-state satisfiability, and model-checking (both under standard semantics and under closed-world semantics) are polynomial-time equivalent to truth in third-order arithmetic. The lower bounds for standard semantics already hold for X_a-free sentences.*

Recall that we only need to prove the lower bounds, as the upper bounds already hold for full $\mathrm{Hyper}^2\mathrm{LTL}$. Furthermore, as closed-world semantics can be reduced to standard semantics, we only need to prove the lower bounds for closed-world semantics.

We begin by constructing the desired guards that have only uncountable models. As a first step, we modify the formulas constructed in the proof of Theorem 3.4: we show that $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlywedge}$ and $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlyvee}$ have formulas that require the interpretation of a free second-order variable to be uncountable. To this end, fix $\mathrm{AP}_{\mathit{allSets}} = \{+, -, \mathtt{s}, \mathtt{x}\}$ and consider the language

$$
\begin{aligned}
T_{\mathit{allSets}} = {} & \{\{+\}^\omega, \{-\}^\omega\} \cup \\
& \{\{+\}^n\{\mathtt{x}, +\}\{+\}^\omega \mid n \in \mathbb{N}\} \cup \\
& \{\{-\}^n\{\mathtt{x}, -\}\{-\}^\omega \mid n \in \mathbb{N}\} \cup \\
& \{(t(0) \cup \{\mathtt{s}\})(t(1) \cup \{\mathtt{s}\})(t(2) \cup \{\mathtt{s}\}) \cdots \mid t \in (2^{\{\mathtt{x}\}})^\omega\},
\end{aligned}
$$

which is an uncountable subset of $(2^{\mathrm{AP}_{\mathit{allSets}}})^\omega$. Note that this definition differs from the one in the proof of Theorem 3.4, as we have added the two traces $\{+\}^\omega, \{-\}^\omega$. This allows us to construct a finite transition system that has exactly these traces (which is not possible without these two traces): Figure 2 depicts a transition system $\mathfrak{T}_{\mathit{allSets}}$ satisfying $\mathrm{Tr}(\mathfrak{T}_{\mathit{allSets}}) = T_{\mathit{allSets}}$.

Lemma 6.4.

(1) *There exists a* $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlywedge}$ *formula* $\varphi_{\mathit{allSets}}^{\curlywedge}$ *over* $\mathrm{AP}_{\mathit{allSets}}$ *with a single free (second-order) variable Z such that* $\Pi \models_{\mathrm{cw}} \varphi_{\mathit{allSets}}^{\curlywedge}$ *if and only if the* $\mathrm{AP}_{\mathit{allSets}}$*-projection of* $\Pi(Z)$ *is* T_{allSets}.
(2) *There exists a* $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}^{\curlyvee}$ *formula* $\varphi_{\mathit{allSets}}^{\curlyvee}$ *over* $\mathrm{AP}_{\mathit{allSets}}$ *with a single free (second-order) variable Z such that* $\Pi \models_{\mathrm{cw}} \varphi_{\mathit{allSets}}^{\curlyvee}$ *if and only if the* $\mathrm{AP}_{\mathit{allSets}}$*-projection of* $\Pi(Z)$ *is* T_{allSets}.

Proof. 1.) Consider $\varphi_{\mathit{allSets}}^{\curlywedge} = \varphi_0 \wedge \cdots \wedge \varphi_4$ where

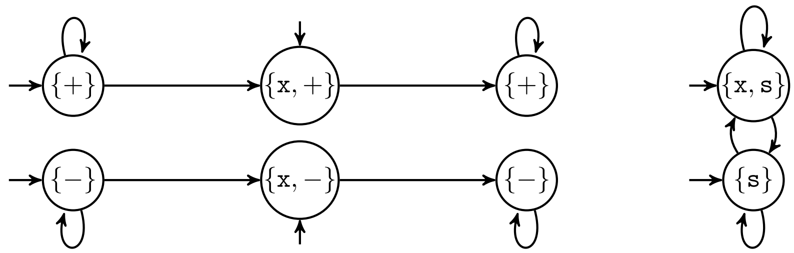


Figure 2: The transition system $\mathfrak{T}_{allSets}$ with $\mathrm{Tr}(\mathfrak{T}_{allSets}) = T_{allSets}$.

- $\varphi_0 = \forall \pi \in Z.\ \bigvee_{\mathtt{p} \in \{+,-,\mathtt{s}\}} \mathbf{G}(\mathtt{p}_\pi \wedge \bigwedge_{\mathtt{p}' \in \{+,-,\mathtt{s}\} \setminus \{\mathtt{p}\}} \neg \mathtt{p}'_\pi)$ expresses that on each trace in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ exactly one of the propositions in $\{+,-,\mathtt{s}\}$ holds at each position and the other two at none. In the following, we speak therefore about type $\mathtt{p}$ traces for $\mathtt{p} \in \{+,-,\mathtt{s}\}$,
- $\varphi_1 = \forall \pi \in Z.\ (+_\pi \vee -_\pi) \rightarrow ((\mathbf{G}\, \neg \mathtt{x}_\pi) \vee (\neg \mathtt{x}_\pi\, \mathbf{U}(\mathtt{x}_\pi \wedge \mathbf{X}\, \mathbf{G}\, \neg \mathtt{x}_\pi)))$ expresses that $\mathtt{x}$ appears at most once on each type $\mathtt{p}$ trace in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$, for both $\mathtt{p} \in \{+,-\}$,
- $\varphi_2 = \bigwedge_{\mathtt{p} \in \{+,-\}} (\exists \pi \in Z.\ \mathtt{p}_\pi \wedge \mathbf{G}\, \neg \mathtt{x}_\pi) \wedge (\exists \pi \in Z.\ \mathtt{p}_\pi \wedge \mathtt{x}_\pi)$ expresses that the type $\mathtt{p}$ traces $\{\mathtt{p}\}^\omega$ and $\{\mathtt{p}\}^0\{\mathtt{x},\mathtt{p}\}\{\mathtt{p}\}^\omega$ are in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$, for both $\mathtt{p} \in \{+,-\}$, and
- $\varphi_3 = \bigwedge_{\mathtt{p} \in \{+,-\}} \forall \pi \in Z.\ \exists \pi' \in Z.\ (\mathtt{p}_\pi \wedge \mathbf{F}\, \mathtt{x}_\pi) \rightarrow (\mathtt{p}_{\pi'} \wedge \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\, \mathtt{x}_{\pi'}))$ expresses that for each type $\mathtt{p}$ trace of the form $\{\mathtt{p}\}^n\{\mathtt{x},\mathtt{p}\}\{\mathtt{p}\}^\omega$ in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$, the trace $\{\mathtt{p}\}^{n+1}\{\mathtt{x},\mathtt{p}\}\{\mathtt{p}\}^\omega$ is also in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$, for both $\mathtt{p} \in \{+,-\}$.

Intuitively, the formulas φ_0 to φ_3 are obtained from the analogous formulas in the proof of Theorem 3.4 by replacing X_d by the free variable Z and by allowing for the two additional traces.

The conjunction of these first four formulas requires that the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ (and thus, under closed-world semantics, also the model) contains at least the traces in

$$\{\{+\}^\omega, \{-\}^\omega\} \cup \{\{+\}^n\{\mathtt{x},+\}\{+\}^\omega \mid n \in \mathbb{N}\} \cup \{\{-\}^n\{\mathtt{x},-\}\{-\}^\omega \mid n \in \mathbb{N}\}, \tag{6.1}$$

and no other type $+$ or type $-$ traces.

We say that a set T of traces is contradiction-free if its $\mathrm{AP}_{allSets}$-projection only contains traces of the form $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ or $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ and if there is no $n \in \mathbb{N}$ such that $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ and $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ are in the $\mathrm{AP}_{allSets}$-projection of T. A trace t is consistent with a contradiction-free T if the following two conditions are satisfied:

(C1): If $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ is in the $\mathrm{AP}_{allSets}$-projection of T then $\mathtt{x} \in t(n)$.
(C2): If $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ is in the $\mathrm{AP}_{allSets}$-projection of T then $\mathtt{x} \notin t(n)$.

Note that T does not necessarily specify the truth value of $\mathtt{x}$ in every position of t, i.e., in those positions $n \in \mathbb{N}$ where neither $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ nor $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ are in the $\mathrm{AP}_{allSets}$-projection of T. Nevertheless, due to (6.1), for every trace t over $\{\mathtt{x}\}$ there exists a contradiction-free subset T of the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ such that the $\{\mathtt{x}\}$-projection of every trace t' that is consistent with T is equal to t. Here, we can, without loss of generality, restrict ourselves to maximal contradiction-free sets, i.e., sets that stop being contradiction-free if more traces are added (note that such sets do specify the truth value of $\mathtt{x}$ for every position of a consistent trace).

Thus, each of the uncountably many traces over $\{\mathtt{x}\}$ is induced by some maximal contradiction-free subset of the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$.

- Hence, we define φ_4 as the formula

$$\forall(X, \curlywedge, \overbrace{\forall\pi \in X.\ (\pi \triangleright Z \wedge (+_\pi \vee -_\pi)) \wedge \forall\pi \in X.\ \forall\pi' \in X.\ (+_\pi \wedge -_{\pi'}) \to \neg\,\mathbf{F}(\mathtt{x}_\pi \wedge \mathtt{x}_{\pi'})}^{X\text{ is contradiction-free}}).$$
$$\exists\pi'' \in Z.\ \forall\pi''' \in X.\ \mathtt{s}_{\pi''} \wedge \underbrace{((+_{\pi'''} \wedge \mathbf{F}\,\mathtt{x}_{\pi'''}) \to \mathbf{F}(\mathtt{x}_{\pi'''} \wedge \mathtt{x}_{\pi''}))}_{\text{(C1)}} \wedge$$
$$\underbrace{((-_{\pi'''} \wedge \mathbf{F}\,\mathtt{x}_{\pi'''}) \to \mathbf{F}(\mathtt{x}_{\pi'''} \wedge \neg\mathtt{x}_{\pi''}))}_{\text{(C2)}},$$

expressing that for every maximal contradiction-free set of traces T, there exists a type $\mathtt{s}$ trace t'' in the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ that is consistent with T.

Thus, if the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ is $T_{allSets}$, then $\Pi \models_{\mathrm{cw}} \varphi^{\curlywedge}_{allSets}$. Dually, we can conclude that $T_{allSets}$ must be a subset of the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ whenever $\Pi \models_{\mathrm{cw}} \varphi^{\curlywedge}_{allSets}$, and that the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ cannot contain other traces (due to φ_0 and φ_1). Hence, $\Pi \models_{\mathrm{cw}} \varphi^{\curlywedge}$ if and only if the $\mathrm{AP}_{allSets}$-projection of $\Pi(Z)$ is $T_{allSets}$.

2.) Here, we will follow a similar approach, but have to overcome one obstacle: there exists a unique minimal contradiction-free set, i.e., the empty set. Hence, we cannot naively replace quantification over maximal contradiction-free sets in φ_4 above by quantification over minimal contradiction-free sets. Instead, we will quantify over minimal contradiction-free sets that have, for each n, either the trace $\{+\}^n\{\mathtt{x},+\}\{+\}^\omega$ or the trace $\{-\}^n\{\mathtt{x},-\}\{-\}^\omega$ in their $\mathrm{AP}_{allSets}$-projection. The minimal sets satisfying this constraint are still *rich* enough to enforce every possible trace over $\{\mathtt{x}\}$.

Formally, we replace φ_4 by φ_4', which is defined as

$$\forall(X, \curlyvee, \varphi_{guard}).$$
$$\exists\pi'' \in Z.\ \forall\pi''' \in X.\ \mathtt{s}_{\pi''} \wedge (+_{\pi'''} \to \mathbf{F}(\mathtt{x}_{\pi'''} \wedge \mathtt{x}_{\pi''})) \wedge (-_{\pi'''} \to \mathbf{F}(\mathtt{x}_{\pi'''} \wedge \neg\mathtt{x}_{\pi''})),$$

where φ_{guard} is the formula

$$\varphi_{complete} \wedge \forall\pi \in X.\ (\pi \triangleright Z \wedge (+_\pi \vee -_\pi)) \wedge \forall\pi \in X.\ \forall\pi' \in X.\ (+_\pi \wedge -_{\pi'}) \to \neg\,\mathbf{F}(\mathtt{x}_\pi \wedge \mathtt{x}_{\pi'})$$

and $\varphi_{complete}$ is the formula

$$\exists\pi \in X.\ (\mathtt{x}_\pi \wedge (+_\pi \vee -_\pi)) \wedge \forall\pi \in X.\ \exists\pi' \in X.\ (+_\pi \vee -_\pi) \to ((+_{\pi'} \vee -_{\pi'}) \wedge \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathtt{x}_{\pi'})),$$

i.e., the only changes are the change of the polarity from $\curlywedge$ to $\curlyvee$ and the addition of $\varphi_{complete}$ in the guard. This ensures that $\varphi^{\curlyvee}_{allSets} = \varphi_0 \wedge \cdots \wedge \varphi_3 \wedge \varphi_4'$ has the desired properties. □

Now, we can begin with the translation of full Hyper^{2}LTL into Hyper^{2}LTL$^{\curlywedge}_{\mathrm{mm}}$ and Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$. Let us fix a Hyper^{2}LTL sentence φ over a set AP of propositions that is, without loss of generality, disjoint from $\mathrm{AP}_{allSets}$. Hence, satisfaction of φ only depends on the projection of traces to AP, i.e., if T_0 and T_1 have the same AP-projection, then $T_0 \models_{\mathrm{cw}} \varphi$ if and only if $T_1 \models_{\mathrm{cw}} \varphi$. We assume without loss of generality that each variable is quantified at most once in φ and that X_a and X_d are not bound by a quantifier in φ, which can always be achieved by renaming variables. Let $X_0, \ldots, X_{k-1}$ be the second-order variables quantified in φ. We can assume $k > 0$, as there is nothing to show otherwise: Without second-order quantifiers, φ is already a Hyper^{2}LTL$^{\curlywedge}_{\mathrm{mm}}$ sentence and a Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$ sentence. To simplify our notation, we define $[k] = \{0, 1, \ldots, k-1\}$. For each $i \in [k]$, we introduce a fresh proposition $\mathtt{m}_i$ so that we can define AP' as the pairwise disjoint union of AP, $\{\mathtt{m}_i \mid i \in [k]\}$, and $\mathrm{AP}_{allSets}$.

Let $i \in [k]$ and consider the formula

$$\varphi^i_{part} = \forall \pi \in X_i.\ ((\mathbf{G}(\mathtt{m}_i)_\pi) \vee (\mathbf{G}\,\neg(\mathtt{m}_i)_\pi) \wedge \bigwedge_{i' \in [k]\setminus\{i\}} \mathbf{G}\,\neg(\mathtt{m}_{i'})_\pi) \wedge$$
$$\forall \pi \in X_i.\ \forall \pi' \in X_i.\ (\pi =_{\mathrm{AP}_{allSets}} \pi') \rightarrow (\pi =_{\mathrm{AP}'} \pi').$$

Note that φ^i_{part} has a single free variable, i.e., X_i, and that it is both a formula of $\mathrm{Hyper}^2\mathrm{LTL}^{\curlyvee}_{\mathrm{mm}}$ and $\mathrm{Hyper}^2\mathrm{LTL}^{\curlywedge}_{\mathrm{mm}}$. Intuitively, φ^i_{part} expresses that each trace in $\Pi(X_i)$ is either marked by $\mathtt{m}_i$ (if $\mathtt{m}_i$ holds at every position) or it is not marked (if $\mathtt{m}_i$ holds at no position), it is not marked by any other $\mathtt{m}_{i'}$, and that there may *not* be two distinct traces $t \neq t'$ in $\Pi(X_i)$ that have the same $\mathrm{AP}_{allSets}$-projection. The former condition means that $\Pi(X_i)$ is partitioned into two (possibly empty) parts, the subset of marked traces and the subset of unmarked traces; the latter condition implies that each trace in $\Pi(X_i)$ is uniquely identified by its $\mathrm{AP}_{allSets}$-projection. Said differently, if two traces in $\Pi(X_i)$ have the same $\mathrm{AP}_{allSets}$-projection, then they also have the same marking.

Fix some $i \in [k]$ and some $\curlyvee\!\!\!\curlywedge \in \{\curlywedge, \curlyvee\}$, and define

$$\varphi^{i,\curlyvee\!\!\!\curlywedge}_{guard} = \varphi^{\curlyvee\!\!\!\curlywedge}_{allSets}[Z/X_i] \wedge \varphi^i_{part},$$

where $\varphi^{\curlyvee\!\!\!\curlywedge}_{allSets}[Z/X_i]$ is the formula obtained from $\varphi^{\curlyvee\!\!\!\curlywedge}_{allSets}$ by replacing each occurrence of Z by X_i. The only free variable of $\varphi^{i,\curlyvee\!\!\!\curlywedge}_{guard}$ is X_i.

The following lemma shows that no strict superset of a set satisfying $\varphi^{i,\curlywedge}_{guard}$ satisfies $\varphi^{i,\curlywedge}_{guard}$ and no strict subset of a set satisfying $\varphi^{i,\curlyvee}_{guard}$ satisfies $\varphi^{i,\curlyvee}_{guard}$, i.e., sets satisfying these guards are in the corresponding solution sets.

Lemma 6.5. *Let Π'_0 and Π'_1 be two variable assignments with $\Pi'_0(X_i) \subseteq (2^{\mathrm{AP}'})^\omega$ and $\Pi'_1(X_i) \subseteq (2^{\mathrm{AP}'})^\omega$.*

(1) *If $\Pi'_0 \models_{\mathrm{cw}} \varphi^{i,\curlywedge}_{guard}$ and $\Pi'_1(X_i) \supsetneq \Pi'_0(X_i)$, then $\Pi'_1 \not\models_{\mathrm{cw}} \varphi^{i,\curlywedge}_{guard}$.*
(2) *If $\Pi'_0 \models_{\mathrm{cw}} \varphi^{i,\curlyvee}_{guard}$ and $\Pi'_1(X_i) \subsetneq \Pi'_0(X_i)$, then $\Pi'_1 \not\models_{\mathrm{cw}} \varphi^{i,\curlyvee}_{guard}$.*

Proof. 1.) Towards a contradiction, assume we have $\Pi'_0 \models_{\mathrm{cw}} \varphi^{i,\curlywedge}_{guard}$ and $\Pi'_1(X_i) \supsetneq \Pi'_0(X_i)$, but also $\Pi'_1 \models_{\mathrm{cw}} \varphi^{i,\curlywedge}_{guard}$. Then, there exists a trace $t_1 \in \Pi'_1(X_i) \setminus \Pi'_0(X_i)$. Due to $\Pi'_0 \models_{\mathrm{cw}} \varphi^{\curlywedge}_{allSets}$ and Lemma 6.4.1, there exists a trace $t_0 \in \Pi'_0(X_i)$ that has the same $\mathrm{AP}_{allSets}$-projection as t_1. Hence, φ^i_{part} implies that t_0 and t_1 have the same AP'-projection, i.e., they are the same trace (here we use $\Pi'_0(X_i) \subseteq (2^{\mathrm{AP}'})^\omega$ and $\Pi'_1(X_i) \subseteq (2^{\mathrm{AP}'})^\omega$). Thus, $t_1 = t_0$ is in $\Pi'_0(X_i)$, i.e., we have derived a contradiction.

2.) The argument here is similar, we just have to swap the roles of the two sets $\Pi'_0(X_i)$ and $\Pi'_1(X_i)$.

Towards a contradiction, assume we have $\Pi'_0 \models_{\mathrm{cw}} \varphi^{i,\curlyvee}_{guard}$ and $\Pi'_1(X_i) \subsetneq \Pi'_0(X_i)$, but also $\Pi'_1 \models_{\mathrm{cw}} \varphi^{i,\curlyvee}_{guard}$. Then, there exists a trace $t_0 \in \Pi'_0(X_i) \setminus \Pi'_1(X_i)$. Due to $\Pi'_1 \models_{\mathrm{cw}} \varphi^{\curlyvee}_{allSets}$ and Lemma 6.4.2, there exists a trace $t_1 \in \Pi'_1(X_i)$ that has the same $\mathrm{AP}_{allSets}$-projection as t_0. Hence, φ^i_{part} implies that t_1 and t_0 have the same AP'-projection, i.e., they are the same trace. Thus, $t_0 = t_1$ is in $\Pi'_1(X_i)$, i.e., we have derived a contradiction. $\square$

Recall that our goal is to show that quantification over minimal/maximal subsets of $(2^{\mathrm{AP}'})^\omega$ satisfying $\varphi^{i,\curlyvee\!\!\!\curlywedge}_{guard}$ mimics quantification over subsets of $(2^{\mathrm{AP}})^\omega$. Now, we can make this statement more formal. Let $T' \subseteq (2^{\mathrm{AP}'})^\omega$. We define $enc_i(T')$ as the set

$$\{t \in (2^{\mathrm{AP}})^\omega \mid t \text{ is the AP-projection of some } t' \in T' \text{ whose } \{\mathtt{m}_i\}\text{-projection is } \{\mathtt{m}_i\}^\omega\}.$$

Now, every $enc_i(T')$ is, by definition, a subset of $(2^{\mathrm{AP}})^\omega$. Our next result shows that, conversely, every subset of $(2^{\mathrm{AP}})^\omega$ can be obtained as an encoding of some T' that additionally satisfies the guard formulas.

Lemma 6.6. *Let $T \subseteq (2^{\mathrm{AP}})^\omega$, $i \in [k]$, and $\curlyvee \in \{\curlywedge, \curlyvee\}$. There exists a $T' \subseteq (2^{\mathrm{AP}'})^\omega$ such that*

- *$T = enc_i(T')$, and*
- *for all Π with $\Pi(X_i) = T'$, we have $\Pi \models_{\mathrm{cw}} \varphi^{i,\curlyvee}_{guard}$.*

Proof. Fix a bijection $f\colon (2^{\mathrm{AP}})^\omega \to T_{allSets}$, which can, e.g., be obtained by applying the Schröder-Bernstein theorem. Now, we define

$$T' = \{t^\frown f(t)^\frown \{\mathtt{m}_i\}^\omega \mid t \in T\} \cup \{t^\frown f(t) \mid t \in (2^{\mathrm{AP}})^\omega \setminus T\}.$$

By definition, we have $enc_i(T') = T$, satisfying the first requirement on T'. Furthermore, the $\mathrm{AP}_{allSets}$-projection of T' is $T_{allSets}$, each trace in T' is either marked by $\mathtt{m}_i$ at every position or at none, the other markers do not appear in traces in T', and, due to f being a bijection, there are no two traces in T' with the same $\mathrm{AP}_{allSets}$-projection. Hence, due to Lemma 6.4, T' does indeed satisfy $\varphi^{i,\curlyvee}_{guard}$. □

Now, we explain how to mimic quantification over arbitrary sets via quantification over maximal or minimal sets satisfying the guards we have constructed above. Let $\curlyvee \in \{\curlyvee, \curlywedge\}$ and let $\varphi^{\curlyvee}$ be the Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$-sentence obtained from φ by inductively replacing

- each $\exists X_i.\ \psi$ by $\exists(X_i, \curlyvee, \varphi^{i,\curlyvee}_{guard}).\ \psi$,
- each $\forall X_i.\ \psi$ by $\forall(X_i, \curlyvee, \varphi^{i,\curlyvee}_{guard}).\ \psi$,
- each $\exists \pi \in X_i.\ \psi$ by $\exists \pi \in X_i.\ (\mathtt{m}_i)_\pi \wedge \psi$, and
- each $\forall \pi \in X_i.\ \psi$ by $\forall \pi \in X_i.\ (\mathtt{m}_i)_\pi \rightarrow \psi$.

We show that φ and $\varphi^{\curlyvee}$ are in some sense equivalent. Obviously, they are not equivalent in the sense that they have the same models, as $\varphi^{\curlyvee}$ uses in general the additional propositions in $\mathrm{AP}' \setminus \mathrm{AP}$, which are not used by φ. But, by *extending* a model of φ we obtain a model of $\varphi^{\curlyvee}$. Similarly, by *ignoring* the additional propositions (i.e., the inverse of the extension) in a model of $\varphi^{\curlyvee}$, we obtain a model of φ.

As the model is captured by the assignment to the variable X_d, this means the interpretation of X_d when evaluating φ and the interpretation of X_d when evaluating $\varphi^{\curlyvee}$ have to satisfy the extension property to be defined. However, we show correctness of our translation by induction. Hence, we need to strengthen the induction hypothesis and require that the variable assignments for φ and $\varphi^{\curlyvee}$ satisfy the extension property for all free variables.

Formally, let Π and Π' be two variable assignments such that

- $\Pi(\pi) \in (2^{\mathrm{AP}})^\omega$ and $\Pi(X) \subseteq (2^{\mathrm{AP}})^\omega$ for all π and X in the domain of Π, and
- $\Pi'(\pi) \in (2^{\mathrm{AP}'})^\omega$ and $\Pi'(X) \subseteq (2^{\mathrm{AP}'})^\omega$ for all π and X in the domain of Π'.

We say that Π' extends Π if they have the same domain (which must contain X_d, but not X_a as this variable may not be used under closed-world semantics) and we have that

- for all π in the domain, the AP-projection of $\Pi'(\pi)$ is $\Pi(\pi)$,
- for all X_i for $i \in [k]$, in the domain, $\Pi(X_i) = enc_i(\Pi'(X_i))$, and

- $\Pi'(X_d) = ext(\Pi(X_d))$ where
$$ext(T) = \{t^\frown t'^\frown\{\mathtt{m}_i\} \mid t \in T, t' \in T_{allSets},\text{ and } i \in [k]\} \cup \{t^\frown t' \mid t \in T \text{ and } t' \in T_{allSets}\},$$
which implies that the AP-projection of $\Pi'(X_d)$ is $\Pi(X_d)$.

Note that if T is a subset of $\Pi(X_d)$, then there exists a subset T' of $\Pi'(X_d)$ such that $enc_i(T') = T$, i.e., $\Pi'(X_d)$ contains enough traces to mimic the quantification of subsets of $\Pi(X_d)$ using our encoding. Furthermore, if $T' \subseteq (2^{\mathrm{AP}'})^\omega$ has the form $T' = ext(T)$ for some $T \subseteq (2^{\mathrm{AP}})^\omega$, then this T is unique.

The following lemma states that our translation of Hyper^{2}LTL into Hyper^{2}LTL$^{\curlywedge}_{\mathrm{mm}}$ and Hyper^{2}LTL$^{\curlyvee}_{\mathrm{mm}}$ is correct.

Lemma 6.7. *Let Π' extend Π. Then, $\Pi \models_{\mathrm{cw}} \varphi$ if and only if $\Pi' \models_{\mathrm{cw}} \varphi^{\text{\ae}}$.*

Proof. By induction over the subformulas ψ of φ.

First, let us consider the case of atomic propositions, i.e., $\psi = \mathtt{p}_\pi$ for some $\mathtt{p} \in \mathrm{AP}$ and some π in the domains of Π and Π'. We have
$$\Pi \models_{\mathrm{cw}} \mathtt{p}_\pi \Leftrightarrow \mathtt{p} \in \Pi(\pi)(0) \Leftrightarrow \mathtt{p} \in \Pi'(\pi)(0) \Leftrightarrow \Pi' \models_{\mathrm{cw}} \mathtt{p}_\pi,$$
as $\Pi(\pi)$ and $\Pi'(\pi)$ have the same AP-projection due to Π' extending Π.

The cases of Boolean and temporal operators are straightforward applications of the induction hypothesis. So, it only remains to consider the quantifiers.

Let $\psi = \exists X_i.\ \psi_0$ for some $i \in [k]$. Then, we have $\varphi^{\text{\ae}} = \exists(X_i, \text{\ae}, \varphi^{i,\text{\ae}}_{guard}).\ \psi_0^{\text{\ae}}$. By induction hypothesis, we have for all $T \subseteq (2^{\mathrm{AP}})^\omega$ and $T' \subseteq (2^{\mathrm{AP}'})^\omega$ with $enc_i(T') = T$, the equivalence
$$\Pi[X_i \mapsto T] \models_{\mathrm{cw}} \psi_0 \Leftrightarrow \Pi'[X_i \mapsto T'] \models_{\mathrm{cw}} \psi_0^{\text{\ae}}.$$
Now, we have
$$\begin{aligned}
&\Pi \models_{\mathrm{cw}} \psi\\
\Leftrightarrow\ &\text{there exists a } T \subseteq \Pi(X_d) \text{ such that } \Pi[X_i \mapsto T] \models_{\mathrm{cw}} \psi_0\\
\overset{*}{\Longleftrightarrow}\ &\text{there exists a } T' \subseteq \Pi'(X_d) \text{ s.t. } \Pi'[X_i \mapsto T'] \models_{\mathrm{cw}} \psi_0^{\text{\ae}} \text{ and } T' \in \mathrm{sol}_{\mathrm{cw}}(\Pi, (X_i, \text{\ae}, \varphi^{i,\text{\ae}}_{guard}))\\
\Leftrightarrow\ &\Pi' \models_{\mathrm{cw}} \psi^{\text{\ae}}.
\end{aligned}$$
Here, the equivalence marked with $*$ is obtained by applying the induction hypothesis:
- For the left-to-right direction, given T, we pick T' such that $enc_i(T') = T$, which is always possible due to Lemma 6.6. Furthermore, T' is in $\mathrm{sol}_{\mathrm{cw}}(\Pi, (X_i, \text{\ae}, \varphi^{i,\text{\ae}}_{guard})$, due to Lemma 6.6 and Lemma 6.5.
- For the right-to-left direction, given T', we pick $T = enc_i(T')$.

Thus, in both directions, $\Pi'[X_i \mapsto T']$ extends $\Pi[X_i \mapsto T]$, i.e., the induction hypothesis is indeed applicable. The argument for universal set quantification is dual.

So, it remains to consider trace quantification. First, let $\psi = \exists\pi \in X_i.\ \psi_0$ for some $i \in [k]$, which implies $\psi^{\text{\ae}} = \exists\pi \in X_i.\ (\mathtt{m}_i)_\pi \wedge \psi'$. Now,
$$\begin{aligned}
&\Pi \models_{\mathrm{cw}} \psi\\
\Leftrightarrow\ &\text{there exists a } t \in \Pi(X_i) \text{ such that } \Pi[\pi \mapsto t] \models_{\mathrm{cw}} \psi_0\\
\overset{*}{\Longleftrightarrow}\ &\text{there exists a } t' \in \Pi'(X_i) \text{ such that } \Pi'[\pi \mapsto t'] \models_{\mathrm{cw}} (\mathtt{m}_i)_\pi \wedge \psi_0^{\text{\ae}}\\
\Leftrightarrow\ &\Pi' \models_{\mathrm{cw}} \psi^{\text{\ae}}.
\end{aligned}$$

Here, the equivalence marked by $*$ follows from the induction hypothesis:

- For the left-to-right direction, given t, we can pick a t' with the desired properties as Π' extends Π, which implies that $\Pi(X_i) = enc_i(\Pi'(X_i))$, which in turn implies that $\Pi'(X_i)$ contains a trace t' marked by $\mathtt{m}_i$ whose AP-projection is t.
- For the right-to-left direction, given t', we pick t as the AP-projection of t', which is in $\Pi(X_i)$, as $\Pi(X_i) = enc_i(\Pi'(X_i))$ and t' is marked by $\mathtt{m}_i$ due to $\Pi'[\pi \mapsto t'] \models_{\mathrm{cw}} (\mathtt{m}_i)_\pi$.

Thus, in both directions, $\Pi'[\pi \mapsto t']$ extends $\Pi[\pi \mapsto t]$, i.e., the induction hypothesis is indeed applicable. The argument for universal trace quantification is again dual.

So, it only remains to consider trace quantification over X_d. Consider $\psi = \exists \pi \in X_d.\ \psi_0$, which implies $\psi^{\chi} = \exists \pi \in X_d.\ \psi_0^{\chi}$. Then, we have

$$
\begin{aligned}
&\Pi \models_{\mathrm{cw}} \psi \\
\Leftrightarrow\ &\text{there exists a } t \in \Pi(X_d) \text{ such that } \Pi[\pi \mapsto t] \models_{\mathrm{cw}} \psi_0 \\
\overset{*}{\Longleftrightarrow}\ &\text{there exists a } t' \in \Pi'(X_d) \text{ such that } \Pi'[\pi \mapsto t'] \models_{\mathrm{cw}} \psi_0^{\chi} \\
\Leftrightarrow\ &\Pi' \models_{\mathrm{cw}} \psi^{\chi},
\end{aligned}
$$

where the equivalence marked with $*$ follows from the induction hypothesis, which is applicable as the AP-projection of $\Pi'(X_d)$ is equal to $\Pi(X_d)$, which implies that we can choose t' from t for the left-to-right direction (and choose t from t' for the right-to-left direction) such that $\Pi'[\pi \mapsto t']$ extends $\Pi[\pi \mapsto t]$. Again, the argument for universal quantification is dual. $\square$

Recall that φ is satisfied by some set $T \subseteq (2^{\mathrm{AP}})^\omega$ of traces (under closed-world semantics) if $\Pi_\emptyset[X_d \mapsto T] \models_{\mathrm{cw}} \varphi$. Similarly, φ^{χ} is satisfied by some set $T' \subseteq (2^{\mathrm{AP}'})^\omega$ of traces (under closed-world semantics) if $\Pi_\emptyset[X_d \mapsto T'] \models_{\mathrm{cw}} \varphi^{\chi}$. The following corollary of Lemma 6.7 holds as $\Pi_\emptyset[X_d \mapsto ext(T)]$ extends $\Pi_\emptyset[X_d \mapsto T]$.

Corollary 6.8. *Let $T \subseteq (2^{\mathrm{AP}})^\omega$, and $\chi \in \{\curlywedge, \curlyvee\}$. Then, $T \models_{\mathrm{cw}} \varphi$ if and only if $ext(T) \models_{\mathrm{cw}} \varphi^{\chi}$.*

So, to conclude the construction, we need to ensure that models of φ^{χ} have the form $ext(T)$ for some $T \subseteq (2^{\mathrm{AP}})^\omega$. For the two satisfiability problems, we do so using a sentence that only has such models, while for the model-checking problem, we can directly transform the transition system we are checking so that it satisfies this property.

First, we construct φ^{χ}_{ext} for $\chi \in \{\curlywedge, \curlyvee\}$ such that $\Pi' \models_{\mathrm{cw}} \varphi^{\chi}_{ext}$ if and only if $\Pi'(X_d) = ext(T)$ for some $T \subseteq (2^{\mathrm{AP}})^\omega$:

$$
\begin{aligned}
\varphi^{\chi}_{ext} = {}& (\forall \pi \in X_d.\ (\mathtt{p}_\pi \wedge \neg \mathtt{p}_\pi)) \vee \\
& \quad \varphi^{\chi}_{allSets}[Z/X_d] \wedge \forall \pi \in X_d.\ \forall \pi' \in X_d. \\
& \quad \left(\bigwedge_{i \in [k]} \exists \pi'' \in X_d.\ \pi =_{\mathrm{AP}_{allSets}} \pi'' \wedge \pi' =_{\mathrm{AP}} \pi'' \wedge \mathbf{G}(\mathtt{m}_i)_{\pi''} \wedge \bigwedge_{i' \in [k] \setminus \{i\}} \mathbf{G}\, \neg (\mathtt{m}_{i'})_{\pi''} \right) \wedge \\
& \quad \left(\exists \pi'' \in X_d.\ \pi =_{\mathrm{AP}_{allSets}} \pi'' \wedge \pi' =_{\mathrm{AP}} \pi'' \wedge \bigwedge_{i \in [k]} \mathbf{G}\, \neg (\mathtt{m}_i)_{\pi''} \right)
\end{aligned}
$$

where $\varphi^{\chi}_{allSets}[Z/X_d]$ is the formula obtained from $\varphi^{\chi}_{allSets}$ by replacing each occurrence of Z by X_d.

The first disjunct of $\varphi^{\curlywedge}_{ext}$ is for the special case of the empty T (where $\mathtt{p}$ is an arbitrary proposition) while the second one expresses intuitively that for each $t \in T_{allSets}$ (bound to π) and each t' in the AP-projection of the interpretation of X_d (bound to π'), the traces $t^\frown t'$ and the traces $t^\frown t'^\frown\{\mathtt{m}_i\}^\omega$, for each $i \in [k]$, are in the interpretation of X_d.

Finally, let us consider the transformation of transition systems: Given a transition system $\mathfrak{T} = (V, E, I, \lambda)$ we construct a transition system $ext(\mathfrak{T})$ satisfying $\mathrm{Tr}(ext(\mathfrak{T})) = ext(\mathrm{Tr}(\mathfrak{T}))$. Recall the transition system $\mathfrak{T}_{allSets}$ depicted in Figure 2, which satisfies $\mathrm{Tr}(\mathfrak{T}_{allSets}) = T_{allSets}$. Let $\mathfrak{T}_{allSets} = (V_a, E_a, I_a, \lambda_a)$. We define $ext(\mathfrak{T}) = (V', E', I', \lambda')$ where

- $V' = V \times V_a \times [k] \times \{0, 1\}$,
- $E' = \{((v, v_a, i, b), (v', v_a', i, b)) \mid (v, v') \in E, (v_a, v_a') \in E_a, i \in [k] \text{ and } b \in \{0, 1\}\}$,
- $I' = I \times I_a \times [k] \times \{0, 1\}$, and
- $\lambda'(v, v_a, i, b) = \begin{cases} \lambda(v) \cup \lambda_a(v_a) \cup \{\mathtt{m}_i\} & \text{if } b = 1, \\ \lambda(v) \cup \lambda_a(v_a) & \text{if } b = 0. \end{cases}$

Note that we indeed have $\mathrm{Tr}(ext(\mathfrak{T})) = ext(\mathrm{Tr}(\mathfrak{T}))$. Hence, if $\Pi(X_d) = \mathrm{Tr}(\mathfrak{T})$ and $\Pi'(X_d) = \mathrm{Tr}(ext(\mathfrak{T}))$, then Π and Π' satisfy the requirement spelled out in Corollary 6.8.

Lemma 6.9. *Let φ be a* Hyper^{2}LTL *sentence and $\varphi^{\curlywedge}$ as defined above (for some $\curlywedge \in \{\curlywedge, \curlyvee\}$), and let $\mathfrak{T}$ be a transition system.*

(1) *φ is satisfiable under closed-world semantics if and only if $\varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext}$ is satisfiable under closed-world semantics.*
(2) *φ is finite-state satisfiable under closed-world semantics if and only if $\varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext}$ is finite-state satisfiable under closed-world semantics.*
(3) *$\mathfrak{T} \models_{\mathrm{cw}} \varphi$ if and only if $ext(\mathfrak{T}) \models_{\mathrm{cw}} \varphi^{\curlywedge}$.*

Proof. 1.) We have

$$\begin{aligned} & \varphi \text{ is satisfiable under closed-world semantics} \\ \Leftrightarrow\ & \text{there exists a } T \subseteq (2^{\mathrm{AP}})^\omega \text{ such that } T \models_{\mathrm{cw}} \varphi \\ \overset{*}{\Longleftrightarrow}\ & \text{there exists a } T' \subseteq (2^{\mathrm{AP}'})^\omega \text{ such that } T' \models_{\mathrm{cw}} \varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext} \\ \Leftrightarrow\ & \varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext} \text{ is satisfiable under closed-world semantics.} \end{aligned}$$

Here, the equivalence marked with $*$ follows from Corollary 6.8:

- For the left-to-right direction, we pick $T' = ext(T)$.
- For the right-to left direction, we pick T to be the unique subset of $(2^{\mathrm{AP}})^\omega$ such that $T' = ext(T)$, which is well-defined as T' satisfies $\varphi^{\curlywedge}_{ext}$.

2.) We have

$$\begin{aligned} & \varphi \text{ is finite-state satisfiable under closed-world semantics} \\ \Leftrightarrow\ & \text{there exists a transition system } \mathfrak{T} \text{ over } 2^{\mathrm{AP}} \text{ such that } \mathrm{Tr}(\mathfrak{T}) \models_{\mathrm{cw}} \varphi \\ \overset{*}{\Longleftrightarrow}\ & \text{there exists a transition system } \mathfrak{T}' \text{ over } 2^{\mathrm{AP}'} \text{ such that } \mathrm{Tr}(\mathfrak{T}') \models_{\mathrm{cw}} \varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext} \\ \Leftrightarrow\ & \varphi^{\curlywedge} \wedge \varphi^{\curlywedge}_{ext} \text{ is finite-state satisfiable under closed-world semantics.} \end{aligned}$$

Here, the equivalence marked with $*$ follows from Corollary 6.8:

- For the left-to-right direction, we pick $\mathfrak{T}' = ext(\mathfrak{T})$, which satisfies $\mathrm{Tr}(ext(\mathfrak{T})) = ext(\mathrm{Tr}(\mathfrak{T}))$.

- For the right-to left direction, we pick $\mathfrak{T}$ to be the transition system obtained from $\mathfrak{T}'$ by removing all propositions in $\mathrm{AP}' \setminus \mathrm{AP}$ from the vertex labels. This implies that $\mathrm{Tr}(\mathfrak{T}')$ is equal to $ext(\mathrm{Tr}(\mathfrak{T}))$, as $\mathrm{Tr}(\mathfrak{T}')$ satisfies $\varphi^{\mathfrak{X}}_{ext}$.

3.) Due to Corollary 6.8 and $\mathrm{Tr}(ext(\mathfrak{T})) = ext(\mathrm{Tr}(\mathfrak{T}))$, we have $\mathfrak{T} \models_{\mathrm{cw}} \varphi$ if and only if $ext(\mathfrak{T}) \models_{\mathrm{cw}} \varphi^{\mathfrak{X}}$. □

Finally, Theorem 6.3 is now a direct consequence of Lemma 6.9, the fact that closed-world semantics can be reduced to standard semantics, and the fact that all three problems for Hyper^{2}LTL (under closed-world semantics) are equivalent to truth in third-order arithmetic [FZ25].

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

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^{2}LTL, and thus as difficult. However, Beutner et al. [BFFM23] note that applications like common knowledge and asynchronous hyperproperties do not even require quantification over smallest/largest sets satisfying arbitrary guards, they "only" require quantification over least fixed points of HyperLTL definable functions. This finally yields a fragment with (considerably) lower complexity: we show that under standard semantics, satisfiability is Σ^2_1-complete while finite-state satisfiability and model-checking are both equivalent to truth in second-order arithmetic. On the other hand, under closed-world semantics, satisfiability is Σ^1_1-complete while the complexity of the other two problems is not affected, i.e., finite-state satisfiability and model-checking are still equivalent to truth in second-order arithmetic. Note that this means that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability under closed-world semantics is as hard as HyperLTL satisfiability.

The intuitive reason that satisfiability under closed-world semantics is simpler is that we show that every satisfiable sentence has a countable model. Thus, not all subsets of natural numbers can be encoded in such a model, which restricts the complexity of the problem. On the other hand, in, e.g., model-checking, the model is part of the input, i.e., one has to potentially evaluate a formula over a set of traces that encodes all subsets of natural numbers. This makes the problem much harder.

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

$$\varphi = \gamma_1.\ Q_1(Y_1, \curlyvee, \varphi^{\mathrm{con}}_1).\ \gamma_2.\ Q_2(Y_2, \curlyvee, \varphi^{\mathrm{con}}_2).\ \dots \gamma_k.\ Q_k(Y_k, \curlyvee, \varphi^{\mathrm{con}}_k).\ \gamma_{k+1}.\ \psi$$

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, we must have $\{X_{\ell_j+1}, \dots, X_{\ell_j}\} \subseteq \{X_a, X_d, Y_1, \dots, Y_{j-1}\}$.
- The free variables of φ^{con}_j are among the trace variables quantified in the $\gamma_{j'}$ with $j' \leq j$ and $X_a, X_d, Y_1, \dots, Y_j$.
- 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-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence[6], if additionally each φ_j^{con} has the form

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

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 $j' \leq j$, as well as X_a and X_d. Then,

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

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

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

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

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

which is the least fixed point of $f_{\Pi,j}$. Due to the minimality constraint on Y_j in φ, $\mathrm{lfp}(\Pi, j)$ is the unique set in $\mathrm{sol}(\Pi, (Y_j, \curlyvee, \varphi_j^{\mathrm{con}}))$. Hence, an induction shows that $\mathrm{lfp}(\Pi, j)$ 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 $\mathrm{sol}(\Pi, (Y_j, \curlyvee, \varphi_j^{\mathrm{con}}))$ 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 7.1. 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$_{\mathrm{mm}}$ is to restrict oneself to X_a-free sentences.

In the remainder of this section, we study the complexity of lfp-Hyper^{2}LTL$_{\mathrm{mm}}$.

[6]Our definition here differs slightly from the one of [BFFM23] 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 [BFFM23] are also of this form.

7.1. **Satisfiability for the X_a-free Fragment.** For satisfiability of X_a-free sentences, the key step is again to study the size of models of satisfiable sentences. For X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$, 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 [FZ17] 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 7.2. *Every satisfiable X_a-free* lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *sentence has a countable model.*

A crucial step to prove this upper bound is to characterize membership of a trace in some least fixed point by the existence of a finite trace-labeled tree. These trees model the stage-wise definition of the semantics of second-order quantification in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$.

To formalize this intuition, we need to introduce some additional notation and concepts. Let

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

be an X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence. We assume without loss of generality that each trace variable is quantified at most once in φ, which can always be achieved by renaming variables. This implies that for each trace variable π quantified in some γ_j, there is a unique second-order variable X_π such that π ranges over X_π. Furthermore, we assume that each Y_j is different from X_d and X_a, which can again be achieved by renaming variables, if necessary.

Recall that set quantifiers in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ "range" over unique least fixed points that do not depend on the choices for the other set variables, but only on the choice for the universe of discourse and the choices for the trace variables quantified before the set variable.

We say that a variable assignment Π is (φ, j)-sufficient, for some $j \in \{1, \ldots, k\}$, if its domain contains $X_d, Y_1, \ldots, Y_{j-1}$, if its domain contains the variables quantified in the $\gamma_{j'}$ with $j' \leq j$, and if $\Pi(\pi) \in \Pi(X_\pi)$ for all π quantified in some $\gamma_{j'}$ with $j' \leq j$. A (φ, j)-sufficient assignment Π contains all information necessary to compute $\mathrm{lfp}(\Pi, j)$. Note that we do not have any requirements on the $\Pi(Y_{j'})$ being the right sets, i.e., being the least fixed points. This will be taken into account later.

Our first results states that the computation of least fixed points is monotonic in the model we are starting with (given by the interpretation of X_d) and the values for the previously quantified $Y_{j'}$.

Lemma 7.3. *Let Π and Π' be (φ, j)-sufficient assignments with $\Pi'(X_d) \subseteq \Pi(X_d)$, $\Pi'(Y_{j'}) \subseteq \Pi(Y_{j'})$ for all $1 \leq j' < j$, and $\Pi(\pi) = \Pi'(\pi)$ for all trace variables appearing in the $\gamma_{j'}$ with $1 \leq j' \leq j$. Then, $\mathrm{lfp}(\Pi', j) \subseteq \mathrm{lfp}(\Pi, j)$.*

Proof. An induction shows that $S'_\ell \subseteq S_\ell$, where $\mathrm{lfp}(\Pi, j) = \bigcup_{\ell \geq 0} S_\ell$ and $\mathrm{lfp}(\Pi', j) = \bigcup_{\ell \geq 0} S'_\ell$ are the stages of the fixed points. □

Now, let Π be a variable assignment whose domain contains at least X_d and all trace variables in the γ_j for $1 \leq j \leq k$. We define $\Pi_0 = \Pi$ and $\Pi_j = \Pi_{j-1}[Y_j \mapsto \mathrm{lfp}(\Pi_{j-1}, j)]$, i.e., we iteratively compute and assign the least fixed points to the set variables in φ. Note that fixed points are uniquely determined by just the values $\Pi(X_d)$ and $\Pi(\pi)$ for π in the γ_j with $1 \leq j \leq k$, but we still need the previous fixed points to determine the j-th one. Now, we define $e(\Pi) = \Pi_k$, which coincides with Π on all variables but the Y_j, which are mapped to the least fixed points induced by Π.

Remark 7.4. Note that each Π_{j-1} with $1 \le j \le k$ is (φ, j)-sufficient. Hence, if we have variable assignments Π and Π' whose domains contain at least X_d and all trace variables in the γ_j for $1 \le j \le k$ and such that $\Pi'(X_d) \subseteq \Pi(X_d)$ and $\Pi(\pi) = \Pi'(\pi)$ for all trace variables appearing in the γ_j with $1 \le j \le k$, then Lemma 7.3 implies that we have $e(\Pi')(Y_j) \subseteq e(\Pi)(Y_j)$ for all j.

Next, we define a tree structure that witnesses the membership of traces in least fixed points assigned to the variables Y_j, which make the inductive construction of the stages of the least fixed points explicit. Intuitively, consider the formula

$$\varphi_j^{\mathrm{con}} = \dot\pi_1 \triangleright Y_j \wedge \cdots \wedge \dot\pi_n \triangleright Y_j \wedge \forall \ddot\pi_1 \in Z_1. \ \ldots \forall \ddot\pi_{n'} \in Z_{n'}. \ \psi_j^{\mathrm{step}} \rightarrow \ddot\pi_m \triangleright Y_j. \tag{7.1}$$

inducing the unique least fixed point 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 a witness. Thus, a witness is in general a tree labeled by traces and indices from $\{1, \ldots, k\}$ denoting the fixed point variable the tree proves membership in. Note that ψ_j^{step} may contain free variables that are quantified in the $\gamma_{j'}$ for $1 \le j' \le j$. The membership of such variables π in $\Pi(X_\pi)$ will not be witnessed by this witness tree, as their values are already determined by Π. Instead, they will be taken care of by their own witness trees.

Formally, let us fix a $j^* \in \{1, \ldots, k\}$, a (φ, j^*)-sufficient assignment Π, and a trace t^*. A Π-witness tree for (t^*, j^*) (which witnesses $t^* \in \mathrm{lfp}(\Pi, j^*)$) is an ordered finite tree whose vertices are labeled by pairs (t, j) where t is a trace from $\Pi(X_d)$ and where j is in $\{1, \ldots, k\} \cup \{\bot\}$ such that the following are satisfied:

- The root is labeled by (t^*, j^*).
- If a vertex is labeled with $(t, \bot)$ for some trace t, then it must be a leaf and $t \in \Pi(X_d)$.
- Let (t, j) be the label of some vertex v with $j \in \{1, \ldots, k\}$ and let φ_j^{con} as in (7.1). If v is a leaf, then we must have $t = \Pi(\dot\pi_i)$ for some $i \in \{1, \ldots, n\}$. If v is an internal vertex, then it must have n' successors labeled by $(t_1, j_1), \ldots, (t_{n'}, j_{n'})$ (in that order) such that $\Pi[\ddot\pi_1 \mapsto t_1, \ldots, \ddot\pi_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}}$, $t = t_m$, and such that the following holds for all $i \in \{1, \ldots, n'\}$:
  - If $Z_i = X_d$, then the i-th successor of v is a leaf, $t_i \in \Pi(X_d)$, and $j_i = \bot$.
  - If $Z_i = Y_{j'}$ for some j' (which satisfies $1 \le j' \le j$ by definition of lfp-Hyper^{2}LTL$_{\mathrm{mm}}$), then we must have that $j_i = j'$ and that the subtree rooted at the i-th successor of v is a Π-witness tree for (t_i, j_i).

Given a Π-witness tree B, let $\mathrm{Tr}(B)$ denote the set of traces labeling the vertices of B, which is a finite subset of $\Pi(X_d)$.

Lemma 7.5. *Let Π be a (φ, j)-sufficient assignment with $\Pi(Y_{j'}) = e(\Pi)(Y_{j'})$ for all $j' < j$.*

(1) *We have $t \in \mathrm{lfp}(\Pi, j)$ if and only if there is a Π-witness tree for (t, j).*
(2) *Let Π' be another (φ, j)-sufficient assignment with $\Pi'(Y_{j'}) = e(\Pi')(Y_{j'})$ for all $j' < j$ and with $\Pi(\pi) = \Pi'(\pi)$ for all trace variables appearing in the $\gamma_{j'}$ with $j' \le j$. If $\Pi'(X_d) \supseteq \mathrm{Tr}(B)$ for a Π-witness tree B for (t, j), then B is also a Π'-witness tree for (t, j).*

Proof. 1. Let $\mathrm{lfp}(\Pi, j) = \bigcup_{\ell \geq 0} S_\ell$. The direction from left to right is proven by an induction over (j, ℓ) (with lexicographic ordering) constructing for each trace t in $\mathrm{lfp}(\Pi, j)$ a Π-witness tree for (t, j). So, let $t \in \mathrm{lfp}(\Pi, j)$. As we have $S_0 = \emptyset$, there is a minimal ℓ such that $t \in S_\ell \setminus S_{\ell-1}$, which satisfies $\ell \geq 1$.

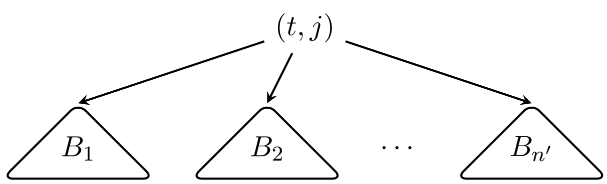
If $\ell = 1$, then there are two cases how t was added to S_1:

- Either we have $t = \Pi(\dot{\pi}_i)$ for some $1 \leq i \leq n$. Then, a tree consisting of a single vertex labeled by (t, j) is a Π-witness tree for (t, j).
- Otherwise, we must have $Z_i \neq Y_j$ for all $1 \leq i \leq n'$ and there must be traces $t_1, \ldots, t_{n'}$ such that each t_i is in $\Pi(Z_i)$, $t = t_m$, and

$$(((\Pi[\ddot{\pi}_1 \mapsto t_1])[\ddot{\pi}_2 \mapsto t_2]) \cdots)[\ddot{\pi}_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}},$$

  i.e., we update all variables of the form $\ddot{\pi}_{j'}$ with $t_{j'}$ in Π before evaluating ψ_j^{step}.

  If $Z_i = X_d$, then let B_i be a tree consisting of a single vertex labeled by $(t_i, \perp)$, if $Z_i = Y_{j'}$ for some $j' < j$ (this is the only other possibility), then let B_i be a Π-witness tree for (t_i, j'), which exists by induction hypothesis. Then,

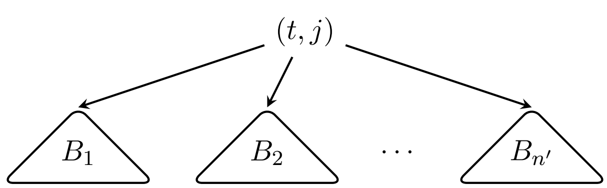


  is a Π-witness tree for (t, j).

Now, consider the case $\ell > 1$. Here, we must have that there is at least one $1 \leq i \leq n'$ such that $Z_i = Y_j$. Again, there must be traces $t_1, \ldots, t_{n'}$ such that each t_i is in $\Pi(Z_i)$ if $Z_i \neq Y_j$, $t_i \in S_{\ell-1}$ if $Z_i = Y_j$, $t = t_m$, and

$$(((\Pi[\ddot{\pi}_1 \mapsto t_1])[\ddot{\pi}_2 \mapsto t_2]) \cdots)[\ddot{\pi}_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}}$$

For i with $Z_i = X_d$ or $Z_i = Y_{j'}$ with $j' < j$, we define B_i as in the second case of the induction start. For $Z_i = Y_j$ we define B_i to be a Π-witness tree for (t_i, j), which exists by induction hypothesis applied to $(j, \ell - 1)$. Then,



is a Π-witness tree for (t, j).

The direction from right to left is proven by an induction over (j, h) (with lexicographic ordering) showing that for each height h Π-witness tree for some (t, j) we have $t \in S_h \subseteq \mathrm{lfp}(\Pi, j)$, where $\mathrm{lfp}(\Pi, j)$ is again $\bigcup_{\ell \geq 0} S_\ell$. Here, we measure the height of a tree whose root is labeled by (t, j) as the length (in the number of vertices) of a longest path starting at the root that visits only vertices that have a j in the second component of their label.

So, let B be a Π-witness tree for some (t, j). If B has height 1, then there are two cases:

- Either, $t = \Pi(\dot{\pi}_i)$ for some $1 \leq i \leq n$. Then, $t \in S_1 \subseteq \mathrm{lfp}(\pi, j)$.

- Otherwise, we must have $Z_i \neq Y_j$ for all $1 \leq i \leq n'$ and there must be traces $t_1, \ldots, t_{n'}$ such that each t_i is in $\Pi(Z_i)$, $t = t_m$, and
$$(((\Pi[\ddot{\pi}_1 \mapsto t_1])[\ddot{\pi}_2 \mapsto t_2]) \cdots)[\ddot{\pi}_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}}.$$
Let B_i be the subtree of B rooted at the i-th successor of the root, which is labeled by (t_i, j') if $Z_i = Y_{j'}$ for some $j' < j$ or by $(t_i, \perp)$, if $Z_i = X_d$. In the latter case, we have $t_i \in \Pi(X_d)$ by the requirement that a Π-witness tree is only labeled by traces from $\Pi(X_d)$. In the former case, B_i is by definition a Π-witness tree for (t_i, j'). Hence, the induction hypothesis yields $t_i \in \mathrm{lfp}(\Pi, j') = \Pi(Y_{j'})$.
Thus, we can apply the definition of $S_1 = f_{\Pi,j}(\emptyset)$ and obtain $t \in S_1 = S_h$.

Now, assume B has height $h > 1$. Then, there must be at least one $1 \leq i \leq n'$ such that $Z_i = Y_j$. Again, there must be traces $t_1, \ldots, t_{n'}$ such that each t_i is in $\Pi(Z_i)$ if $Z_i \neq Y_j$, $t_i \in S_{\ell-1}$ if $Z_i = Y_j$, $t = t_m$, and
$$(((\Pi[\ddot{\pi}_1 \mapsto t_1])[\ddot{\pi}_2 \mapsto t_2]) \cdots)[\ddot{\pi}_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}}.$$
For i with $Z_i = Y_{j'}$ for some $j < j'$ and for i with $Z_i = X_d$ we can argue as in the second case of the induction start that we have $t_i \in \Pi(Z_i)$. So, consider the remaining i, which satisfy $Z_i = Y_j$. By definition, the subtree B_i rooted at the i-th successor of the root is a Π-witness tree for (t_i, j) of height at most $h - 1$. Hence, the induction hypothesis applied to $(j, h-1)$ yields $t_i \in S_{h-1}$. Thus, we can apply the definition of $S_h = f_{\Pi,j}(S_{h-1})$ and obtain $t \in S_h$.

2. This can be shown by induction over the height of witness trees exploiting the fact that B being a Π^*-witness tree only depends on $\Pi^*(X_d)$ containing $\mathrm{Tr}(B)$ and the values $\Pi^*(\pi)$ for trace variables appearing in the $\gamma_{j'}$ with $j' \leq j$. □

Now we have collected all the prerequisites to prove Lemma 7.2 which states that every satisfiable X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$-sentence has a countable model.

Proof of Lemma 7.2. Let
$$\varphi = \gamma_1.\ Q_1(Y_1, \curlyvee, \varphi_1^{\mathrm{con}}).\ \gamma_2.\ Q_2(Y_2, \curlyvee, \varphi_2^{\mathrm{con}}).\ \ldots \gamma_k.\ Q_k(Y_k, \curlyvee, \varphi_k^{\mathrm{con}}).\ \gamma_{k+1}.\ \psi$$
be a satisfiable X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence. Again, we assume without loss of generality that each trace variable is quantified at most once in φ, which implies that for each trace variable π quantified in some γ_j, there is a unique second-order variable X_π such that π ranges over X_π. Furthermore, we assume that each Y_j is different from X_d and X_a, which can again be achieved by renaming variables, if necessary.

As φ is satisfiable, there exists a set T of traces such that $T \models \varphi$. In the following, we assume that T is uncountable (and therefore nonempty), as the desired result otherwise holds trivially. We prove 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.

So, let E be the set of existentially quantified trace variables in the γ_j and let, for each $\pi \in E$, k_π denote the number of trace variables that are universally quantified before π (in any block γ_j). As T satisfies φ, there is a Skolem function $f_\pi \colon T^{k_\pi} \to T$ for each $\pi \in E$, i.e., functions witnessing $T \models \varphi$ in the following sense: $\Pi \models \psi$ for every variable assignment Π where

- X_d is mapped to T,

- each second-order variable Y_j is mapped to $e(\Pi)(Y_j)$,
- for each universally quantified trace variable π that appears in a γ_j we have $\Pi(\pi) \in \Pi(X_\pi)$, and
- for each existentially quantified trace variable π that appears in a γ_j, we have $\Pi(\pi) = f_\pi(\Pi(\pi_1), \ldots, \Pi(\pi_{k_\pi}))$, which has to be in $\Pi(X_\pi)$. Here, $\pi_1, \ldots, \pi_{k_\pi}$ are the universally quantified trace variables that appear in some γ_j before π.

We fix such Skolem functions for the rest of the proof.

Given a set $T' \subseteq T$, we define

$$f_\pi(T') = \{f_\pi(t_1, \ldots, t_{k_\pi}) \mid t_1, \ldots, t_{k_\pi} \in T'\}.$$

If T' is finite, then $f_\pi(T')$ is also finite.

For each (φ, j)-sufficient assignment Π with $\Pi(X_d) = T$ and each $t \in \mathrm{lfp}(\Pi, j)$, we fix a Π-witness tree $B(\Pi, t, j)$ for (t, j).

Now, we fix some arbitrary trace $t_0 \in T$ and define $R_0 = \{t_0\}$,

$$R_{\ell+1} = R_\ell \cup \bigcup_{\pi \in E} f_\pi(R_\ell)$$

for even $\ell \geq 0$ as well as

$$R_{\ell+1} = R_\ell \cup \bigcup_{\Pi} \bigcup_{j=1}^{k} \bigcup_{t \in \Pi(Y_j) \cap R_\ell} \mathrm{Tr}(B(\Pi, t, j)),$$

for odd $\ell \geq 0$, where Π ranges over all assignments that satisfy the following requirements:

(1) The domain of Π contains exactly X_d, the set variables $Y_1, \ldots, Y_k$ appearing in φ, and the trace variables quantified in the γ_j.
(2) $\Pi(X_d) = T$.
(3) $\Pi(Y_j) = e(\Pi)(Y_j)$ for all Y_j.
(4) $\Pi(\pi) \in \Pi(X_\pi) \cap R_\ell$ for all π that are universally quantified in some γ_j.
(5) $\Pi(\pi) = f_\pi(\Pi(\pi_1), \ldots, \Pi(\pi_{k_\pi})) \in \Pi(X_\pi)$, where $\pi_1, \ldots, \pi_{k_\pi}$ are again the universally quantified trace variables that appear in some γ_j before π.

We call a Π with this property admissible at ℓ.

We first show that each R_ℓ is a finite subset of T. This is obviously true for $\ell = 0$ and if it is true for even ℓ, then it is also true for $\ell + 1$, as applying the finitely many Skolem functions to tuples of traces from R_ℓ (of which there are, by induction hypothesis, only finitely many) adds only finitely many traces from T. So, let us consider an odd ℓ and consider the Π that are admissible at ℓ: The values $\Pi(X_d)$, $\Pi(Y_j)$ for all j, and $\Pi(\pi)$ for π existentially quantified in some γ_j are all determined, so the only degree of freedom are the values $\Pi(\pi)$ for π universally quantified in some γ_j. However, as these must come from R_ℓ, which is finite by induction hypothesis, there are only finitely many Π that are admissible at ℓ. Now, we can conclude that $R_{\ell+1}$ is obtained by adding the traces of finitely many witness trees to R_ℓ, each of which contains finitely many traces from T. Thus, R_ℓ is a finite subset of T as well.

Now, define $R = \bigcup_{\ell \geq 0} R_\ell$, which is a countable subset of T, as it is a countable union of finite subsets of T. Recall that the Skolem functions f_π above witness $T \models \varphi$. We show that they also witness $R \models \varphi$, which completes the proof.

So, let Π' be a variable assignment such that

- X_d is mapped to R,

- each second-order variable Y_j is mapped to $e(\Pi')(Y_j)$,
- for each universally quantified trace variable π that appears in a γ_j we have $\Pi'(\pi) \in \Pi'(X_\pi)$, and
- for each existentially quantified trace variable π that appears in a γ_j, we have $\Pi'(\pi) = f_\pi(\Pi'(\pi_1), \ldots, \Pi'(\pi_{k_\pi}))$, where $\pi_1, \ldots, \pi_{k_\pi}$ are the universally quantified trace variables that appear in some γ_j before π. Note that we do not claim or require (yet) that $\Pi'(\pi)$ is in $\Pi'(X_\pi)$.

We show that $\Pi' \models \psi$ and that $\Pi'(\pi)$ is indeed in $\Pi'(X_\pi)$. If this is true for every such Π', then we can conclude that the Skolem functions witness $R \models \varphi$.

To this end, let the variable assignment Π be obtained by defining

- $\Pi(X_d) = T$,
- $\Pi(Y_j) = e(\Pi)(Y_j)$,
- $\Pi(\pi) = \Pi'(\pi)$ for all trace variables π that appear in some γ_j, and
- Π is undefined for all other variables.

Due to Remark 7.4, we have $\Pi(Y_j) \supseteq \Pi'(Y_j)$. Hence, in Π, each universal variable π is mapped to a trace in $\Pi(X_\pi)$: If $X_\pi = X_d$, then we have

$$\Pi(\pi) = \Pi'(\pi) \in \Pi'(X_\pi) = \Pi'(X_d) = R \subseteq T = \Pi(X_d) = \Pi(X_\pi),$$

and if $X_\pi = Y_j$, then we have

$$\Pi(\pi) = \Pi'(\pi) \in \Pi'(X_\pi) = \Pi'(Y_j) \subseteq \Pi(Y_j) = \Pi(X_\pi).$$

Thus, as the traces for existentially quantified variables are obtained via the Skolem functions witnessing $T \models \varphi$, we conclude $\Pi \models \psi$. Now, as ψ is quantifier-free, $\Pi \models \psi$ only depends on the restriction of Π to trace variables. As Π and Π' coincide on the these variables, we have $\Pi' \models \psi$.

This almost concludes the proof, we just have to argue that we indeed have that $\Pi'(\pi) = f_\pi(\Pi'(\pi_1), \ldots, \Pi'(\pi_{k_\pi}))$ is in $\Pi'(X_\pi)$. Note that the traces $\Pi(\pi)$ for universally quantified π's are all in R and thus there is an even ℓ^* such that all the traces $\Pi'(\pi)$ for all such π are in R_{ℓ^*}. This is in particular true for the traces $\Pi'(\pi_1), \ldots, \Pi'(\pi_{k_\pi})$. We now consider two cases.

If $X_\pi = X_d$, then we have

$$\Pi'(\pi) \in R_{\ell^*+1} \subseteq R = \Pi'(X_d) = \Pi'(X_\pi)$$

by definition, as R_{ℓ^*+1} contains $f_\pi(R_{\ell^*})$, which contains $f_\pi(\Pi'(\pi_1), \ldots, \Pi'(\pi_{k_\pi}))$, as all these are universally quantified traces.

For the case $X_\pi \neq X_d$ we need some more preparation. As just argued, we have $\Pi'(\pi) = \Pi(\pi) \in R_{\ell^*+1}$. We show that Π as defined above is admissible at $\ell^* + 1$ by verifying that all five requirements are satisfied:

(1) The domain requirement is satisfied by definition.
(2) We have $\Pi(X_d) = T$ by definition.
(3) By definition, we have $\Pi(Y_{j'}) = e(\Pi)(Y_{j'})$ for all $Y_{j'}$.
(4) We have $\Pi(\pi) \in \Pi(X_\pi)$ for all universally quantified π as argued above and we have $\Pi(\pi) \in R_{\ell^*}$ by the choice of ℓ^*.
(5) For existentially quantified π, we have $\Pi(\pi) = f_\pi(\Pi'(\pi_1), \ldots, \Pi'(\pi_{k_\pi}))$ by definition, which is furthermore in $\Pi(X_\pi)$, as f_π is a Skolem function.

Thus, all requirements on Π for being admissible at $\ell^* + 1$ are satisfied.

Hence, we can conclude that the traces of a Π-witness tree for $(\Pi(\pi), j_\pi)$ are contained in $R_{\ell^*+2} \subseteq R$ for each existentially quantified π, where j_π is defined such that $X_\pi = Y_{j_\pi}$. So, there is a Π-witness tree B for $(\Pi(\pi), j_\pi)$ for each such π, as R is a subset of $T = \Pi(X_d)$. Thus, Lemma 7.5.2 yields that B is also a Π'-witness tree for $(\Pi(\pi), j_\pi) = (\Pi'(\pi), j_\pi)$ for each such π. Hence, Lemma 7.5.1 yields that $\Pi'(\pi) \in \mathrm{lfp}(\Pi', j_\pi) = \Pi'(Y_{j_\pi})$ for all such π. $\square$

Before we continue with our complexity results, let us briefly mention that the formula from Remark 3.2 on Page 9 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 (without loss of generality) countable models of a given X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ 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) witnessing that T satisfies φ. This places satisfiability in Σ_1^1 while the matching lower bound already holds for HyperLTL [FKTZ21].

Theorem 7.6. lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *satisfiability for X_a-free sentences is Σ_1^1-complete.*

Proof. The Σ_1^1 lower bound already holds for HyperLTL satisfiability [FKTZ21], which is a fragment of X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ (see Remark 2.1). Hence, we focus in the following on the upper bound, which is a generalization of the corresponding upper bound for HyperLTL [FKTZ21].

Throughout this proof, we fix an X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence

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

where ψ is quantifier-free, and let Φ denote the set of quantifier-free subformulas of φ (including those in the guards). As before, we assume without loss of generality that each trace variable is quantified at most once in φ, i.e., 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.

Due to Lemma 7.2, φ is satisfiable if and only if it has a countable model T. Thus, not only is T countable, but the second-order quantifiers range over subsets of T, i.e., over countable sets. Finally, recall that the least fixed point assigned to Y_j depends on the variable assignment to trace variables in the blocks $\gamma_1, \ldots, \gamma_j$, but not on the second-order variables $Y_{j'}$ with $j' < j$, as their values are also uniquely determined by the variables in the blocks $\gamma_1, \ldots, \gamma_{j'}$.

To prove that the lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability problem for X_a-free sentences is in Σ_1^1, we express, for a given X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ (encoded as a natural number), the existence of a countable set T of traces and a witness that T is indeed a model of φ. As we work with second-order arithmetic we can quantify only over natural numbers (type 0 objects) and sets of natural numbers (type 1 objects). To simplify our notation, we note that there is a bijection between finite sequences over $\mathbb{N}$ and $\mathbb{N}$ itself and that one can encode functions mapping natural numbers to natural numbers as sets of natural numbers via their graphs. As both can be implemented in arithmetic, we will freely use vectors of natural numbers, functions mapping natural numbers to natural numbers, and combinations of both.

Furthermore, as we only have to work with countable sets of traces, we can use natural numbers to "name" the traces. Hence, the restriction of a variable assignment to trace variables can be encoded by a vector of trace names, listing (in some fixed order), the names of the traces assigned to the trace variables. Finally, the countable sets assigned to the Y_j

depend on the variable assignment to the trace variables quantified before Y_j, but not on the sets assigned to the $Y_{j'}$ with $j' < j$. Thus we do not have to consider second-order variables in variable assignments: in the following, when we speak of a variable assignment we only consider those that are undefined for all second-order variables. Let k' denote the number of trace variables in φ. Then, variable assignments are encoded by vectors in $\mathbb{N}^{k'}$.

First, we show how to capture the semantics of quantifier-free formulas in arithmetic. Recall that ψ is the maximal quantifier-free subformula of ψ and that Φ denotes the set of quantifier-free subformulas of φ (including those in the guards). Let Π be a variable assignment whose domain contains all the trace variables of φ, and therefore in particular all free variables of ψ. Then, $\Pi \models \psi$ if and only if there is a function $e\colon \Phi \times \mathbb{N} \to \{0,1\}$ with $e(\psi, 0) = 1$ satisfying the following consistency conditions:

- $e(a_\pi, j) = 1$ if and only if $a \in \Pi(\pi)(j)$.
- $e(\neg\psi_1, j) = 1$ if and only if $e(\psi_1, j) = 0$.
- $e(\psi_1 \vee \psi_2, j) = 1$ if and only if $e(\psi_1, j) = 1$ or $e(\psi_2, j) = 1$.
- $e(\mathbf{X}\,\psi_1, j) = 1$ if and only if $e(\psi_1, j+1) = 1$.
- $e(\psi_1 \mathbf{U}\,\psi_2, j) = 1$ if and only if there is a $j' \geq j$ such that $e(\psi_2, j') = 1$ and $e(\psi_1, j'') = 1$ for all j'' in the range $j \leq j'' < j'$.

In fact, there is exactly one function e satisfying these consistency conditions, namely the function $e_{\varphi,\Pi}$ defined as

$$e_{\varphi,\Pi}(\psi', j) = \begin{cases} 1 & \text{if } \Pi[j,\infty) \models \psi', \\ 0 & \text{otherwise.} \end{cases}$$

Now, given φ we express the existence of the following type 1 objects:

- A countable set of traces over the propositions of φ encoded as a function T from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, mapping trace names (i.e., natural numbers) and positions to (encodings of) subsets of the set of propositions appearing in φ.
- For each $j \in \{1, 2, \ldots, k\}$ a function T_j from $\mathbb{N}^{k'} \times \mathbb{N}$ to $\{0,1\} \times \mathbb{N}$ mapping a variable assignment $\overline{a}$ and a trace name n to a pair (b, ℓ) where the bit b encodes whether the trace named n is in the set assigned to Y_j (w.r.t. the variable assignment encoded by $\overline{a}$) and where the natural number ℓ is intended to encode in which stage of the fixed point computation the trace named n was added to the fixed point (computed with respect to $\overline{a}$). That this is correct will be captured by the formula we are constructing.
- A function Sk from $\mathbb{N} \times \mathbb{N}^*$ to $\mathbb{N}$ to be interpreted as Skolem functions for the existentially quantified trace variables of φ, i.e., we map a variable name and a variable assignment of the variables preceding it to a trace name.
- A function E from $\mathbb{N}^{k'} \times \mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, where, for a fixed $\overline{a} \in \mathbb{N}^{k'}$ encoding a variable assignment Π, the function $x, j \mapsto E(\overline{a}, x, j)$ is intended to encode the function $e_{\varphi,\Pi}$, i.e., x encodes a subformula in Φ and j is a position. That this is correct will again be captured by the formula we are constructing.

Then, we express the following properties which characterize T being a model of φ:

- The function Sk is indeed a Skolem function for φ, i.e., for all variable assignments $\overline{a} \in \mathbb{N}^{\ell}$, if the traces assigned to universally quantified variables π are in the set X_π that π ranges over, then the traces assigned to the existentially quantified variables π' are in the set $X_{\pi'}$ that π' ranges over. Note that X_π and $X_{\pi'}$ can either be T or one of the Y_j, for which we can check membership via the functions T_j. Also note that this formula refers to traces by their names (which are natural numbers) and quantifies over variable assignments (which are encoded by natural numbers), i.e., it is a first-order formula.

- For every variable assignment $\overline{a}$ such that the traces assigned to universally quantified variables π are in the set X_π that π ranges over and that is consistent with the Skolem functions encoded by Sk: the function $x, j \mapsto E(\overline{a}, x, j)$ satisfies the consistency conditions characterising the expansion, and we have $E(\overline{a}, x_0, 0) = 1$, where x_0 is the encoding of ψ. Again, this formula is first-order.
- For every variable assignment $\overline{a}$ as above, the set assigned to Y_j by T_j w.r.t. $\overline{a}$ is indeed the least fixed point of $f_{\overline{a},j}$. Here, we use the information about the stages encoded by T_j and again have to use the expansion E to check that the step formula is satisfied by the selected traces. As before, this can be done in first-order arithmetic.

We leave the tedious, but straightforward, details to the reader. □

7.2. **Satisfiability for full lfp-Hyper^{2}LTL$_{\mathrm{mm}}$.** We just proved that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability for X_a-free sentences is in Σ^1_1, relying on an upper bound on the size of models: This bound is obtained by showing that every model of a satisfiable sentence contains a countable subset that is also a model of the sentence. This is correct, as in X_a-free sentences, only traces from the model are quantified over. On the other hand, the lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence $\forall\pi \in X_a.\ \exists\pi' \in X_d.\ \pi =_{\mathrm{AP}} \pi'$ has only uncountable models, if $|\mathrm{AP}| > 1$.

Thus, in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ formulas with X_a, one can refer to all traces and thus mimic quantification over sets of natural numbers. Furthermore, the satisfiability problem asks for the existence of a model. This *implicit* existential quantifier can be used to mimic existential quantification over sets of sets of natural numbers. Together with the fact that one can implement addition and multiplication in HyperLTL, we show that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability for sentences with X_a is Σ^2_1-hard.

To prove a matching upper bound, we capture the existence of a model and Skolem functions witnessing that it is indeed a model in Σ^2_1. Here, the challenge is to capture the least fixed points when mimicking the second-order quantification of lfp-Hyper^{2}LTL$_{\mathrm{mm}}$. Naively, this requires an existential quantifier ("there exists a set that satisfies the guard") followed by a universal one ("all strict subsets do not satisfy the guard"). However, as traces are encoded as sets, this would require universal quantification of type 3 objects. Thus, this approach is not sufficient to prove a Σ^2_1 upper bound. Instead, we do not explicitly quantify the fixed points, but instead use witness trees for the membership of traces in the fixed points, which are type 1 objects. This is sufficient, as the sets of traces quantified in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ are only used as ranges for trace quantification.

Theorem 7.7. lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *satisfiability is* Σ^2_1*-complete.*

Proof. We begin with the lower bound. Let $S \in \Sigma^2_1$, i.e., there exists a formula of arithmetic of the form

$$\varphi(x) = \exists\mathcal{Y}_1 \subseteq 2^{\mathbb{N}}.\ \cdots \exists\mathcal{Y}_k \subseteq 2^{\mathbb{N}}.\ \psi(x, \mathcal{Y}_1, \ldots, \mathcal{Y}_k)$$

with a single free (first-order) variable x such that $S = \{n \in \mathbb{N} \mid (\mathbb{N}, +, \cdot, <, \in) \models \varphi(n)\}$, where ψ is a formula with arbitrary quantification over type 0 and type 1 objects (but no third-order quantifiers) and free third-order variables $\mathcal{Y}_i$, in addition to the free first-order variable x. We present a polynomial-time translation from natural numbers n to lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences φ_n such that $n \in S$ (i.e., $(\mathbb{N}, +, \cdot, <, \in) \models \varphi(n)$) if and only if φ_n is satisfiable. This implies that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability is Σ^2_1-hard.

Intuitively, we ensure that each model of φ_n contains enough traces to encode each subset of $\mathbb{N}$ by a trace (this requires the use of X_a). Furthermore, we have additional propositions $\mathtt{m}_i$, one for each third-order variable $\mathcal{Y}_i$ existentially quantified in φ, to label the traces encoding sets. Thus, the set of traces marked by $\mathtt{m}_i$ encodes a set of sets of natural numbers, i.e., we have mimicked existential third-order quantification by quantifying over potential models. Hence, it remains to mimic quantification over natural numbers and sets of natural numbers as well as addition and multiplication, which can all be done in HyperLTL: quantification over traces mimics quantification over sets and singleton sets (i.e., numbers) and addition and multiplication can be implemented in HyperLTL (see Proposition 4.1).

To this end, let $\mathrm{AP} = \{\mathtt{x}\} \cup \mathrm{AP}_{\mathtt{m}} \cup \mathrm{AP}_{arith}$ with $\mathrm{AP}_{\mathtt{m}} = \{\mathtt{m}_1, \ldots, \mathtt{m}_k\}$ and $\mathrm{AP}_{arith} = \{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$ and consider the following two formulas (both with a free variable π'):

- $\psi_0 = \left(\bigwedge_{\mathtt{p} \in \mathrm{AP}_{arith}} \mathbf{G}\, \neg \mathtt{p}_{\pi'}\right) \wedge \left(\bigwedge_{i=1}^{k} (\mathbf{G} (\mathtt{m}_i)_{\pi'}) \vee (\mathbf{G}\, \neg (\mathtt{m}_i)_{\pi'})\right)$ expressing that the interpretation of π' may not contain any propositions from AP_{arith} and, for each $i \in [k]$, is either marked by $\mathtt{m}_i$ (if $\mathtt{m}_i$ holds at every position) or is not marked by $\mathtt{m}_i$ (if $\mathtt{m}_i$ holds at no position). Note that there is no restriction on the proposition $\mathtt{x}$, which therefore encodes a set of natural numbers on each trace. Thus, we can use trace quantification to mimic quantification over sets of natural numbers and quantification of natural numbers (via singleton sets). In our encoding, a trace bound to some variable π encodes a singleton set if and only if the formula $(\neg \mathtt{x}_\pi) \mathbf{U} (\mathtt{x}_\pi \wedge \mathbf{X}\, \mathbf{G}\, \neg \mathtt{x}_\pi)$ is satisfied.

  As explained above, we use the markings to encode the membership of such sets in the $\mathcal{Y}_i$, thereby mimicking the existential quantification of the $\mathcal{Y}_i$.

  However, we need to ensure that this marking is done consistently, i.e., there is no trace t over $\{\mathtt{x}\}$ in the model that is both marked by m_i and not marked by m_i. This could happen, as these are just two different traces over AP. However, the formula
  $$\psi_{cons} = \forall \pi \in X_d.\ \forall \pi' \in X_d.\ \pi =_{\{x\}} \pi' \rightarrow \pi =_{\{\mathtt{x}\} \cup \mathrm{AP}_{\mathtt{m}}} \pi'$$
  disallows this.
- $\psi_1 = \bigwedge_{\mathtt{p} \in \{\mathtt{x}\} \cup \mathrm{AP}_{\mathtt{m}}} \mathbf{G}\, \neg \mathtt{p}_{\pi'}$ expresses that the interpretation of π' may only contain propositions in AP_{arith}. We use such traces to implement addition and multiplication in HyperLTL.

Let $\varphi_{(+,\cdot)}$ be the sentence from Proposition 4.1 implementing addition and multiplication: The $\{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$-projection of every model of $\varphi_{(+,\cdot)}$ is $T_{(+,\cdot)}$ as defined for Proposition 4.1. By closely inspecting the formula $\varphi_{(+,\cdot)}$, we can see that it can be brought into the form required for guards in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$. Call the resulting formula $\varphi'_{(+,\cdot)}$. It uses two free variables $\pi_{\mathtt{add}}$ and $\pi_{\mathtt{mult}}$ as "seeds" for the fixed point computation and comes with another LTL formula ψ_s with free variables $\pi_{\mathtt{add}}$ and $\pi_{\mathtt{mult}}$ that ensures that the seeds have the right format.

Now, given $n \in \mathbb{N}$ we define the lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence

$$\begin{aligned}\varphi_n = \psi_{cons} \wedge \forall \pi \in X_a.\ \big((\exists \pi' \in X_d.\ \pi =_{\{\mathtt{x}\}} \pi' \wedge \psi_0) \wedge (\exists \pi' \in X_d.\ \pi =_{\mathrm{AP}_{arith}} \pi' \wedge \psi_1)\big) \wedge \\ \exists \pi_{\mathtt{add}} \in X_d.\ \exists \pi_{\mathtt{mult}} \in X_d.\ \psi_s \wedge \exists (X_{arith}, \curlyvee, \varphi'_{(+,\cdot)}).\ hyp(\exists x.\ (x = n \wedge \psi)).\end{aligned}$$

Intuitively, φ_n requires that the model contains, for each subset S of $\mathbb{N}$, a unique trace encoding S (additionally marked with the $\mathtt{m}_i$ to encode membership of S in the set bound to $\mathcal{Y}_i$), contains each trace over AP_{arith}, the set X_{arith} is interpreted with $T_{(+,\cdot)}$, and the formula $hyp(\exists x.\ (x = n \wedge \psi))$ is satisfied, where the translation hyp is defined below. Note that we use the constant n, which is definable in first-order arithmetic (with a formula that is polynomial in $\log(n)$, using the fact that the constant 2 is definable in first-order arithmetic and then using powers of 2 to define n).

Now, hyp is defined inductively as follows:

- For second-order variables Y, $hyp(\exists Y.\ \psi) = \exists \pi_Y \in X_d.\ \neg \mathtt{add}_{\pi_Y} \wedge \neg \mathtt{mult}_{\pi_Y} \wedge hyp(\psi)$, as only traces not being labeled by add or mult encode sets.
- For second-order variables Y, $hyp(\forall Y.\ \psi) = \forall \pi_Y \in X_d.\ (\neg \mathtt{add}_{\pi_Y} \wedge \neg \mathtt{mult}_{\pi_Y}) \rightarrow hyp(\psi)$.
- For first-order variables y, $hyp(\exists y.\ \psi) = \exists \pi_y \in X_d.\ \neg \mathtt{add}_{\pi_y} \wedge \neg \mathtt{mult}_{\pi_y} \wedge ((\neg \mathtt{x}_{\pi_y}) \mathbf{U} (\mathtt{x}_{\pi_y} \wedge \mathbf{X}\, \mathbf{G}\, \neg \mathtt{x}_{\pi_y})) \wedge hyp(\psi)$.
- For first-order variables y, $hyp(\forall y.\ \psi) = \forall \pi_y \in X_d.\ (\neg \mathtt{add}_{\pi_y} \wedge \neg \mathtt{mult}_{\pi_y} \wedge (\neg \mathtt{x}_{\pi_y}) \mathbf{U} (\mathtt{x}_{\pi_y} \wedge \mathbf{X}\, \mathbf{G}\, \neg \mathtt{x}_{\pi_y})) \rightarrow hyp(\psi)$.
- $hyp(\psi_1 \vee \varphi_2) = hyp(\psi_1) \vee hyp(\psi_2)$.
- $hyp(\neg \psi) = \neg hyp(\psi)$.
- For third-order variables $\mathcal{Y}_i$ and second-order variables Y, $hyp(Y \in \mathcal{Y}_i) = (\mathtt{m}_i)_{\pi_Y}$.
- For second-order variables Y and first-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, $hyp(y_1 + y_2 = y) = \exists \pi \in X_{arith}.\ \mathtt{add}_\pi \wedge \mathbf{F}(\mathtt{x}_{\pi_{y_1}} \wedge \mathtt{arg1}_\pi) \wedge \mathbf{F}(\mathtt{x}_{\pi_{y_2}} \wedge \mathtt{arg2}_\pi) \wedge \mathbf{F}(\mathtt{x}_{\pi_y} \wedge \mathtt{res}_\pi)$.
- For first-order variables y_1, y_2, y, $hyp(y_1 \cdot y_2 = y) = \exists \pi \in X_{arith}.\ \mathtt{mult}_\pi \wedge \mathbf{F}(\mathtt{x}_{\pi_{y_1}} \wedge \mathtt{arg1}_\pi) \wedge \mathbf{F}(\mathtt{x}_{\pi_{y_2}} \wedge \mathtt{arg2}_\pi) \wedge \mathbf{F}(\mathtt{x}_{\pi_y} \wedge \mathtt{res}_\pi)$.

While φ_n 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. An induction shows that we indeed have that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi(n)$ if and only if φ_n is satisfiable.

For the upper bound, we show that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ satisfiability is in Σ^2_1. More formally, we show how to construct a formula of the form

$$\theta(x) = \exists \mathcal{Y}_1 \subseteq 2^{\mathbb{N}}.\ \cdots \exists \mathcal{Y}_k \subseteq 2^{\mathbb{N}}.\ \psi(x, \mathcal{Y}_1, \ldots, \mathcal{Y}_k)$$

with a single free (first-order) variable x such that an lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ is satisfiable if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \theta(\langle \varphi \rangle)$. Here, ψ is a formula of arithmetic with arbitrary quantification over type 0 and type 1 objects (but no third-order quantifiers) and free third-order variables $\mathcal{Y}_i$, in addition to the free first-order variable x, and $\langle \cdot \rangle$ is a polynomial-time computable injective function mapping lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences to natural numbers.

In the following, we assume, without loss of generality, that AP is fixed, so that we can use $|\mathrm{AP}|$ as a constant in our formulas (which is definable in arithmetic). Here, we reuse the encoding of traces as sets of natural numbers as introduced in the proof of Theorem 4.2 using the pairing function $pair$ and a bijection $e \colon \mathrm{AP} \rightarrow \{0, 1, \ldots, |\mathrm{AP}| - 1\}$. Recall that we encode a trace $t \in (2^{\mathrm{AP}})^\omega$ by the set $S_t = \{pair(j, e(\mathtt{p})) \mid j \in \mathbb{N} \text{ and } \mathtt{p} \in t(j)\} \subseteq \mathbb{N}$. A finite collection of sets $S_1, \ldots, S_k$ is uniquely encoded by the set $\{pair(n, j) \mid n \in S_j\}$, i.e., we can encode finite sets of sets by type 1 objects. In particular, we can encode a variable assignment whose domain is finite and contains only trace variables by a set of natural numbers and we can write a formula that checks whether a trace (encoded by a set) is assigned to a certain variable.

Now, the overall proof idea is to let the formula θ of arithmetic express the existence of Skolem functions for the existentially quantified variables in the lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ such that each variable assignment that is consistent with the Skolem functions satisfies the maximal quantifier-free subformula of φ. For trace variables π, a Skolem function is a type 2 object, i.e., a function mapping a tuple of sets of natural numbers (encoding a tuple of traces, one for each variable quantified universally before π) to a set of natural numbers (encoding a trace for π). However, to express that the interpretation of a second-order variable X is indeed a least fixed point we need both existential quantification ("there exists a set" that satisfies the guard) and universal quantification ("every other set that satisfies the guard is larger"). Thus, handling second-order quantification this way does not yield the desired Σ_1^2 upper bound.

Instead we use that fact that membership of a trace in an HyperLTL-definable least fixed point can be witnessed by a finite tree labeled by traces (Lemma 7.5), i.e., by a type 1 object . Thus, instead of capturing the full least fixed point in arithmetic, we verify on-the-fly for each trace quantification of the form $\exists\pi \in Y_j$ or $\forall\pi \in Y_j$ whether the interpretation of π is in the interpretation of Y_j, which only requires the quantification of a witness tree.

To this end, fix some

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

where ψ is quantifier-free. As before, we assume without loss of generality that each trace variable is quantified at most once in φ, which can always be achieved by renaming variables. This implies that for each trace variable π quantified in some γ_j, there exists a unique second-order variable X_π such that π ranges over X_π. Furthermore, we assume that each Y_j is different from X_d and X_a, which can again be achieved by renaming variables, if necessary. Recall that the values of the least fixed points are uniquely determined by the interpretations of X_d, X_a, and the trace variables in the γ_j.

We need to adapt the definition of witness trees (and related concepts) to account for the fact that we here allow the usage of the variable X_a. Hence, we say that a variable assignment Π is φ-sufficient if Π's domain contains exactly the variables X_d, X_a, and the trace variables in the γ_j.

The expansion of a formula ξ with respect to Π is still denoted by $e_{\Pi,\xi}\colon \Xi \times \mathbb{N} \to \{0,1\}$ and defined as in the proof of Theorem 7.6. Here, Ξ denotes the set of subformulas of ξ. Let us also remark for further use that a function from $\Xi \times \mathbb{N}$ to $\{0,1\}$ is a type 1 object (as functions from $\mathbb{N}$ to $\mathbb{N}$ can be encoded as subsets of $\mathbb{N}$ via their graph and the pairing function introduced below) and that all five requirements on the expansion can then be expressed in first-order logic.

Now, we adapt the definition of witness trees as follows. Let

$$\varphi_j^{\mathrm{con}} = \dot\pi_1 \triangleright Y_j \wedge \cdots \wedge \dot\pi_n \triangleright Y_j \wedge \forall \ddot\pi_1 \in Z_1.\ \ldots \forall \ddot\pi_{n'} \in Z_{n'}.\ \psi_j^{\mathrm{step}} \to \ddot\pi_m \triangleright Y_j. \tag{7.2}$$

inducing the unique least fixed point that Y_j ranges over. In the original definition in Section 7.1, each Z_i is either X_d or one of the $Y_{j'}$ with $j' \leq j$. Here, it may also be X_a. However, selecting a trace from the interpretation of X_a does not introduce any constraints (as it is interpreted by the set of all traces, as we do not allow to re-quantify X_a). Hence, such a node will be just a leaf without any constraint.

Formally, let us fix a $j^* \in \{1, \ldots, k\}$, a φ-sufficient assignment Π, and a trace t^*. A Π-witness tree for (t^*, j^*) (which witnesses $t^* \in \mathrm{lfp}(\Pi, j^*)$) is an ordered finite tree whose vertices are labeled by pairs (t, j) where t is a trace and where j is in $\{1, \ldots, k\} \cup \{a, d\}$ (where a and d are fresh symbols) such that the following are satisfied:

- The root is labeled by (t^*, j^*).

- If a vertex is labeled with (t, a) for some trace t, then it must be a leaf. Note that t is in $\Pi(X_a)$, as that set contains all traces.
- If a vertex is labeled with (t, d) for some trace t, then it must be a leaf and $t \in \Pi(X_d)$.
- Let (t, j) be the label of some vertex v with $j \in \{1, \ldots, k\}$ and let φ_j^{con} as in (7.2). If v is a leaf, then we must have $t = \Pi(\dot\pi_i)$ for some $i \in \{1, \ldots, n\}$. If v is an internal vertex, then it must have n' successors labeled by $(t_1, j_1), \ldots, (t_{n'}, j_{n'})$ (in that order) such that $\Pi[\ddot\pi_1 \mapsto t_1, \ldots, \ddot\pi_{n'} \mapsto t_{n'}] \models \psi_j^{\mathrm{step}}$, $t = t_m$, and such that the following holds for all $i \in \{1, \ldots, n'\}$:
  - If $Z_i = X_a$, then the i-th successor of v is a leaf and $j_i = a$.
  - If $Z_i = X_d$, then the i-th successor of v is a leaf, $t_i \in \Pi(X_d)$, and $j_i = d$.
  - If $Z_i = Y_{j'}$ for some j' (which satisfies $1 \leq j' \leq j$ by definition of lfp-Hyper^{2}LTL$_{\mathrm{mm}}$), then we must have that $j_i = j'$ and that the subtree rooted at the i-th successor of v is a Π-witness tree for (t_i, j_i).

The following proposition states that membership in the fixed points is witnessed by witness trees. It is obtained by generalizing the similar argument for X_a-free sentences presented in Lemma 7.5.

Lemma 7.8. *Let Π be a (φ, j)-sufficient assignment with $\Pi(Y_{j'}) = e(\Pi)(Y_{j'})$ for all $j' < j$. Then, we have $t \in \mathrm{lfp}(\Pi, j)$ if and only if there is a Π-witness tree for (t, j).*

Note that a witness tree is a type 1 object: The (finite) tree structure can be encoded by

- a natural number $s > 0$ (encoding the number of vertices),
- a function from $\{1, \ldots, s\} \times \{1, \ldots, n''\} \to \{0, 1, \ldots, s\}$ encoding the child relation, i.e., $(v, j) \mapsto v'$ if and only if the j-th child of v is v' (where we use 0 for undefined children and n'' is the maximum over all n' in the φ_j^{con}),
- s traces over AP and s values in $\{1, \ldots, k, k+1, k+2\}$ to encode the labeling (where we use $k+1$ for a and $k+2$ for d).

Note that the function encoding the child relation can be encoded by a finite set by encoding its graph using the pairing function while all other objects can directly be encoded by sets of natural numbers, and thus can be encoded by a single set as explained above.

Furthermore, one can write a second-order formula $\psi_{hasTree}(X_D, A, X_{t^*}, j^*)$ with free third-order variable X_D (encoding a set of traces T), free second-order variables A (encoding a variable assignment Π whose domain contains exactly the trace variables in the γ_j) and X_{t^*} (encoding a trace t^*), and free first-order variable j^* that holds in $(\mathbb{N}, +, \cdot, <, \in)$ if and only if there exists a $\Pi[X_d \mapsto T, X_a \mapsto (2^{\mathrm{AP}})^\omega]$-witness tree for (t^*, j^*). To evaluate the formulas $\psi_{j'}^{\mathrm{step}}$ as required by the definition of witness trees, we rely on the expansion as introduced above, which here is a mapping from vertices in the tree and subformulas of the $\psi_{j'}^{\mathrm{step}}$ to $\{0, 1\}$, and depends on the set of traces encoded by X_D and the variable assignment encoded by A. Such a function is a type 1 object and can therefore be quantified in $\psi_{hasTree}(X_D, A, X_{t^*}, j^*)$.

Recall that we construct a formula $\theta(x)$ with a free first-order variable x such that an lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ is satisfiable if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \theta(\langle\varphi\rangle)$, where $\langle\cdot\rangle$ is a polynomial-time computable injective function mapping lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences to natural numbers.

With this preparation, we can define $\theta(x)$ such that it is not satisfied in $(\mathbb{N}, +, \cdot, <, \in)$ if the interpretation of x does not encode an lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence. If it does encode such a sentence φ, let ψ be its maximal quantifier-free subformula (which is "computable" in first-order arithmetic using a suitable encoding $\langle\cdot\rangle$). Then, θ expresses the existence of

- a set T of traces (bound to the third-order variable X_D and encoded as a set of sets of natural numbers, i.e., a type 2 object),
- Skolem functions for the existentially quantified trace variables in the γ_j (which can be encoded by functions from $(2^{\mathbb{N}})^\ell$ to $2^{\mathbb{N}}$ for some ℓ, i.e., by a type 2 object), and
- a function $e\colon 2^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N} \to \{0,1\}$

such that the following is true for all variable assignments Π (restricted to the trace variables in the γ_j and bound to the second-order variable A): If

- Π is consistent with the Skolem functions for all existentially quantified variables,
- for all universally quantified π in some γ_j with $X_\pi = X_d$, we have $\Pi(\pi) \in T$, and
- for all universally quantified π in some γ_j with $X_\pi = Y_{j^*}$, $\psi_{hasTree}(X_D, A, \Pi(\pi), j^*)$ holds,

then

- for all existentially quantified π in some γ_j with $X_\pi = X_d$, we have $\Pi(\pi) \in T$, and
- for all existentially quantified π in some γ_j with $X_\pi = Y_{j^*}$, $\psi_{hasTree}(X_D, A, \Pi(\pi), j^*)$ holds, and
- the function $(x,y) \mapsto e(A,x,y)$ is the expansion of ψ with respect to Π and we have $e(A,\psi,0) = 1$ (here we identify subformulas of ψ by natural numbers).

We leave the tedious, but routine, details to the reader. □

Note that in the lower bound proof, a single second-order quantifier (for the set of traces implementing addition and multiplication) suffices.

7.3. **Finite-State Satisfiability and Model-Checking for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$.** In this subsection, we settle the complexity of finite-state satisfiability and model-checking for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$, both for general sentences and X_a-free sentences.

Theorem 7.9. lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *finite-state satisfiability and model-checking are polynomial-time equivalent to truth in second-order arithmetic. The lower bounds already hold for X_a-free formulas.*

The result follows from the following building blocks, which are visualized in Figure 3:

- All lower bounds proven for X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences trivially also hold for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ while upper bounds for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ also hold for the fragment of X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences.
- We show that lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ finite-state satisfiability can in polynomial time be reduced to lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ model-checking for X_a-free sentences (see Lemma 7.10 below).
- lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ finite-state satisfiability for X_a-free sentences is at least as hard as truth in second-order arithmetic (see Lemma 7.11 below).
- lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ model-checking can in polynomial time be reduced to truth in second-order arithmetic (see Lemma 7.12 below).

We begin with reducing finite-state satisfiability to model-checking.

Lemma 7.10. lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *finite-state satisfiability is polynomial-time reducible to* lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *model-checking for X_a-free sentences.*

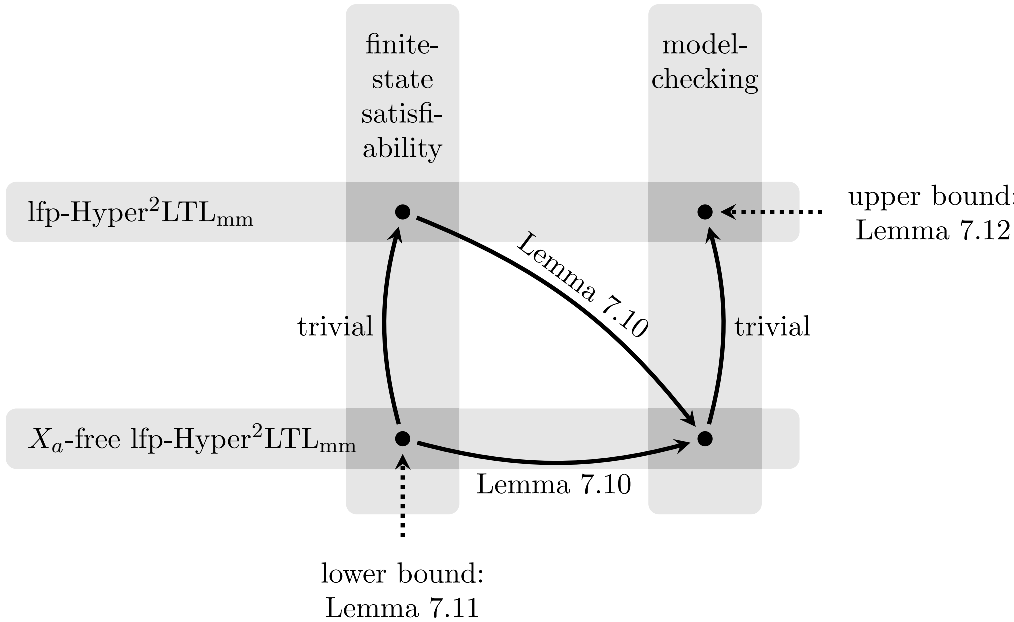


Figure 3: The reductions (drawn as solid arrows) and lemmata proving Theorem 7.9.

Proof. Intuitively, we reduce finite-state satisfiability of φ to model-checking by writing a sentence φ' existentially quantifying a finite transition system $\mathfrak{T}$ (encoded by two traces) and expressing that $\mathfrak{T}$ satisfies φ and then model-checking φ' in a fixed transition system $\mathfrak{T}'$.

To simplify our construction, we begin by showing that we can restrict ourselves without loss of generality to transition systems with a unique initial vertex, i.e., we show that a Hyper^{2}LTL sentence is finite-state satisfiable if and only if it is satisfied by a transition system with a single initial vertex. Then, we show how to encode finite transition systems by pairs of traces, one trace encoding the labels of the vertices and the other one the adjacency matrix of transition relation. Using this encoding, we then show how to capture the set of (encodings of) prefixes of traces using a least fixed point. Then, we can translate finite-state satisfiability of a sentence φ to model-checking by existentially quantifying traces encoding a finite transition system $\mathfrak{T}$ and then relativizing the quantifiers of φ to traces whose prefixes are all in the least fixed point, which implies that they are traces of $\mathfrak{T}$. To this end, it suffices to model-check the resulting formula in a fixed transition system $\mathfrak{T}'$ that contains all traces over the atomic propositions occurring in φ as well as some additional propositions used in the construction.

We begin by showing that we can restrict ourselves, without loss of generality, to finite-state satisfiability by transition systems with a single initial vertex, which simplifies our construction. Let $\mathfrak{T} = (V, E, I, \lambda)$ be a transition system and φ a Hyper^{2}LTL sentence. Consider the transition system $\mathfrak{T}_{\mathbf{X}} = (V \cup \{v_I\}, E', \{v_I\}, \lambda')$ with a fresh initial vertex v_I,

$$E' = E \cup \{(v_I, v_I)\} \cup \{(v_I, v) \mid v \in I\},$$

and $\lambda'(v_I) = \{\$\}$ (for a fresh proposition $\$$) and $\lambda'(v) = \lambda(v)$ for all $v \in V$. Here, we add the self-loop on the fresh initial vertex v_I to deal with the special case of I being empty, which would make v_I terminal without the self-loop.

Now, we have
$$\mathrm{Tr}(\mathfrak{T}_{\mathbf{X}}) = \{\$\}^\omega \cup \{\{\$\}^+ \cdot t \mid t \in \mathrm{Tr}(\mathfrak{T})\}.$$
Furthermore, let $\varphi_{\mathbf{X}}$ be the formula obtained from φ by adding an $\mathbf{X}$ to the maximal quantifier-free subformula of φ and by inductively replacing

- each $\exists \pi \in X.\ \psi$ by $\exists \pi \in X.\ \mathbf{X}(\neg \$_\pi \wedge \psi)$ and
- each $\forall \pi \in X.\ \psi$ by $\forall \pi \in X.\ \mathbf{X}(\neg \$_\pi \rightarrow \psi)$.

Then, $\mathfrak{T} \models \varphi$ if and only if $\mathfrak{T}_{\mathbf{X}} \models \varphi_{\mathbf{X}}$. Thus, φ is finite-state satisfiable if and only if there exists a finite transition system with a single initial vertex that satisfies $\varphi_{\mathbf{X}}$.

Now, we show how to encode a finite transition system using two traces. To this end, let φ be a lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence (over AP, which we assume to be fixed). We define $\mathrm{AP}' = \mathrm{AP} \cup \{\mathtt{x}, \#\} \cup \{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$. Throughout the construction, we use a second-order variable Y_{arith} which will be interpreted by $T_{(+,\cdot)}$ (see Proposition 4.1). We now explain how we encode a finite transition system by two traces over AP':

- Consider the formula $\varphi_V = \neg\#_{\pi_V} \wedge (\neg\#_{\pi_V})\,\mathbf{U}(\#_{\pi_V} \wedge \mathbf{X}\,\mathbf{G}\,\neg\#_{\pi_V})$ expressing that the interpretation of π_V contains a unique position where $\#$ holds. This position must not be the first one. Hence, the AP-projection of such a trace up to the $\#$ is a nonempty word $w(0)w(1)\cdots w(n{-}1)$ for some $n > 0$. It induces $V = \{0, 1, \ldots, n{-}1\}$ and $\lambda\colon V \rightarrow 2^{\mathrm{AP}}$ with $\lambda(v) = w(v) \cap \mathrm{AP}$. Furthermore, we fix $I = \{0\}$.
- Let t be a trace over AP'. It induces the edge relation $E = \{(i,j) \in V \times V \mid \mathtt{x} \in t(i{\cdot}n{+}j)\}$, i.e., we interpret the first n^2 truth values of $\mathtt{x}$ in t as an adjacency matrix (encoded as a sequence of rows of the matrix). Furthermore, if t is the interpretation of π_E satisfying φ_E defined below, then every vertex in $\{0, 1, \ldots, n-1\}$ has a successor in E.

$$\begin{aligned}\varphi_E = \forall \pi \in Y_{arith}.\ & [\mathtt{mult}_\pi \wedge (\mathbf{F}(\mathtt{arg1}_\pi \wedge \mathbf{X}\,\mathbf{F}\,\#_{\pi_V})) \wedge \mathbf{F}(\mathtt{arg2}_\pi \wedge \#_{\pi_V})] \rightarrow \\ & \exists \pi' \in Y_{arith}.\ \mathtt{add}_{\pi'} \wedge \mathbf{F}(\mathtt{arg1}_{\pi'} \wedge \mathtt{res}_\pi) \wedge (\mathbf{F}(\mathtt{arg2}_{\pi'} \wedge \mathbf{X}\,\mathbf{F}\,\#_{\pi_V}) \wedge \mathbf{F}(\mathtt{res}_{\pi'} \wedge \mathtt{x}_{\pi_E}).\end{aligned}$$

Intuitively, the formula φ_E expresses that for every $i \in V = \{0, 1, \ldots, n-1\}$ there is a $j \in V$ such that the edge (i,j) is in the edge relation encoded by the trace assigned to π_E, which is the case if and only if the proposition $\mathtt{x}$ holds at position $i \cdot n + j$. To this end, we universally quantify a trace (assigned to π) encoding a multiplication $n_1 \cdot n_2 = n_3$. Now, if

- n_1 is strictly smaller than n, expressed using the subformula $\mathbf{F}(\mathtt{arg1}_\pi \wedge \mathbf{X}\,\mathbf{F}\,\#_{\pi_V})$ (recall that $\#$ holds at position n (and nowhere else) of the trace assigned to π_V), and if
- n_2 is equal to n, expressed using the subformula $\mathbf{F}(\mathtt{arg2}_\pi \wedge \#_{\pi_V})$,

i.e., n_1 corresponds to i and n_2 corresponds to n in $i \cdot n + j$, then we existentially quantify a trace (assigned to π') encoding an addition $n_4 + n_5 = n_6$. Here, we require that

- n_4 is equal to n_3 (and thus equal to $i \cdot n$), using the subformula $\mathbf{F}(\mathtt{arg1}_{\pi'} \wedge \mathtt{res}_\pi)$,
- n_5 is strictly smaller than n, using the subformula $\mathbf{F}(\mathtt{arg2}_{\pi'} \wedge \mathbf{X}\,\mathbf{F}\,\#_{\pi_V})$, and
- the proposition $\mathtt{x}$ holds at position n_6 of the trace t assigned to π_E, using the subformula $\mathbf{F}(\mathtt{res}_{\pi'} \wedge \mathtt{x}_{\pi_E})$,

i.e., n_5 corresponds to j in $i \cdot n + j$ and n_6 is therefore equal to $i \cdot n + j$.

Thus, π_V and π_E satisfying the above formulas indeed encode a finite transition system $\mathfrak{T}$ with a single initial vertex as described above.

Our next step is to encode trace prefixes of the encoded transition system and then inductively construct the set of all such prefixes, by describing it as the minimal fixed point that contains the trace prefix of the unique path of length one starting at the initial vertex and that is closed under extending any trace prefix by one transition. Note that this only requires us to keep track of the last vertex of the path inducing the trace. We do so by marking a position of the trace that encodes the vertex as done before.

Consider the formula

$$\varphi_{prefix} = [(\neg\#_\pi)\,\mathbf{U}(\#_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg\#_\pi)] \wedge [(\neg\mathtt{x}_\pi)\,\mathbf{U}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg\mathtt{x}_\pi)] \wedge \mathbf{F}(\mathtt{x}_\pi \wedge \mathbf{X}\,\mathbf{F}\,\#_{\pi_V}),$$

which is satisfied if the interpretation t of π has a unique position ℓ at which $\#$ holds and if it has a unique position v where $\mathtt{x}$ holds. Furthermore, v must be strictly smaller than n, i.e., it is in V. In this situation, we interpret the AP-projection of $t(0)\cdots t(\ell-1)$ as a trace prefix over AP and v as a vertex (intuitively, this is interpreted as the vertex where the path prefix inducing the trace prefix ends).

Using this encoding, the formula

$$\varphi_{init} = \varphi_{prefix}[\pi/\pi_{init}] \wedge \mathtt{x}_{\pi_{init}} \wedge \mathbf{X}\,\#_{\pi_{init}} \wedge \bigwedge_{\mathtt{p}\in\mathrm{AP}} \mathtt{p}_{\pi_V} \leftrightarrow \mathtt{p}_{\pi_{init}}$$

ensures that π_{init} is interpreted with a trace that encodes the trace prefix of the unique path of length one starting at the initial vertex. Here, $\varphi_{prefix}[\pi/\pi_{init}]$ denotes the formula obtained from φ_{prefix} by replacing each occurrence of π by π_{init}.

Recall that we have fixed a lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ over AP and that we defined $\mathrm{AP}' = \mathrm{AP} \cup \{\mathtt{x}, \#\} \cup \{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$. Let $\mathfrak{T}'$ be a fixed transition system such that $\mathrm{Tr}(\mathfrak{T}') = (2^{\mathrm{AP}'})^\omega$. We can construct $\mathfrak{T}'$ such that it has $2^{|\mathrm{AP}'|}$ many vertices (which is constant as AP is fixed!) Given φ, we construct an X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ' (over AP$'$) such that φ is satisfied by a finite transition system with a single initial vertex if and only if $\mathfrak{T}' \models \varphi'$.

To this end, first consider the formula

$$\varphi'' = \exists \pi_V \in X_d.\ \exists \pi_E \in X_d.\ \exists \pi_{init} \in X_d.\ \varphi_V \wedge \varphi_E \wedge \varphi_{init} \wedge \exists(Y_{prefixes}, \curlyvee, \varphi_{\mathrm{con}}).\ rel(\varphi),$$

where φ_{con} and $rel(\varphi)$ are introduced below. Here, we quantify π_V and π_E and ensure that they encode a finite transition system $\mathfrak{T}$ as described above, and want to express that the interpretation of $Y_{prefixes}$ must contain exactly the encodings of prefixes of traces of $\mathfrak{T}$.

To this end, φ_{con} is defined as

$$\varphi_{\mathrm{con}} = \pi_{init} \triangleright Y_{prefixes} \wedge \forall \pi_0 \in Y_{prefixes}.\ \forall \pi_1 \in X_d.\ \psi_{\mathrm{step}} \rightarrow \pi_1 \triangleright Y_{prefixes}$$

with the formula ψ_{step}

$$\begin{aligned}\varphi_{prefix}[\pi/\pi_1] \wedge (\bigwedge_{\mathtt{p}\in\mathrm{AP}} \mathtt{p}_{\pi_0} \leftrightarrow \mathtt{p}_{\pi_1})\,\mathbf{U}(\#_{\pi_0} \wedge \mathbf{X}\,\#_{\pi_1}) \wedge \\ \Big(\bigwedge_{\mathtt{p}\in\mathrm{AP}} [\mathbf{F}(\mathtt{p}_{\pi_V} \wedge \mathtt{x}_{\pi_1})] \leftrightarrow [\mathbf{F}(\mathtt{p}_{\pi_1} \wedge \mathbf{X}\,\#_{\pi_1})]\Big) \wedge \varphi_{edge},\end{aligned}$$

where φ_{edge} checks whether there is an edge in the encoded transition system $\mathfrak{T}$ between the vertex induced by π_0 and the vertex induced by π_1 (analogously to φ_E):

$$\begin{aligned}\varphi_{edge} = \exists \pi_m \in Y_{arith}.\ \exists \pi_a \in Y_{arith}.\ \mathtt{mult}_{\pi_m} \wedge \mathbf{F}(\mathtt{arg1}_{\pi_m} \wedge \mathtt{x}_{\pi_0}) \wedge \mathbf{F}(\mathtt{arg2}_{\pi_m} \wedge \#_{\pi_V}) \wedge \\ \mathtt{add}_{\pi_a} \wedge \mathbf{F}(\mathtt{arg1}_{\pi_a} \wedge \mathtt{res}_{\pi_m}) \wedge \mathbf{F}(\mathtt{arg2}_{\pi_a} \wedge \mathtt{x}_{\pi_1}) \wedge \mathbf{F}(\mathtt{res}_{\pi_a} \wedge \mathtt{x}_{\pi_E})\end{aligned}$$

Intuitively, the until subformula of ψ_{step} ensures that the finite trace encoded by the interpretation of π_0 is a prefix of the finite trace encoded by the interpretation of π_1, which has one additional letter. The equivalence $\mathbf{F}(\mathtt{p}_{\pi_V} \wedge \mathtt{x}_{\pi_1}) \leftrightarrow \mathbf{F}(\mathtt{p}_{\pi_1} \wedge \mathbf{X}\,\#_{\pi_1})$ then ensures that the additional letter is the label of the vertex v induced by the interpretation of π_1:

the left-hand side of the equivalence "looks up" the truth values of the label of v in π_V and the right-hand side the truth values at the additional letter in π_1, which comes right before the $\#$.

Thus, the least fixed point induced by ψ_{con} contains exactly the encodings of the prefixes of traces of $\mathfrak{T}$ as the prefix of length one is included and ψ_{step} extends the prefixes by one more letter.

Thus, it that remains is to relativize the quantifiers of φ (the sentence we started with) so that they range over the traces of the transition system $\mathfrak{T}$ we have quantified using π_V and π_E. Recall that the formula φ' we are constructed is model-checked over a transition system $\mathfrak{T}'$ that contains all traces over AP'. Thus, quantification over X_a in φ (which ranges over arbitrary traces over AP) is in φ' replaced by quantification over X_d (which ranges over arbitrary traces in $\mathrm{AP}' \supseteq \mathrm{AP}$ and the additional propositions are irrelevant here). Further, quantification over X_d in φ (which ranges over traces of a finite transition system) is in φ' replaced by quantification over X_d, but restricted to traces whose prefixes are all in Y_{prefixes}. Such a trace must be a trace of the transition system, as it is finite. Note that trace quantification over second-order variables $Y \notin \{X_d, X_a\}$ does not have to not relativized, as the traces added to the fixed points interpreting Y come from X_d, X_a, or previously quantified second-order variables.

So, the formula $\mathit{rel}(\varphi)$ is defined by inductively replacing

- each $\exists \pi \in X_a.\ \psi$ by $\exists \pi \in X_d.\ \psi$,
- each $\forall \pi \in X_a.\ \psi$ by $\forall \pi \in X_d.\ \psi$,
- each $\exists \pi \in X_d.\ \psi$ by $\exists \pi \in X_d.\ \psi_r \wedge \psi$, and
- each $\forall \pi \in X_d.\ \psi$ by $\forall \pi \in X_d.\ \psi_r \rightarrow \psi$,

where ψ_r expresses that the trace assigned to π must be one of the transition system $\mathfrak{T}$:

$$\forall \pi' \in X_d.\ \neg\#_{\pi'} \wedge ((\neg\#_{\pi'})\,\mathbf{U}(\#_{\pi'} \wedge \mathbf{X}\,\mathbf{G}\,\neg\#_{\pi'})) \rightarrow$$
$$\exists \pi_p \in Y_{\mathit{prefixes}}.\ \mathbf{F}(\#_{\pi'} \wedge \#_{\pi_p}) \wedge (\bigwedge_{\mathtt{p} \in \mathrm{AP}} \mathtt{p}_{\pi} \leftrightarrow \mathtt{p}_{\pi_p})\,\mathbf{U}\,\#_{\pi_p}$$

Here, we use the fact that if all prefixes of a trace are in the prefixes of the traces of $\mathfrak{T}$, then the trace itself is also one of $\mathfrak{T}$. Thus, we can require for every $n > 0$, there is a trace in Y_{prefixes} encoding a prefix of length n of a trace of $\mathfrak{T}$ such that π has that prefix.

To finish the proof, we need to ensure that Y_{arith} does indeed contain the traces implementing addition and multiplication (see Proposition 4.1 and the proof of Theorem 7.7 on Page 39 for details): $\mathfrak{T}'$ is a model of

$$\varphi' = \exists \pi_{\mathtt{add}} \in X_d.\ \exists \pi_{\mathtt{mult}} \in X_d.\ \psi_s \wedge \exists (X_{\mathit{arith}}, \curlyvee, \varphi'_{(+,\cdot)}).\ \varphi''$$

if and only if φ is satisfied by a finite transition system with a single initial vertex. □

Now, we prove the lower bound for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ finite-state satisfiability.

Lemma 7.11. *Truth in second-order arithmetic is polynomial-time reducible to finite-state satisfiability for X_a-free* lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *sentences.*

Proof. Given a sentence φ of second-order arithmetic, we show how to construct an X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ' such that $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$ if and only if φ' is finite-state satisfiable. Intuitively, we write a formula that is finite-state satisfiable, but only by finite transition systems that contain all traces over $\{\mathtt{x}\}$. As before, this allows us to mimic first- and second-order quantification by quantification of traces. Together with the fact that addition and multiplication can be implemented in HyperLTL, this suffices to obtain the desired construction. Note that this only works because we are asking for a finite transition system to satisfy φ': general satisfiability for X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences is simpler (see Theorem 7.6).

Let $\mathrm{AP} = \{\mathtt{x}, \#\}$. We begin by presenting a (satisfiable) lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ_{Prefs} that ensures that the set of prefixes of the $\{\mathtt{x}\}$-projection of each of its models is equal to $(2^{\{\mathtt{x}\}})^*$. To this end, consider the conjunction φ_{Prefs} of the following formulas:

- $\forall \pi \in X_d.\ \mathbf{G}(\#_\pi \rightarrow \mathbf{X}\,\mathbf{G}\,\neg\#_\pi))$, which expresses that each trace in the model has at most one position where the proposition # holds.
- $\exists \pi \in X_d.\ \#_\pi$, which expresses that each model contains a trace where # holds at the first position.
- $\forall \pi \in X_d.\ \exists \pi' \in X_d.\ (\mathbf{F}\,\#_\pi) \rightarrow (\mathtt{x}_\pi \leftrightarrow \mathtt{x}_{\pi'})\,\mathbf{U}(\#_\pi \wedge \neg\mathtt{x}_{\pi'} \wedge \mathbf{X}\,\#_{\pi'})$ expressing that for every trace t in the model with a #, there is another trace t' in the model such that the $\{\mathtt{x}\}$-projection w of t up to the # and the $\{\mathtt{x}\}$-projection w' of t' up to the # satisfy $w' = w\emptyset$.
- $\forall \pi \in X_d.\ \exists \pi' \in X_d.\ (\mathbf{F}\,\#_\pi) \rightarrow (\mathtt{x}_\pi \leftrightarrow \mathtt{x}_{\pi'})\,\mathbf{U}(\#_\pi \wedge \mathtt{x}_{\pi'} \wedge \mathbf{X}\,\#_{\pi'})$ expressing that for every trace t in the model with a # there is another trace t' in the model such that the $\{\mathtt{x}\}$-projection w of t up to the # and the $\{\mathtt{x}\}$-projection w' of t' up to the # satisfy $w' = w\{\mathtt{x}\}$.

A straightforward induction shows that the set of prefixes of the $\{\mathtt{x}\}$-projection of each model of φ_{Prefs} is equal to $(2^{\{\mathtt{x}\}})^*$. Furthermore, let $\mathfrak{T}$ be a finite transition system that is a model of φ_{Prefs}, i.e., with $\mathrm{Tr}(\mathfrak{T}) \models \varphi_{\mathit{Prefs}}$. Then, the $\{\mathtt{x}\}$-projection of $\mathrm{Tr}(\mathfrak{T})$ must be equal to $(2^{\{\mathtt{x}\}})^\omega$, which follows from the fact that the set of traces of a finite transition system is closed (see, e.g., [BK08] for the necessary definitions). Thus, as usual, we can mimic set quantification over $\mathbb{N}$ by trace quantification.

Further, recall that we can implement addition and multiplication in lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ (see Proposition 4.1 and the proof of Theorem 7.7 on Page 39 for how to turn the formula $\varphi_{(+,\cdot)}$ into a least fixed point guard): there is an lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ guard for a second-order quantifier such that the $\{\mathtt{arg1}, \mathtt{arg2}, \mathtt{res}, \mathtt{add}, \mathtt{mult}\}$-projection of the unique least fixed point that satisfies the guard is equal to $T_{(+,\cdot)}$.

Thus, we can mimic set quantification over $\mathbb{N}$ and implement addition and multiplication, which allow us to reduce truth in second-order arithmetic to finite-state satisfiability for X_a-free sentences using the function *hyp* presented in the proof of Theorem 4.2:

$$\varphi' = \varphi_{\mathit{Prefs}} \wedge \exists \pi_{\mathtt{add}} \in X_d.\ \exists \pi_{\mathtt{mult}} \in X_d.\ \psi_s \wedge \exists (X_{\mathit{arith}}, \curlyvee, \varphi'_{(+,\cdot)}).\ \mathit{hyp}(\varphi)$$

is finite-state satisfiable if and only if $(\mathbb{N}, +, \cdot, <, \in) \models \varphi$. □

Note that the formula φ' we have constructed in the proof just uses one second-order quantifier.

This result, i.e., that finite-state satisfiability for X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences is at least as hard as truth in second-order arithmetic, should be contrasted with the general satisfiability problem for X_a-free lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentences, which is "only" Σ^1_1-complete [FZ25], i.e., much simpler. The reason is that every satisfiable lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence has a countable model (i.e., a countable set of traces). This is even true for the formula φ_{Prefs} we have constructed. However, every *finite-state* transition system that satisfies the formula must have uncountably many traces. This fact allows us to mimic quantification over arbitrary subsets of $\mathbb{N}$, which is not possible in a countable model. Thus, the general satisfiability problem is simpler than the finite-state satisfiability problem.

Finally, we prove the upper bound for lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ model-checking.

Lemma 7.12. lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ *model-checking is polynomial-time reducible to truth in second-order arithmetic.*

Proof. As in the (upper bound) proof of Theorem 4.2 (and several times thereafter), we mimic trace quantification via quantification of sets of natural numbers and capture the temporal operators using addition. To capture the "quantification" of fixed points, we use again witness trees (see Subsection 7.1 for details). Recall that these depend on a variable assignment of the trace variables. Thus, in our construction, we need to explicitly handle such an assignment as well (encoded by a set of natural numbers) in order to be able to correctly apply witness trees.

Let $\mathfrak{T} = (V, E, I, \lambda)$ be a finite transition system. We assume without loss of generality that $V = \{0, 1, \ldots, n\}$ for some $n \geq 0$. Recall that $\mathit{pair}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ denotes Cantor's pairing function defined as $\mathit{pair}(i, j) = \frac{1}{2}(i+j)(i+j+1)+j$, which is a bijection and implementable in arithmetic. We encode a path $\rho(0)\rho(1)\rho(2)\cdots$ through $\mathfrak{T}$ by the set $\{\mathit{pair}(j, \rho(j)) \mid j \in \mathbb{N}\} \subseteq \mathbb{N}$. Not every set encodes a path, but the first-order formula

$$\begin{aligned}\varphi_{\mathit{isPath}}(Y) = {}& \forall x.\ \forall y.\ y > n \to \mathit{pair}(x, y) \notin Y \wedge\\ & \forall x.\ \forall y_0.\ \forall y_1.\ (\mathit{pair}(x, y_0) \in Y \wedge \mathit{pair}(x, y_1) \in Y) \to y_0 = y_1 \wedge\\ & \bigvee_{v \in I} \mathit{pair}(0, v) \in Y \wedge\\ & \forall j.\ \bigvee_{(v,v') \in E} \mathit{pair}(j, v) \in Y \wedge \mathit{pair}(j+1, v') \in Y\end{aligned}$$

checks if a set does encode a path of $\mathfrak{T}$.

Furthermore, fix some bijection $e\colon \mathrm{AP} \to \{0, 1, \ldots, |\mathrm{AP}| - 1\}$. As before, we encode a trace $t \in (2^{\mathrm{AP}})^\omega$ by the set $S_t = \{\mathit{pair}(j, e(\mathtt{p})) \mid j \in \mathbb{N} \text{ and } \mathtt{p} \in t(j)\} \subseteq \mathbb{N}$. While not every subset of $\mathbb{N}$ encodes some trace t, the first-order formula

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

checks if a set does encode a trace.

Finally, the first-order formula

$$\varphi_{\mathfrak{T}}(Y) = \exists Y_p.\ \mathit{isPath}(Y_p) \wedge \forall j.\ \bigwedge_{\mathtt{p} \in \mathrm{AP}} \left(\mathit{pair}(j, e(\mathtt{p})) \in Y \leftrightarrow \bigvee_{v\colon \mathtt{p} \in \lambda(v)} \mathit{pair}(j, v) \in Y_p \right)$$

checks whether the set Y encodes the trace of some path through $\mathfrak{T}$.

As in the proof of Theorem 7.7, we need to encode variable assignments (whose domain is restricted to trace variables) via sets of natural numbers. Using this encoding, one can "update" encoded assignments, i.e., there exists a formula $\varphi^{\pi}_{\mathit{update}}(A, A', Z)$ that is satisfied if and only if

- the set A encodes a variable assignment Π,
- the set A' encodes a variable assignment Π',
- the set Z encodes a trace t, and
- $\Pi[\pi \mapsto t] = \Pi'$.

Now, we inductively translate an lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ sentence φ into a formula $ar_{\mathfrak{T}}(\varphi)$ of second-order arithmetic. This formula has two free variables, one first-order one and one second-order one. The former encodes the position at which the formula is evaluated while the latter one encodes a variable assignment (which, as explained above, is necessary to give context for the witness trees). We construct $ar_{\mathfrak{T}}(\varphi)$ such that $\mathfrak{T} \models \varphi$ if and only if $(\mathbb{N}, +, \cdot, <, \in) \models (ar_{\mathfrak{T}}(\varphi))(0, \emptyset)$, where the empty set encodes the empty variable assignment.

- $ar_{\mathfrak{T}}(\exists (Y, \curlyvee, \psi_j^{\mathrm{con}}).\ \psi) = ar_{\mathfrak{T}}(\psi)$. Here, the free variables of $ar_{\mathfrak{T}}(\exists (Y, \curlyvee, \psi_j^{\mathrm{con}}).\ \psi)$ are the free variables of $ar_{\mathfrak{T}}(\psi)$. Thus, we ignore quantification over least fixed points, as we instead use witness trees to check membership in these fixed points.
- $ar_{\mathfrak{T}}(\forall (Y, \curlyvee, \psi_j^{\mathrm{con}}).\ \psi) = ar_{\mathfrak{T}}(\psi)$. Here, the free variables of $ar_{\mathfrak{T}}(\forall (Y, \curlyvee, \psi_j^{\mathrm{con}}).\ \psi)$ are the free variables of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\exists \pi \in X_a.\ \psi) = \exists Z_\pi.\ \exists A'.\ \varphi_{\mathit{isTrace}}(Z_\pi) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi) \wedge ar_{\mathfrak{T}}(\psi)$. Here, the free second-order variable of $ar_{\mathfrak{T}}(\exists \pi \in X_a.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\exists \pi \in X_a.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\forall \pi \in X_a.\ \psi) = \forall Z_\pi.\ \forall A'.\ (\varphi_{\mathit{isTrace}}(Z_\pi) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi)) \rightarrow ar_{\mathfrak{T}}(\psi)$. Here, the free second-order variable of $ar_{\mathfrak{T}}(\forall \pi \in X_a.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\forall \pi \in X_a.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\exists \pi \in X_d.\ \psi) = \exists Z_\pi.\ \exists A'.\ \varphi_{\mathfrak{T}}(Z_\pi) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi) \wedge ar_{\mathfrak{T}}(\psi)$. Here, the free second-order variable of $ar_{\mathfrak{T}}(\exists \pi \in X_d.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\exists \pi \in X_d.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\forall \pi \in X_d.\ \psi) = \forall Z_\pi.\ \forall A'.\ (\varphi_{\mathfrak{T}}(Z_\pi) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi)) \rightarrow ar_{\mathfrak{T}}(\psi)$. Here, the free second-order variable of $ar_{\mathfrak{T}}(\forall \pi \in X_d.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\forall \pi \in X_d.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\exists \pi \in Y_j.\ \psi) = \exists Z_\pi.\ \exists A'.\ \varphi_{\mathit{hasTree}}(A, Z_\pi, j) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi) \wedge ar_{\mathfrak{T}}(\psi)$, where $\varphi_{\mathit{hasTree}}(A, Z_\pi, j)$ is a formula of second-order arithmetic that captures the existence of a witness tree for the trace being encoded by Z_π being in the fixed point assigned to Y_j w.r.t. the variable assignment encoded by A. Its construction is similar to the corresponding formula in the proof of Theorem 7.7, but we replace the free third-order variable X_D encoding the model there by hardcoding the set of traces of $\mathfrak{T}$ using the formula $\varphi_{\mathfrak{T}}$ from above.

  Here, the free second-order variable of $ar_{\mathfrak{T}}(\exists \pi \in Y_j.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\exists \pi \in Y_j.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\forall \pi \in Y_j.\ \psi) = \forall Z_\pi.\ \forall A'.\ (\varphi_{\mathit{hasTree}}(A, Z_\pi, j) \wedge \varphi^{\pi}_{\mathit{update}}(A, A', Z_\pi)) \rightarrow ar_{\mathfrak{T}}(\psi)$. Here, the free second-order variable of $ar_{\mathfrak{T}}(\forall \pi \in Y_j.\ \psi)$ is A while A' is the free second-order variable of $ar_{\mathfrak{T}}(\psi)$. The free first-order variable of $ar_{\mathfrak{T}}(\forall \pi \in Y_j.\ \psi)$ is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\psi_1 \vee \psi_2) = ar_{\mathfrak{T}}(\psi_1) \vee ar_{\mathfrak{T}}(\psi_2)$. Here, we require that the free variables of $ar_{\mathfrak{T}}(\psi_1)$ and $ar_{\mathfrak{T}}(\psi_2)$ are the same (which can always be achieved by variable renaming), which are then also the free variables of $ar_{\mathfrak{T}}(\psi_1 \vee \psi_2)$.
- $ar_{\mathfrak{T}}(\neg\psi) = \neg ar_{\mathfrak{T}}(\psi)$. Here, the free variables of $ar_{\mathfrak{T}}(\neg\psi)$ are the free variables of $ar_{\mathfrak{T}}(\psi)$.

- $ar_{\mathfrak{T}}(\mathbf{X}\,\psi) = \exists i'(i' = i + 1) \wedge ar_{\mathfrak{T}}(\psi)$, where i' is the free first-order variable of $ar_{\mathfrak{T}}(\psi)$ and i is the free first-order variable of $ar_{\mathfrak{T}}(\mathbf{X}\,\psi)$. The free second-order variable of $ar_{\mathfrak{T}}(\mathbf{X}\,\psi)$ is equal to the free second-order variable of $ar_{\mathfrak{T}}(\psi)$.
- $ar_{\mathfrak{T}}(\psi_1 \,\mathbf{U}\, \psi_2) = \exists i_2.\ i_2 \geq i \wedge ar_{\mathfrak{T}}(\psi_2) \wedge \forall i_1.\ (i \leq i_1 \wedge i_1 < i_2) \rightarrow ar_{\mathfrak{T}}(\psi_1)$, where i_j is the free first-order variable of $ar_{\mathfrak{T}}(\psi_j)$, and i is the free first-order variable of $ar_{\mathfrak{T}}(\psi_1 \,\mathbf{U}\, \psi_2)$. Furthermore, we require that the free second-order variables of the $ar_{\mathfrak{T}}(\psi_j)$ are the same, which is then also the free second-order variable of $ar_{\mathfrak{T}}(\psi_1 \,\mathbf{U}\, \psi_2)$. Again, this can be achieved by renaming variables.
- $ar_{\mathfrak{T}}(\mathtt{p}_\pi) = pair(i, e(\mathtt{p})) \in Z_\pi$, i.e., i is the free first-order variable of $ar_{\mathfrak{T}}(\mathtt{p}_\pi)$. Note that this formula does not have a free second-order variable. For completeness, we can select an arbitrary one to serve that purpose.

Now, an induction shows that $\mathfrak{T} \models \varphi$ if and only if $(\mathbb{N}, +, \cdot, <, \in)$ satisfies $(ar(\varphi))(0, \emptyset)$, where $\emptyset$ again encodes the empty variable assignment. □

8. Related Work

As mentioned in Section 1, the complexity problems for HyperLTL were thoroughly studied [FH16, FKTZ21, FKTZ25]. For Hyper^{2}LTL, Beutner et al. mainly focused on the algorithmic aspects by providing model checking [BFFM23] and monitoring [BFFM24] algorithms. Finkbeiner et al. studied the complexity of Hyper^{2}LTL and lfp-Hyper^{2}LTL$_{\mathrm{mm}}$ over tree-shaped and acyclic structures, which contain finitely many traces [FFR26], thus all problems become decidable. None of the above studied the respective complexity problems for general structures in depth.

Logics related to Hyper^{2}LTL are asynchronous and epistemic logics. Much research has been done regarding epistemic properties [Dim08, FHMV95, LR06, vdMS99] and their relations to hyperproperties [BMP15]. However, most of this work concerns expressiveness and decidability results (e.g., [BMM24]), and not complexity analysis for the undecidable fragments. This is similar for asynchronous hyperlogics [BHNdC23, BCB$^+$21, BF23, BBST24, BPS21, BPS22, GMO21, KSV23, KSV24, KMVZ18], where most work concerns decidability results and expressive power, but not complexity analysis.

Another related logic is TeamLTL [KMVZ18], a hyperlogic for the specification of dependence and independence. Lück [Lüc20] 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 [KS21] 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. [KSV23] study set semantics for asynchronous TeamLTL, and provide positive complexity and decidability results. Gutsfeld et al. [GMOV22] 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.

Finally, the logic HyperQPTL, which extends HyperLTL by quantification over propositions [Rab16], is also related to Hyper^{2}LTL. With uniform quantification, HyperQPTL satisfiability is equivalent to truth in second-order arithmetic [RZ24] while finite-state satisfiability and model-checking have the same complexity as for HyperLTL [FH16, Rab16]. Non-uniform quantification makes HyperQPTL as expressive as second-order HyperLTL [RZ24], which implies that all three problems are equivalent to truth in third-order arithmetic.

9. Conclusion

We have investigated and settled the complexity of satisfiability, finite-state satisfiability, and model-checking for $\mathrm{Hyper}^2\mathrm{LTL}$ and its fragments $\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ and lfp-$\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$. 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^0_1-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-$\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$, in which second-order quantification degenerates to least fixed point computations, the complexity is lower (albeit still high): satisfiability is Σ^2_1-complete while finite-state satisfiability and model-checking are equivalent to truth in second-order arithmetic. All our results, but the one for lfp-$\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ satisfiability, also hold for closed-world semantics. On the other hand, lfp-$\mathrm{Hyper}^2\mathrm{LTL}_{\mathrm{mm}}$ satisfiability under closed-world semantics is "only" Σ^1_1-complete, i.e., as hard as HyperLTL satisfiability.

In future work, we aim to study the rich landscape of temporal logics for asynchronous hyperproperties. Here, the first results [RZ25a] for generalized HyperLTL with stuttering and contexts [BBST24] have been obtained: Model-checking is equivalent to truth in second-order arithmetic while satisfiability is Σ^1_1-complete.

Acknowledgments.
Co-funded by the European Union Gaëtan Regaud has been supported by the European Union. Martin Zimmermann has been supported by DIREC – Digital Research Centre Denmark. Hadar Frenkel has been supported in part by the Israel Science Foundation (grant No. 655/25).

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".

References

[BBST24] Alberto Bombardelli, Laura Bozzelli, César Sánchez, and Stefano Tonetta. Unifying asynchronous logics for hyperproperties. In Siddharth Barman and Slawomir Lasota, editors, *FSTTCS 2024*, volume 323 of *LIPIcs*, pages 14:1–14:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.FSTTCS.2024.14.

[BCB+21] 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.

[BF23] 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.

[BFFM23] 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.

[BFFM24] 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.

[BHNdC23] 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.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

[BMM24] 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.

[BMP15] 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.

[BPS21] 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.

[BPS22] 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.

[CFK+14] 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.

[CS10] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *J. Comput. Secur.*, 18(6):1157–1210, 2010. doi:10.3233/JCS-2009-0393.

[Dim08] 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.

[EH86] 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.

[FFR26] Bernd Finkbeiner, Hadar Frenkel, and Tim Rohde. Complexity of model checking second-order hyperproperties on finite structures. In *FoSSaCS 2026*, 2026. To appear. doi:10.48550/arXiv.2601.12361.

[FH16] 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.

[FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge.* MIT Press, 1995. doi:10.7551/MITPRESS/5803.001.0001.

[FKTZ21] 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.

[FKTZ25] Marie Fortin, Louwe B. Kuijer, Patrick Totzke, and Martin Zimmermann. HyperLTL satisfiability is highly undecidable, HyperCTL* is even harder. *Log. Methods Comput. Sci.*, 21(1):3, 2025. URL: https://doi.org/10.46298/lmcs-21(1:3)2025, doi:10.46298/LMCS-21(1:3)2025.

[FRS15] 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.

[FZ17] 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.

[FZ25] Hadar Frenkel and Martin Zimmermann. The complexity of second-order HyperLTL. In Jörg Endrullis and Sylvain Schmitz, editors, *CSL 2025*, volume 326 of *LIPIcs*, pages 10:1–10:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.CSL.2025.10.

[GMO21] 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.

[GMOV22] 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.

[KMVZ18] 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.

[KS21] 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.

[KSV23] 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.

[KSV24] 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.

[LR06] 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.

[Lüc20] 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.

[MZ20] 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.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *FOCS 1977*, pages 46–57. IEEE, Oct 1977. doi:10.1109/SFCS.1977.32.

[Rab16] 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/.

[Rog87] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* MIT Press, Cambridge, MA, USA, 1987.

[RZ24] Gaëtan Regaud and Martin Zimmermann. The complexity of HyperQPTL. *arXiv*, 2412.07341, 2024. doi:10.48550/ARXIV.2412.07341.

[RZ25a] Gaëtan Regaud and Martin Zimmermann. The complexity of generalized HyperLTL with stuttering and contexts. In Giorgio Bacci and Adrian Francalanza, editors, *GandALF 2025*, volume 428 of *EPTCS*, pages 161–176. Open Publishing Association, 2025. doi:10.4204/EPTCS.428.12.

[RZ25b] Gaëtan Regaud and Martin Zimmermann. The complexity of fragments of second-order HyperLTL. *arXiv*, 2501.19046, 2025. doi:10.48550/ARXIV.2501.19046.

[SC85] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985. doi:10.1145/3828.3837.

[vdMS99] 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.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, 171 Second St, Suite 300, San Francisco, CA 94105, USA, or Eisenacher Strasse 2, 10777 Berlin, Germany