On the expressiveness of hyperlogics*

Antonis Achilleos[1], Elli Anastasiadi[2], R. Govind[2], and Jana Wagemaker[1]

[1] ICE-TCS, Department of Computer Science, Reykjavik University, Iceland
[2] Department of Information Technology, Uppsala University, Sweden
antonios@ru.is, elli.anastasiadi@it.uu.se, govind.rajanbabu@it.uu.se, janaw@ru.is

Abstract

We compare the expressiveness between hyperlogics, i.e., logics interpreted over sets of traces, defined as extensions of LTL, FO, and the μ-calculus.

1 Introduction

Hyperlogics are a family of logics that started emerging 15 years ago. They were first suggested as a formalism rich enough to capture information flow security properties [5]. At their core, hyperproperties are extensions of properties of traces to properties of *sets of traces* (denoted T). Having properties of sets of traces captures situations from computer science, where a *set* of users (or executions) might exhibit some bad behavior, or might *together* assert some guarantee. The most popular hyperlogic is HyperLTL [4], the extension of LTL, which uses trace quantifiers and trace variables to refer to multiple traces. For example, the formula:

$$\forall\pi\forall\pi' G(a_\pi \equiv a_{\pi'}) \tag{1}$$

expresses that all traces must either satisfy a, or all traces must satisfy $\neg a$, at each spot. This property is trivially satisfied by any trace but can be violated when interpreted over sets.

An important question about logics of this type is whether they maintain (or somehow lift) language-theoretic, complexity, or expressive equivalence results from their non-hyper counterparts. For example, we know that every satisfiable LTL formula has a model that is an ultimately periodic trace [12]. On an even more fundamental level, Kamp's seminal theorem [9] (in the formulation due to Gabbay et al. [8]) states that LTL is expressively equivalent to first-order logic FO[$<$] over the natural numbers with order.

FO[$<, \mathbf{E}$], i.e. FO[$<$], equipped with the "equal level" arrity 2 predicate $\mathbf{E}$, was proposed by [7] to capture the expressive power of HyperLTL. This logic is essentially interpreted over multiple copies of the natural numbers with order, and thus the models of its sentences are sets of traces, just like with hyperlogics. Variables in FO[$<, \mathbf{E}$] are mapped to "places", i.e., pairs of a trace and an index in that trace, as opposed to a simple index in the case of FO[$<$]. $\mathbf{E}(x, y)$ holds only when the two quantified variables x, y are mapped to the same position of possibly different traces. For example, Property 1 is formulated as:

$$\forall x.\forall y.\mathbf{E}(x, y) \rightarrow (\mathbf{P}_a(x) \equiv \mathbf{P}_a(y)) \tag{2}$$

where $\mathbf{P}_a(x)$ is a unary predicate that encodes the occurrence of symbol a at position x. It turns out that this logic is strictly more expressive than HyperLTL [7]. Although the authors of that work do propose a logic (called HyperFO) that is expressively equivalent to HyperLTL by restricting FO[$<, \mathbf{E}$], there is still no temporal counterpart to the full FO[$<, \mathbf{E}$] logic.

*The work reported in this paper is supported by the project 'Mode(l)s of Verification and Monitorability' (MoVeMent) (grant no 217987) of the Icelandic Research Fund.

A property that is expressible in FO$[<, \mathbf{E}]$ but not HyperLTL, is: "there exists an $n \in \mathbb{N}$, such that $t(n) = a$, for all $t \in T$". This property is a *consensus property* (and very relevant to the context of hyperlogics), and it is also not expressible in HyperCTL* [3]. Thus, to produce a temporal equivalent of FO$[<, \mathbf{E}]$, one would have to look at more expressive logics. Such a logic could be the extension of μHML on hypertraces, which was recently studied by the authors and collaborators [1]. In this work, we discuss the spectrum of expressiveness between these three logics and prove that 1) the gap of expressivity between LTL and μHML is preserved in their hyper extensions, and that 2) FO$[<, \mathbf{E}]$ does not cover the full Hyper-μHML in expressiveness.

2 Preliminaries

Let AP denote the set of all atomic propositions. An atomic proposition a, where $a \in \mathsf{AP}$, expresses some fact about states. Thus, all the propositional information for a state is described by an *action* $\alpha \in \text{Act} = 2^{\mathsf{AP}}$. TR stands for Act^ω, the set of all traces. A *hypertrace* T is a subset of TR, and we denote with $\mathsf{HTrc} = 2^{\mathsf{TR}}$ the set of hypertraces. Let $t \in \mathsf{TR}$ be a trace. We use $t[i]$ to denote the element i of t, where $i \in \mathbb{N}$. Hence, $t[0]$ is the first element of t. We write $t[0, i]$ to denote the prefix of t up to and including element i, and $t[i, \infty]$ to denote the infinite suffix of t beginning with element i. We can also lift the suffix notation to hypertraces $T \in \mathsf{HTrc}$: $T[i, \infty] == \{t[i, \infty] \in \mathsf{TR} \mid t \in T\}$. In what follows, we consider formulas with trace variables and trace quantifiers. We will call a formula *closed* if a trace quantifier binds every occurrence of a trace variable.

HyperLTL We introduce here the logic HyperLTL as it was described originally in [4].

$$
\begin{array}{rclcccccccc}
\psi & ::= & \exists \pi.\, \psi & \mid & \forall \pi.\, \psi & \mid & \varphi & & & & \\
\varphi & ::= & a_\pi & \mid & \neg\varphi & \mid & \varphi \vee \varphi & \mid & X\varphi & \mid & \varphi U \varphi
\end{array}
$$

True and false, written $\mathtt{tt}$ and $\mathtt{ff}$, are respectively defined as $a_\pi \vee \neg a_\pi$ and $\neg\mathtt{tt}$. The satisfaction judgment for HyperLTL formulas is written $\Pi \models_T \psi$, where T is a set of traces, and $\Pi : \mathcal{V} \to \mathsf{TR}$ is a *trace assignment* (i.e., a *valuation*), which is a partial function mapping trace variables to traces in T. Let $\Pi[\pi \mapsto t]$ denote the same function as Π, except that π is mapped to t. One can think of these semantics in two layers: one for ψ as it is, and one for φ that only depends on Π. Satisfaction is defined as follows:

$$
\begin{array}{lll}
\Pi \models_T \exists \pi.\, \psi & \text{iff} & \text{there exists } t \in T : \Pi[\pi \mapsto t] \models_T \psi \\
\Pi \models_T \forall \pi.\, \psi & \text{iff} & \text{for all } t \in T : \Pi[\pi \mapsto t] \models_T \psi \\
\Pi \models_T a_\pi & \text{iff} & a \in \Pi(\pi)[0] \\
\Pi \models_T \neg\varphi & \text{iff} & \Pi \not\models_T \varphi \\
\Pi \models_T \varphi_1 \vee \varphi_2 & \text{iff} & \Pi \models_T \varphi_1 \text{ or } \Pi \models_T \varphi_2 \\
\Pi \models_T X\varphi & \text{iff} & \Pi[1, \infty] \models_T \varphi \\
\Pi \models_T \varphi_1 U \varphi_2 & \text{iff} & \text{there exists } i \geq 0 : \Pi[i, \infty] \models_T \varphi_2 \\
 & & \quad \text{and for all } 0 \leq j < i \text{ we have } \Pi[j, \infty] \models_T \varphi_1
\end{array}
$$

If $\Pi_\emptyset \models_T \varphi$ holds for the empty assignment $\Pi_\emptyset$, then T *satisfies* φ.

Hyper-μHML We present Hyper-μHML as a logic to specify hyperproperties. Hyper-μHML extends the linear-time interpretation of μHML [10, 11, 13] by allowing quantification over traces. We assume two disjoint and countably infinite sets Π and V of trace variables and recursion variables, ranged over by π and x, respectively.

Definition 1. *Formulae of Hyper-μHML are constructed as follows:*

$$\begin{aligned}\varphi ::= \mathsf{tt} \mid \mathsf{ff} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \max x.\varphi \mid \min x.\varphi \mid x \\ \mid \exists\pi.\varphi \mid \forall\pi.\varphi \mid \pi = \pi \mid \pi \neq \pi \mid [a_\pi]\varphi \mid \langle a_\pi\rangle\varphi\end{aligned}$$

To help us simplify the definition of the semantics, we consider hypertraces of a fixed size k, and we identify hypertraces with k-tuples $T = (T(0), T(1), \ldots, T(k-1)) \in \mathsf{HTrc}_k = \mathsf{TR}^k$. The semantics of a Hyper-μHML formula φ is defined for each such k by exploiting two partial functions: $\rho\colon V \rightharpoonup 2^{\mathsf{HTrc}_k}$, that assigns a set of hypertraces of size k to all free recursion variables of φ, and $\sigma\colon \Pi \rightharpoonup \{0, 1, \ldots, k-1\}$, that assigns a position in each tuple T to each free trace variable of φ. The semantics is given by:

$$\begin{aligned}
&[\![\mathsf{tt}]\!]^\rho_\sigma = \mathsf{HTrc}_k \qquad [\![\mathsf{ff}]\!]^\rho_\sigma = \emptyset && [\![x]\!]^\rho_\sigma = \rho(x)\\
&[\![\varphi \wedge \varphi']\!]^\rho_\sigma = [\![\varphi]\!]^\rho_\sigma \cap [\![\varphi']\!]^\rho_\sigma && [\![\varphi \vee \varphi']\!]^\rho_\sigma = [\![\varphi]\!]^\rho_\sigma \cup [\![\varphi']\!]^\rho_\sigma\\
&[\![\max\ x.\psi]\!]^\rho_\sigma = \bigcup\{S \mid S \subseteq [\![\psi]\!]^{\rho[x\mapsto S]}_\sigma\} && [\![\min\ x.\psi]\!]^\rho_\sigma = \bigcap\{S \mid S \supseteq [\![\psi]\!]^{\rho[x\mapsto S]}_\sigma\}\\
&[\![\exists\pi.\varphi]\!]^\rho_\sigma = \bigcup_{i=0}^{k-1} [\![\varphi]\!]^\rho_{\sigma[\pi\mapsto i]} && [\![\forall\pi.\varphi]\!]^\rho_\sigma = \bigcap_{i=0}^{k-1} [\![\varphi]\!]^\rho_{\sigma[\pi\mapsto i]}\\
&[\![\pi = \pi']\!]^\rho_\sigma = \{T \in \mathsf{HTrc}_k \mid T(\sigma(\pi) = T(\sigma(\pi'))\} && [\![\pi \neq \pi']\!]^\rho_\sigma = \mathsf{HTrc}_k \setminus [\![\pi = \pi']\!]^\rho_\sigma\\
&[\![[a_\pi]\varphi]\!]^\rho_\sigma = \{T \mid \sigma(\pi)[0] = a \text{ implies } T[1,\infty] \in [\![\varphi]\!]^\rho_\sigma\}\\
&[\![\langle a_\pi\rangle\varphi]\!]^\rho_\sigma = \{T \mid \sigma(\pi)[0] = a\ \wedge\ T[1,\infty] \in [\![\varphi]\!]^\rho_\sigma)\}
\end{aligned}$$

Whenever φ is *closed*, the semantics is given by $[\![\varphi]\!]^\emptyset_\emptyset$, where $\emptyset$ denotes the partial function with empty domain, and we simply write $[\![\varphi]\!]$ instead of $[\![\varphi]\!]^\emptyset_\emptyset$. We use the standard notation $T \models \varphi$ to denote that the set of traces T satisfies φ (and similarly for $T \not\models \varphi$). As an example, consider the alphabet $\{a, b\}$. The property

$$\forall\pi.\max \mathrm{x}.(\langle \mathrm{b}_\pi\rangle \mathrm{x} \vee (\exists\pi'.(\pi' \neq \pi) \wedge \langle \mathrm{a}_{\pi'}\rangle \mathrm{x})) \tag{3}$$

means that, for every trace, whenever there is an a, there is another trace that also has a.

The logic FO$[<, \mathbf{E}]$

Definition 2 (From [7])**.** *FO$[<, \mathbf{E}]$ is defined over the signature $\{\mathbf{E}, <\} \cup \{\mathbf{P}_a \mid a \in \mathsf{AP}\}$, i.e., with atomic formulas $x = y$, $x < y$, $\mathbf{E}(x, y)$, and $\mathbf{P}_a(x)$ for $a \in \mathsf{AP}$, and disjunction, conjunction, negation, and existential and universal quantification over elements.*

The semantics of this logic is the standard semantics of FO and comes in accordance with the semantics of FO$[<]$. We interpret FO$[<, \mathbf{E}]$ formulas over a set of traces $T \subseteq \textsc{Act}^\omega$ and an interpretation $I : \mathcal{V} \to T \times \mathbb{N}$, which assigns a tuple (t, n) to each variable x, with $t \in T$, $n \in \mathbb{N}$. Given a set of traces T, the operations $<, \mathbf{E}$, and $\mathbf{P}_a, a \in \mathsf{AP}$ are interpreted as:

- $<^T := \{((t, n), (t, n')) \mid t \in T \text{ and } n < n' \in \mathbb{N}\}$,
- $\mathbf{E}^T := \{((t, n), (t', n)) \mid t, t' \in T \text{ and } n \in \mathbb{N}\}$, and
- $\mathbf{P}^T_a := \{(t, n) \mid t \in T \text{ and } n \in \mathbb{N} \text{ and } a \in t(n)\}$.

3 Expressiveness comparisons

We start the comparison of expressiveness from the single-trace setting. Hyper-μHML is an extension of the linear-time interpretation of μHML. The logic μHML is expressive enough to strictly include LTL, and even CTL* in its usual, branching-time interpretation [2]. Quantification over traces and trace comparisons are allowed in any part of the formula, which means our syntax subsumes the syntax of HyperLTL, using straightforward translations. We show that the strictness of the inclusion of LTL in μHML is preserved for their hyper-trace extensions.

Theorem 1. *Hyper-μHML is strictly more expressive than HyperLTL.*

Proof. The simple inclusion follows from the embedding of LTL in μHML and the more liberal ability to quantify over traces. To demonstrate the strictness of this inclusion, we bring forward two arguments. First, we reference the work of Wolper in [14], which describes formulas of μHML that require an event a to occur at least in all even positions of a trace. The following μHML formula describes exactly this (over the set of actions a, b):

$$\varphi_e := \max x.([a]\langle a\rangle x \wedge [b]\langle a\rangle x) \tag{4}$$

Let φ_{h_e} be the formula that occurs if one adds an existential trace quantifier $\exists\pi$ at the beginning of φ_e, and replaces all modalities with π-indexed ones:

$$\varphi_{h_e} := \exists\pi.\, \max x.([a_\pi]\langle a_\pi\rangle x \wedge [b_\pi]\langle a_\pi\rangle x), \tag{5}$$

whose evaluation over singleton hypertraces coincides with the evaluation of φ_e. Assume now that a formula φ_{h-LTL} is expressively equivalent to φ_{h_e} over hypertraces. We would like to use this to extract an LTL formula that is expressively equivalent to φ_e. We cannot trivially claim that φ_{h-LTL} only contains a single quantifier $\exists\pi$. Instead, though, we know that over singleton hypertraces, say for $T = \{t_0\}$, $T \models \varphi_{h-LTL}$ iff $T \models \varphi_{h_e}$. Since T contains only a single trace, we know that all the trace variables in φ_{h-LTL} must be mapped to t_0. Consequently, all propositional variables that occur in φ_{h-LTL} must be mapped to t_0. Therefore, for this variable mapping, we get an LTL formula that expresses exactly that a trace (t_0) satisfies Wolper's property. We then replace all propositional variables with non-trace quantified ones and, remove all quantifiers, which brings us to plain LTL, and arrive at a contradiction. □

Remark 1. *In the proof above, we demonstrate that the property "there exists a trace for which a holds on at least all even positions" is not expressible in HyperLTL but is expressible in Hyper-μHML. The same argument can be repeated for any period k.*

Furthermore, we demonstrate that Hyper-μHML is more expressive than FO$[<, \mathbf{E}]$. Intuitively, one factor that gives Hyper-μHML significant expressive power is its ability to use quantifiers at any part of the syntax. This is also allowed in other temporal logics, such as, for example, $HyperCTL^*$. A key difference is that Hyper-μHML can nest quantifiers within a fixed-point operator. For example, we see that the property from Example 3 will potentially spawn an unbounded number of quantifiers due to the recursion unfolding caused by encountering a events. We argue that due to the ability to nest quantifiers at any point of our syntax, Hyper-μHML is more expressive than HyperLTL, and it can express properties that $HyperCTL^*$ and FO$[<, \mathbf{E}]$ cannot.

Theorem 2. *Hyper-μHML contains properties not expressible in $HyperCTL^*$ and $FO[<, \mathbf{E}]$.*

Proof. For the first part, we refer the reader to the work of Bozzelli, Maubert, and Pinchinat [3], who show that the property "there is an $n \geq 0$ such that $a \neq t(n)$ for every $t \in T$" is not expressible in $HyperCTL^*$. In Hyper-μHML, this property is expressible (over the set of actions $\{a, b\}$) with the formula:

$$\min x.((\forall \pi \langle b_\pi \rangle \mathsf{tt}) \vee (\forall \pi'([a_{\pi'}]x \wedge [b_{\pi'}]x))) \ . \tag{6}$$

In this formula, either all traces have b, or all traces take a step. Since this happens within the scope of a minimal fix-point, we get that to satisfy the formula, this process needs to terminate, and thus, we get exactly the property we wanted.

For the second part, we use Wolper's property φ_{h_e} (Property 5). Due to the expressive equivalence of LTL and FO[<] (from [8]), we can use a similar proof as for Theorem 1. The key is after projecting an FO[$<, \mathbf{E}$] formula over uniset hypertraces to replace all occurrences of $\mathbf{E}(x, y)$ with $x = y$, as the two predicates coincide over such models. This leads us again to a property in FO[<] which expresses φ_E (Property 4, and we get a contradiction from the expressive equivalence of FO[<] and LTL (from [8]). □

4 Conclusion and future work

We have shown that the expressive power of Hyper-μHML is above HyperLTL, and possibly above (or at least incomparable with) FO[$<, \mathbf{E}$]. We would like to extend Theorem 1 to fully characterize whether FO[$<, \mathbf{E}$] is contained in Hyper-μHML. In case they are incomparable, it would suffice to produce a property in FO[$<, \mathbf{E}$] that is not expressible in Hyper-μHML. Any properties we tried to that end, however, were not able to distinguish the two logics. Thus, we are left with the conjecture that Hyper-μHML subsumes FO[$<, \mathbf{E}$]. At this point, we have partially produced an embedding of FO[$<, \mathbf{E}$] into Hyper-μHML, and we believe one does exist. Finishing such an embedding would also imply that to produce a temporal equivalent of FO[$<, \mathbf{E}$], one would need to find a middle ground between the syntax of Hyper-μHML and HyperLTL. On the other hand, a non-temporal equivalent of Hyper-μHML in the style of FO[$<, \mathbf{E}$] could be MSO over hypertraces (and possibly with the equality predicate $\mathbf{E}$).

In the future, we aim to answer the following questions. The first is to fully produce such an encoding and prove its correctness. The second is to find a temporal equivalent of FO[$<, \mathbf{E}$]. We believe this is not a trivial question at all. For instance, increasing the quantification power of HyperLTL to allow non-normalized formulae would not be enough since HyperCTL*, which allows this, cannot express the consensus property. Moreover, we are interested in finding a classical logic characterization of Hyper-μHML. As we discussed, HyperLTL is expressively equivalent to a fragment of FO[$<, \mathbf{E}$], as proven in [7], and as we have shown Hyper-μHML is not the temporal counterpart of FO[$<, \mathbf{E}$]. We would like to fill this expressiveness gap. A good candidate for this could be some version of MSO over sets of traces. Indeed, there is work already done in this direction (see [6]), although so far, there seems to be no logic that can capture the properties 5, or 6. Finally, just like it is known that μHML corresponds to ω-regular languages, it would be interesting to find language-theoretic counterparts of HyperLTL, Hyper-μHML, and FO[$<, \mathbf{E}$].

References

[1] Luca Aceto, Antonis Achilleos, Elli Anastasiadi, and Adrian Francalanza. Monitoring hyperproperties with circuits. In Mohammad Reza Mousavi and Anna Philippou, editors, *Formal Techniques for Distributed Objects, Components, and Systems - 42nd IFIP WG 6.1 International Conference,*

FORTE 2022, Held as Part of the 17th International Federated Conference on Distributed Computing Techniques, DisCoTec 2022, Lucca, Italy, June 13-17, 2022, Proceedings, volume 13273 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2022.

[2] Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. Adventures in monitorability: From branching to linear time and back again. *Proceedings of the ACM on Programming Languages*, 3(POPL):52:1–52:29, 2019.

[3] Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Unifying hyper and epistemic temporal logics. In Andrew M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2015.

[4] 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, *Principles of Security and Trust - Third International Conference, POST 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8414 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2014.

[5] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. volume 18, pages 1157–1210, 2010.

[6] Norine Coenen, Bernd Finkbeiner, Christopher Hahn, and Jana Hofmann. The hierarchy of hyperlogics. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2019.

[7] Bernd Finkbeiner and Martin Zimmermann. The first-order logic of hyperproperties. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[8] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne, editors, *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, USA, January 1980*, pages 163–173. ACM Press, 1980.

[9] Hans Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, Ucla, 1968.

[10] Dexter C. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[11] Kim G. Larsen. Proof Systems for Satisfiability in Hennessy-Milner Logic with recursion. *Theoretical Computer Science*, 72(2):265 – 288, 1990.

[12] D.E. Muller, A. Saoudi, and P.E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *[1988] Proceedings. Third Annual Symposium on Logic in Computer Science*, pages 422–427, 1988.

[13] Moshe Y. Vardi. A Temporal Fixpoint Calculus. In *POPL*, pages 250–259, New York, NY, USA, 1988. ACM.

[14] Pierre Wolper. Temporal logic can be more expressive. In *FOCS*, pages 340–348. IEEE Computer Society, 1981.