Multiplayer Games With Incomplete Information for Hyperproperty Verification

Extended Abstract

Raven Beutner
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

Bernd Finkbeiner
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

ABSTRACT

Hyperproperties are system properties that relate multiple execution traces in a system and can, e.g., express security policies, knowledge properties, path planning, and robustness requirements. Logics for expressing temporal hyperproperties - such as HyperLTL - extend LTL by quantifying over executions of a system. Many properties used in practice require one or more *quantifier alternations*, which presents a major challenge for verification. Complete verification methods require a system complementation for each quantifier alternation, making it infeasible in practice. A cheaper method interprets the verification of a HyperLTL formula as a two-player parity game between universal and existential quantifiers. This game-based approach is very efficient and allows for interactive proofs, but is limited to $\forall^*\exists^*$ HyperLTL formulas, leaving important properties out of reach. In this paper, we argue that we can extend the game-based verification approach to *arbitrary* HyperLTL formulas, by utilizing *multiple players* and *incomplete information.*

KEYWORDS

LTL; HyperLTL; Game-Based Verification; Incomplete Information; Hierarchical Information

ACM Reference Format:
Raven Beutner and Bernd Finkbeiner. 2025. Multiplayer Games With Incomplete Information for Hyperproperty Verification: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

In 2004, Clarkson and Schneider [21] coined the term *hyperproperties* for the rich class of system requirements that relate multiple executions. In contrast to *trace properties* – i.e., properties over individual executions, expressed, e.g., in linear-time temporal logics (LTL) [36] – hyperproperties can express important properties related to information flow, robustness, and security.

While originating in the study of information flow properties, hyperproperties have since established themself as a much more general framework that captures properties from many different areas, including, e.g., knowledge properties in multi-agent systems (MAS) [12, 13, 26, 38]. As an example, we consider some MAS with agents $1, \dots, n$, and some LTL property ψ, and assume that we want to verify that there exists at least one execution of the MAS such that agent i *knows* that ψ holds. Formally, *knowing* that ψ holds on some execution trace t means that ψ must hold on all traces that are indistinguishable from t for agent i (cf. [26]), which *is* a hyperproperty. To express such properties, we can either employ dedicated knowledge operators (see, e.g., LTL_K [26]), or use logics that can natively express (much more general) hyperproperties. For example, we can express the above requirement in HyperLTL – an extension of LTL with explicit quantification over execution traces [20] – as follows

$$\exists \pi_1.\forall \pi_2.\, \pi_1 \sim_i \pi_2 \rightarrow \psi[\pi_2], \tag{K_1}$$

where we write $\psi[\pi_2]$ to indicate that ψ holds on trace π_2, and $\pi_1 \sim_i \pi_2$ denotes that executions π_1 and π_2 appear identical under agent i's observations of the MAS. The above formula thus states that there exists some trace π_1, such that all traces π_2 that appear identical to agent i satisfy ψ.

Using the flexibility of HyperLTL's trace quantification, we can also express nested knowledge properties. For example, we can express that – on some execution – agent i knows that agent j does *not* know that ψ holds:

$$\begin{aligned}\exists \pi_1.\forall \pi_2.\exists \pi_3.\exists \pi_4.\, \pi_1 \sim_i \pi_2 \rightarrow \\ \big(\pi_2 \sim_j \pi_3 \land \pi_2 \sim_j \pi_4 \land \psi[\pi_3] \land \neg\psi[\pi_4]\big).\end{aligned} \tag{K_2}$$

That is, for every trace π_2 that agent i cannot distinguish from π_1, there exists two traces π_3, π_4 that agent j cannot distinguish from π_2, one of which satisfies ψ and one violates ψ. The existence of π_3, π_4 ensures that agent j does not know whether ψ holds on π_2.

Apart from knowledge properties, HyperLTL can express a wide range of other properties that are relevant in MASs, including fairness (where we, e.g., need to compare multiple executions of applicants that only differ in sensitive attributes) [40], robustness (where we, e.g., need to check that any pair of executions with similar input also have similar output) [15, 19, 35], opacity (where the behavior of some agent should not reveal secret information, i.e., the same observable behavior can be generated by a trace without secret information) [2, 33], and optimal planning (where some path is at least as good as all alternatives) [39].

Verification of HyperLTL. We are interested in verifying that a given finite-state transition system $\mathcal{T}$ satisfies a given HyperLTL formula φ. Unsurprisingly, the quantifier structure of φ directly impacts the complexity of the verification problem. Alternation-free formulas (i.e., formulas where all quantifiers are of the same type) can be checked very efficiently on the self-composition of the system [4, 27]. Verification gets much more challenging when

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

the formula includes quantifier alternations, e.g., K_1, K_2. Here, complete approaches require an expensive system complementation or inclusion check for each alternation in the formula [10, 20, 27].

A cheaper (but incomplete) verification method for $\forall^*\exists^*$ formulas (i.e., formulas where an arbitrary number of universal quantifiers is followed by an arbitrary number of existential quantifiers) is based on a game-based interpretation [8, 11, 22]. Concretely, we interpret the verification of a HyperLTL formula $\forall\pi_1.\exists\pi_2.\psi$ (where ψ is the LTL body) as a game between two players. A *refuter* controls the universally quantified trace by moving through a copy of the underlying system, thereby constructing a concrete trace for π_1. The *verifier* reacts to the moves by the refuter and moves through a separate copy of the system, thereby producing a concrete trace for π_2. The goal of the verifier is to ensure that π_1 and π_2, together, satisfy ψ. If we can find such a strategy for the verifier, we can conclude that the $\forall^*\exists^*$ property is satisfied. We can think of the verifier's strategy as providing a step-wise construction of a concrete witness trace for π_2, no matter what trace we choose for π_1. This game-based approach is efficient (polynomial in the system size), and, perhaps even more importantly, allows for interactive proofs and easy-to-check certificates. For example, we can use the game-based framework to let the user construct a strategy manually (potentially supported by a proof assistant) [23], allowing verification even in situations where automated techniques fail. Likewise, we can use a winning strategy for the verifier as an (easy-to-check) certificate that the property is indeed satisfied [14].

Unsoundness Beyond $\forall^*\exists^*$. The game-based verification approach has proven useful in many applications [8, 22], including the verification of infinite-state systems [9, 23, 32]. However, the approach is limited to $\forall^*\exists^*$ properties. Intuitively, as soon as we consider properties beyond $\forall^*\exists^*$, the step-wise selection of the traces leads to unsoundness, i.e., cases where a winning strategy exists even though the property is violated. This unsoundness occurs for simple $\exists^*\forall^*$ properties like K_1 and also for properties with multiple quantifier alternations like K_2. Consequently, for properties outside the $\forall^*\exists^*$ fragment,no efficient verification approach exists, nor does there exist any approach that enables interactive proofs or supports easy-to-check witnesses.

The Solution: Incomplete Information. We propose a novel adaption of the game-based method that allows for sound verification of *arbitrary* HyperLTL formulas. Our key observation is that the unsoundess of the game is directly linked to the players' information. In a $\exists\pi_1.\forall\pi_2.\,\psi$ property, we must find a *unique* witness trace for π_1 that works for all possible choices of π_2. However, if we view verification as a game, the players construct the witness traces in a step-wise fashion. The verifier – who is in control of the existentially quantified π_1 – can thus observe the previous steps of the refuter. In particular, the witness trace for π_1 (which is constructed by the verifier) can depend on prefixes of π_2 (constructed by the refuter), leading to situations where the verifier can win the game, even though no witness for π_1 exists.

Consequently, our key conceptual contribution is the observation that, to achieve soundness, we need to reason about *incomplete information*. We replace the simple $\forall^*\exists^*$ two-player game (played under full information) by a multiplayer game where each player corresponds to one trace variable quantified in the formula. These players construct their respective trace step-wise; similar to the $\forall^*\exists^*$ game. Within the resulting game, we are interested in the joint strategic ability of all players in control of an existentially quantified trace, i.e., we search for strategies for all players that correspond to existentially quantified traces (which we can think of as the verifier coalition) that, together, ensure that the LTL body of the formula is satisfied no matter how players in control of universally quantified traces (the refuter coalition) behave. To ensure soundness of this game, we carefully design an observation model for each player. That is, a player constructing some trace cannot observe the global state of the game (which would lead to unsoundness) but has a limited view on the behavior of the other players. Intuitively, our observation model ensures that the player controlling some trace π can only observe the behavior of the players that construct the traces quantified *before* π.

Example 1. *Consider the $\exists\forall\exists\exists$ property K_2 from before. In our game, we use 4 players controlling traces $\pi_1, \pi_2, \pi_3, \pi_4$, respectively. The game maintains the current system state for all traces and uses an automaton to track whether the traces generated during the game satisfy the LTL body. In each round, the players update their current state by moving along some transition of the underlying system. Using incomplete information, we ensure that the strategy controlling a trace π_i only depends on the traces quantified before trace π_i. For example, the player controlling π_1 plays oblivious, i.e., does not observe the behavior of any other player; the player controlling π_4 can observe the behavior of all other players.*

We obtain a multiplayer game played under incomplete information that, if won by the verifier coalition (i.e., all players that control existentially quantified traces), ensures that the formula holds on the given system.

Hierarchical Information. In general, multiplayer games under incomplete information are undecidable. However, the resulting verification game falls in a well-known class of games whose winner can be computed effectively. Namely, games where the information of the players is *hierarchical*, i.e., the players can be totally ordered according to their information [5–7, 18, 29, 34].

Potential Applications. Similar to the full-information game for $\forall^*\exists^*$ properties, our game-based approach can be used for interactive verification and certificate generation of – now *arbitrary* – HyperLTL formulas. Moreover, our approach allows us to leverage the extensive research on agent behavior under incomplete information (studied, e.g., in the MAS community) for automated verification. For example, our results allow us to apply techniques developed for partially observable non-deterministic (POND) planning [16], multi-agent planning [28, 30], multi-agent reinforcement learning [17], and multi-agent partially observable Markov decision processes (POMDP) [1, 37, 41]. Moreover, our approach facilitates the verification of hyperproperties on infinite-state systems. For such systems, complementation is impossible, but infinite-state game solvers exist [3, 24, 25, 31].

ACKNOWLEDGEMENTS

This work was supported by the European Research Council (ERC) Grant HYPER (101055412), and by the German Research Foundation (DFG) as part of TRR 248 (389792660).

REFERENCES

[1] Christopher Amato and Frans A. Oliehoek. 2015. Scalable Planning and Learning for Multiagent POMDPs. In *Conference on Artificial Intelligence, AAAI 2015*. https://doi.org/10.1609/AAAI.V29I1.9439

[2] Mahathi Anand, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2024. Verification of Hyperproperties for Dynamical Systems via Barrier Certificates. *IEEE Trans. Autom. Control.* (2024). https://doi.org/10.1109/TAC.2024.3384448

[3] Christel Baier, Norine Coenen, Bernd Finkbeiner, Florian Funke, Simon Jantsch, and Julian Siber. 2021. Causality-Based Game Solving. In *International Conference on Computer Aided Verification, CAV 2021*. https://doi.org/10.1007/978-3-030-81685-8_42

[4] Gilles Barthe, Pedro R. D'Argenio, and Tamara Rezk. 2011. Secure information flow by self-composition. *Math. Struct. Comput. Sci.* (2011). https://doi.org/10.1017/S0960129511000193

[5] Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. 2021. Strategy Logic with Imperfect Information. *ACM Trans. Comput. Log.* (2021). https://doi.org/10.1145/3427955

[6] Dietmar Berwanger, Krishnendu Chatterjee, Martin De Wulf, Laurent Doyen, and Thomas A. Henzinger. 2010. Strategy construction for parity games with imperfect information. *Inf. Comput.* (2010). https://doi.org/10.1016/J.IC.2009.09.006

[7] Dietmar Berwanger, Anup Basil Mathew, and Marie van den Bogaard. 2018. Hierarchical information and the synthesis of distributed strategies. *Acta Informatica* (2018). https://doi.org/10.1007/S00236-017-0306-5

[8] Raven Beutner and Bernd Finkbeiner. 2022. Prophecy Variables for Hyperproperty Verification. In *Computer Security Foundations Symposium, CSF 2022*. https://doi.org/10.1109/CSF54842.2022.9919658

[9] Raven Beutner and Bernd Finkbeiner. 2022. Software Verification of Hyperproperties Beyond k-Safety. In *International Conference on Computer Aided Verification, CAV 2022*. https://doi.org/10.1007/978-3-031-13185-1_17

[10] Raven Beutner and Bernd Finkbeiner. 2023. AutoHyper: Explicit-State Model Checking for HyperLTL. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2023*. https://doi.org/10.1007/978-3-031-30823-9_8

[11] Raven Beutner and Bernd Finkbeiner. 2024. Non-deterministic Planning for Hyperproperty Verification. In *International Conference on Automated Planning and Scheduling, ICAPS 2024*. https://doi.org/10.1609/ICAPS.V34I1.31457

[12] Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. 2023. Second-Order Hyperproperties. In *International Conference on Computer Aided Verification, CAV 2023*. https://doi.org/10.1007/978-3-031-37703-7_15

[13] Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. 2024. Monitoring Second-Order Hyperproperties. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024*. https://doi.org/10.5555/3635637.3662865

[14] Raven Beutner, Bernd Finkbeiner, and Angelina Göbl. 2024. Visualizing Game-Based Certificates for Hyperproperty Verification. In *International Symposium on Formal Methods, FM 2024*. https://doi.org/10.1007/978-3-031-71177-0_5

[15] Sebastian Biewer, Rayna Dimitrova, Michael Fries, Maciej Gazda, Thomas Heinze, Holger Hermanns, and Mohammad Reza Mousavi. 2022. Conformance Relations and Hyperproperties for Doping Detection in Time and Space. *Log. Methods Comput. Sci.* (2022). https://doi.org/10.46298/LMCS-18(1:14)2022

[16] Ronen I. Brafman, Guy Shani, and Shlomo Zilberstein. 2013. Qualitative Planning under Partial Observability in Multi-Agent Domains. In *Conference on Artificial Intelligence, AAAI 2013*. https://doi.org/10.1609/AAAI.V27I1.8643

[17] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part C* (2008). https://doi.org/10.1109/TSMCC.2007.913919

[18] Krishnendu Chatterjee and Laurent Doyen. 2010. The Complexity of Partial-Observation Parity Games. In *International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2010*. https://doi.org/10.1007/978-3-642-16242-8_1

[19] Swarat Chaudhuri, Sumit Gulwani, and Roberto Lublinerman. 2012. Continuity and robustness of programs. *Commun. ACM* (2012). https://doi.org/10.1145/2240236.2240262

[20] Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. 2014. Temporal Logics for Hyperproperties. In *International Conference on Principles of Security and Trust, POST 2014*. https://doi.org/10.1007/978-3-642-54792-8_15

[21] Michael R. Clarkson and Fred B. Schneider. 2008. Hyperproperties. In *Computer Security Foundations Symposium, CSF 2008*. https://doi.org/10.1109/CSF.2008.7

[22] Norine Coenen, Bernd Finkbeiner, César Sánchez, and Leander Tentrup. 2019. Verifying Hyperliveness. In *International Conference on Computer Aided Verification, CAV 2019*. https://doi.org/10.1007/978-3-030-25540-4_7

[23] Arthur Correnson and Bernd Finkbeiner. 2025. Coinductive Proofs for Temporal Hyperliveness. *Proc. ACM Program. Lang.* POPL (2025). https://doi.org/10.1145/3704889

[24] Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. 2001. Symbolic Algorithms for Infinite-State Games. In *International Conference on Concurrency Theory, CONCUR 2001*. https://doi.org/10.1007/3-540-44685-0_36

[25] Marco Faella and Gennaro Parlato. 2023. Reachability Games Modulo Theories with a Bounded Safety Player. In *Conference on Artificial Intelligence, AAAI 2023*. https://doi.org/10.1609/AAAI.V37I5.25779

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

[27] Bernd Finkbeiner, Markus N. Rabe, and César Sánchez. 2015. Algorithms for Model Checking HyperLTL and HyperCTL*. In *International Conference on Computer Aided Verification, CAV 2015*. https://doi.org/10.1007/978-3-319-21690-4_3

[28] Maris F. L. Galesloot, Thiago D. Simão, Sebastian Junges, and Nils Jansen. 2024. Factored Online Planning in Many-Agent POMDPs. In *Conference on Artificial Intelligence, AAAI 2024*. https://doi.org/10.1609/AAAI.V38I16.29689

[29] Paul Gastin, Nathalie Sznajder, and Marc Zeitoun. 2009. Distributed synthesis for well-connected architectures. *Formal Methods Syst. Des.* (2009). https://doi.org/10.1007/S10703-008-0064-7

[30] Piotr J. Gmytrasiewicz and Prashant Doshi. 2005. A Framework for Sequential Planning in Multi-Agent Settings. *J. Artif. Intell. Res.* (2005). https://doi.org/10.1613/JAIR.1579

[31] Philippe Heim and Rayna Dimitrova. 2024. Solving Infinite-State Games via Acceleration. *Proc. ACM Program. Lang.* POPL (2024). https://doi.org/10.1145/3632899

[32] Shachar Itzhaky, Sharon Shoham, and Yakir Vizel. 2024. Hyperproperty Verification as CHC Satisfiability. In *European Symposium on Programming, ESOP 2024*. https://doi.org/10.1007/978-3-031-57267-8_9

[33] Isabella Mastroeni and Michele Pasqua. 2022. Verifying opacity by abstract interpretation. In *Symposium on Applied Computing, SAC 2022*. https://doi.org/10.1145/3477314.3507119

[34] Bastien Maubert and Aniello Murano. 2018. Reasoning about Knowledge and Strategies under Hierarchical Information. In *International Conference on Principles of Knowledge Representation and Reasoning, KR 2018*. https://aaai.org/ocs/index.php/KR/KR18/paper/view/17996

[35] Luan Viet Nguyen, James Kapinski, Xiaoqing Jin, Jyotirmoy V. Deshmukh, and Taylor T. Johnson. 2017. Hyperproperties of real-valued signals. In *International Conference on Formal Methods and Models for System Design, MEMOCODE 2017*. https://doi.org/10.1145/3127041.3127058

[36] Amir Pnueli. 1977. The Temporal Logic of Programs. In *Symposium on Foundations of Computer Science, FOCS 1977*. https://doi.org/10.1109/SFCS.1977.32

[37] Joris Scharpff, Diederik M. Roijers, Frans A. Oliehoek, Matthijs T. J. Spaan, and Mathijs Michiel de Weerdt. 2016. Solving Transition-Independent Multi-Agent MDPs with Sparse Interactions. In *Conference on Artificial Intelligence, AAAI 2016*. https://doi.org/10.1609/AAAI.V30I1.10405

[38] Wiebe van der Hoek and Michael J. Wooldridge. 2002. Model Checking Knowledge and Time. In *International Workshop on Model Checking of Software, SPIN 2002*. https://doi.org/10.1007/3-540-46017-9_9

[39] Yu Wang, Siddhartha Nalluri, and Miroslav Pajic. 2020. Hyperproperties for Robotics: Planning via HyperLTL. In *International Conference on Robotics and Automation, ICRA 2020*. https://doi.org/10.1109/ICRA40945.2020.9196874

[40] Yu Wang, Mojtaba Zarei, Borzoo Bonakdarpour, and Miroslav Pajic. 2019. Statistical Verification of Hyperproperties for Cyber-Physical Systems. *ACM Trans. Embed. Comput. Syst.* (2019). https://doi.org/10.1145/3358232

[41] Chongjie Zhang and Victor R. Lesser. 2011. Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs. In *Conference on Artificial Intelligence, AAAI 2011*. https://doi.org/10.1609/AAAI.V25I1.7886