Thesis Proposal:
Learning Models for Concurrency

1 Introduction

Verification as a field encompasses various techniques such as model checking that ensure a system meets a given specification. The former are often represented by abstract machines, and the latter, by temporal logic formulae. Manually modeling a system is a complex and error-prone task. Moreover, the resulting models are often difficult to maintain, especially for cyber-physical systems with restricted or heterogeneous components. This challenge motivates automatic generation of formal models, a key step of model checking being the identification of a suitable automaton for verification.

1.1 Formal learning

This is where *automata learning* comes into play, in order to automatically infer a structured representation of a system's behavior. It follows two main paradigms: *active learning*, where the learner interacts with a black-box system through queries, and *passive learning,* which infers a model from a dataset without direct interaction. Starting from the pioneering works on active learning by Angluin [4] and on passive learning by Gold and Biermann [6, 15, 16], both approaches have been extensively studied for sequential models. This has led to the development of numerous learning algorithms for automata-based models, including deterministic finite automata (DFA) and Mealy machines [17, 18, 21, 24], visibly pushdown automata (VPA) [19], timed automata [28] and weighted automata [27].

While learning has been effective for inferring automata that model sequential programs, there is a lack of learnable models suitable for *concurrent* programs. These programs require models that can effectively represent parallel behaviors and partial orders of events. To address this, various concurrency-aware models have been explored, including *Higher-Dimensional Automata* (HDA), communicating automata, asynchronous automata, pomset automata and trace models. However, existing approaches to learning concurrent models are scarce.

1.2 Higher-Dimensional Automata

In recent years, HDAs have emerged as a prominent research topic in concurrency theory, offering an automata-like formalism that captures non-interleaving concurrency with precision. Initially explored through geometrical and categorical approaches, the study of HDA has increasingly focused on language theory, particularly since [10]. Key theoretical results include a Kleene theorem [11], a Myhill-Nerode theorem [13], and a Büchi theorem [2]. *Higher-dimensional timed automata* were introduced in [9], their associated languages being examined in [1]. These results demonstrate the robustness of the theory, establishing a strong foundation for further developments, see the (i)Po(m)set Project[1].

An HDA consists of a collection of cells where events occur concurrently, structured by face maps that define their initiation and termination. Its language is expressed as a set of *interval pomsets* [14] with interfaces (interval ipomsets or *iipomsets*) [12]. Each event within an execution P of an HDA corresponds to a time interval of process activity, and executions are constructed by composing elementary steps — segments of P — that seamlessly connect to form a coherent process. This

[1] https://ulifahrenberg.github.io/pomsetproject/

composition mechanism allows events to persist across segments, ensuring continuity. Moreover, since any order extension of P remains a valid execution, HDAs languages are inherently closed under *subsumption*, meaning that every possible interleaving of an execution is accepted, facilitating partial-order reduction and potentially improving state-space exploration when modeling a system with HDAs.

One of the strengths of HDAs is their suitability for providing operational semantics to models of concurrent systems. They offer a general framework for concurrency, extending well-established models such as event structures, safe Petri nets [25], asynchronous transition systems [5, 29], and Kleene automata. Among these frameworks, Petri nets stand out as one of the most established models for concurrency. They capture various concurrency semantics through a built-in notion of resources (tokens) and are widely used in both academia and industry due to their intuitive graphical representation combined with high expressiveness. In [3], HDA and their generalizations are shown to provide an operational semantics for Petri nets and many of their extensions, including inhibitor, transfer arcs or generalized self-modifying net. These translations have been implemented in the prototype tool pn2hda[2].

For illustration, Fig. 1 compares Petri net and HDA representations of a system with two events, labeled a and b. On the left, both models capture the (mutually exclusive) interleaving of a and b, allowing only the sequential executions $a.b$ or $b.a$. On the right, they depict concurrent execution, where $a \parallel b$ forms a continuous path that traverses the surface of the filled-in square from the initial to the final node. The shape of this path encodes an interval-based scheduling of a and b, reflecting their overlapping durations.

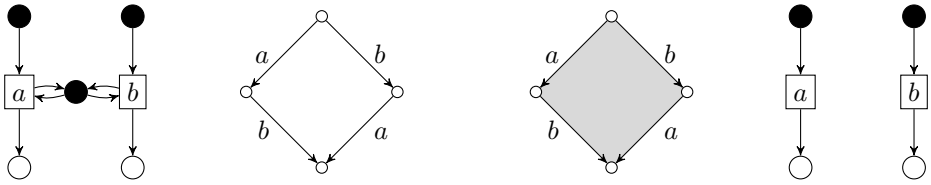

Figure 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b + b.a$; right: models for $a \parallel b$.

Another emerging application of HDA is their connection to logic and model checking. Recent developments have established formal links between HDAs and logical frameworks [2, 8], providing a foundation for verification techniques that leverage their non-interleaving structure. In particular, ω-HDAs, which accept infinite pomsets with interfaces, have laid the groundwork for reasoning about infinite behaviors [22]. These advances open new possibilities for model checking in higher-dimensional concurrency.

2 Objectives of the thesis

The objective of this thesis is to develop efficient learning algorithms for concurrent systems, with a primary focus on HDAs and their generalizations, while also exploring other concurrency models. Our main objective is to extend automata-learning techniques to concurrency-expressing structures, capturing key aspects such as causality, concurrency, and non-interleaving behaviors. A particular focus will be placed on designing learning algorithms for series-parallel pomsets, interval pomsets, and other concurrency models capable of describing a broad range of systems.

2.1 Learnability of HDA

While automata learning has been extensively studied for sequential models, its application to concurrency remains largely unexplored. Existing approaches remain limited: [7] introduces an algorithm for inferring communicating automata from message sequence charts, though its complexity limits

[2]https://gitlabev.imtbs-tsp.eu/philipp.schlehuber-caissier/pn2hda

practical applicability. [26] is the first article to study learning for pomsets, focusing on a restricted class of tree automata recognizing series-parallel pomsets. [23] iterates upon this work by extending optimized learning techniques developed for sequential traces to series-parallel pomsets. More recently, [20] proposed two active learning algorithms for sound deterministic negotiations.

A key motivation for studying HDAs in this context is their connection to core active learning principles. The Myhill-Nerode theorem, which plays a central role in automata learning, has been established for HDAs [13], suggesting that similar techniques can be adapted to this setting. Additionally, HDAs have gained increasing attention in concurrency theory, with ongoing research exploring both their structural properties and practical applications. HDAs provide a flexible and expressive framework, enabling precise reasoning about concurrency while allowing for meaningful comparisons between different models and their extensions. Their structure makes them well-suited for developing learning algorithms, as well as for studying the theoretical properties that influence learnability.

Beyond algorithmic developments, this project will also explore fundamental properties of HDAs, their variants, and other concurrency models to better understand their learnability. Investigating structural characteristics — such as regularity notions, factorization properties, or minimization techniques — will help clarify the feasibility and complexity of inference techniques in these settings. Establishing a solid theoretical foundation is crucial for ensuring that the learning algorithms we develop are both efficient and broadly applicable.

2.2 Practical applications

We will focus on learning algorithms applicable to a wide variety of concurrency models, ensuring that the developed techniques can generalize across different system classes. This work also contributes to model inference for verification, making learned models usable for model-checking purposes. By adapting inference methods to concurrency settings, we aim to provide tools that facilitate the automated analysis of concurrent systems.

We also plan to investigate applications in network protocol verification and distributed systems. These systems are often complex, with specifications — such as RFCs — that may contain ambiguities, whether intentional or not. In this context, HDAs provide a compact and natural representation of concurrency, where an n-dimensional cell encodes $n!$ different sequential executions. This representation opens up new possibilities for inference techniques that scale to realistic system models.

To ensure practical impact, the learning algorithms developed in this project will be implemented within a dedicated library, designed to interface with existing frameworks such as LearnLib. This integration will allow us to leverage established learning techniques while extending their applicability to concurrency models, making our contributions both theoretically grounded and directly usable in practice.

3 Supervision

This thesis will be conducted at *Laboratoire de Recherche de l'EPITA* (LRE) under the supervision of Amazigh Amrane and Adrien Pommellet.

References

[1] Amazigh Amrane, Hugo Bazille, Emily Clement, and Uli Fahrenberg. Languages of higher-dimensional timed automata. In Lars Michael Kristensen and Jan Martijn E. M. van der Werf, editors, *Application and Theory of Petri Nets and Concurrency - 45th International Conference, PETRI NETS 2024, Geneva, Switzerland, June 26-28, 2024, Proceedings*, volume 14628 of *Lecture Notes in Computer Science*, pages 197–219. Springer, 2024.

[2] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Marie Fortin. Logic and languages of higher-dimensional automata. In *International Conference on Developments in Language Theory*, pages 51–67. Springer, 2024.

[3] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, Loïc Hélouët, and Philipp Schlehuber-Caissier. Petri nets and higher-dimensional automata, 2025.

[4] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.

[5] Marek A. Bednarczyk. *Categories of Asynchronous Systems*. PhD thesis, University of Sussex, UK, 1987.

[6] Alan W. Biermann and Jerome A. Feldman. On the synthesis of finite-state machines from samples of their behavior. *IEEE Trans. Computers*, 21(6):592–597, 1972.

[7] Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, and Martin Leucker. Learning communicating automata from mscs. *IEEE Trans. Software Eng.*, 36(3):390–408, 2010.

[8] Emily Clement, Enzo Erlich, and Jérémy Ledent. Expressivity of linear temporal logic for pomset languages of higher dimensional automata. *CoRR*, abs/2410.12493, 2024.

[9] Uli Fahrenberg. Higher-dimensional timed automata. In Alessandro Abate, Antoine Girard, and Maurice Heemels, editors, *ADHS*, volume 51 of *IFAC-PapersOnLine*, pages 109–114. Elsevier, 2018.

[10] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021.

[11] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. A Kleene theorem for higher-dimensional automata. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *CONCUR*, volume 243 of *LIPIcs*, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[12] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285(B):104914, 2022.

[13] Uli Fahrenberg and Krzysztof Ziemiański. A Myhill-Nerode theorem for higher-dimensional automata. In Luís Gomes and Robert Lorenz, editors, *PETRI NETS*, volume 13929 of *Lecture Notes in Computer Science*, pages 167–188. Springer-Verlag, 2023.

[14] Peter C. Fishburn. Intransitive indifference with unequal indifference intervals. *Journal of Mathematical Psychology*, 7(1):144–149, 1970.

[15] E Mark Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

[16] E. Mark Gold. Complexity of automaton identification from given data. *Inf. Control.*, 37(3):302–320, 1978.

[17] Falk Howar and Bernhard Steffen. Active automata learning as black-box search and lazy partition refinement. In Nils Jansen, Mariëlle Stoelinga, and Petra van den Bos, editors, *A Journey from Process Algebra via Timed Automata to Model Learning - Essays Dedicated to Frits Vaandrager on the Occasion of His 60th Birthday*, volume 13560 of *Lecture Notes in Computer Science*, pages 321–338. Springer, 2022.

[18] Malte Isberner, Falk Howar, and Bernhard Steffen. The ttt algorithm: A redundancy-free approach to active automata learning. In Borzoo Bonakdarpour and Scott A. Smolka, editors, *Runtime Verification*, pages 307–322, Cham, 2014. Springer International Publishing.

[19] Xiaodong Jia and Gang Tan. V-star: Learning visibly pushdown grammars from program inputs. *Proc. ACM Program. Lang.*, 8(PLDI):2003–2026, 2024.

[20] Anca Muscholl and Igor Walukiewicz. Active learning for sound negotiations. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 21:1–21:12. ACM, 2022.

[21] José Oncina, Pedro Garcia, et al. Inferring regular languages in polynomial update time. *Pattern recognition and image analysis*, 1(49-61):10–1142, 1992.

[22] Luc Passemard, Amazigh Amrane, and Uli Fahrenberg. Higher-dimensional automata: Extension to infinite tracks. *CoRR*, abs/2503.07881, 2025.

[23] Adrien Pommellet, Amazigh Amrane, and Edgar Delaporte. Active learning techniques for pomset recognizers. *CoRR*, abs/2501.03914, 2025.

[24] Frits Vaandrager, Bharat Garhewal, Jurriaan Rot, and Thorsten Wißmann. A new approach for active automata learning based on apartness. In Dana Fisman and Grigore Rosu, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 223–243, Cham, 2022. Springer International Publishing.

[25] Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006.

[26] Gerco van Heerdt, Tobias Kappé, Jurriaan Rot, and Alexandra Silva. Learning pomset automata. In Stefan Kiefer and Christine Tasson, editors, *Foundations of Software Science and Computation Structures - 24th International Conference, FOSSACS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings*, volume 12650 of *Lecture Notes in Computer Science*, pages 510–530. Springer, 2021.

[27] Gerco van Heerdt, Clemens Kupke, Jurriaan Rot, and Alexandra Silva. Learning weighted automata over principal ideal domains. *Lecture Notes in Computer Science*, 12077:602–621, 2020.

[28] Masaki Waga. Active learning of deterministic timed automata with myhill-nerode style characterization. In Constantin Enea and Akash Lal, editors, *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part I*, volume 13964 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2023.

[29] Glynn Winskel and Mogens Nielsen. Models for concurrency. In Samson Abramsky, Dov M. Gabbay, and Thomas S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4. Clarendon Press, Oxford, 1995.