Thesis Proposal:
Higher-Dimensional Automata on Infinite Pomsets

1 Introduction

Automata theory is fundamental for modeling and analyzing computational systems, playing a crucial role in verifying system correctness, inferring models for unknown systems, synthesizing components from specifications, and developing decision procedures. Finite automata over words, also known as Kleene automata, model terminating sequential systems with finite memory, where accepted words represent execution sequences. Their theory, supported by the Kleene, Büchi, and Myhill-Nerode theorems, connects regular expressions, monadic second-order logic, and semigroups.

For concurrent systems, executions are often represented as *pomsets* (partially ordered multisets) [35] instead of words. In a pomset, concurrent events are represented as labeled elements with no specific order relative to each other. Various classes of pomsets and their corresponding automata models exist, reflecting different interpretations of concurrency. Examples include branching automata and series-parallel pomsets [27–30], step transition systems and local trace languages [19], communicating finite-state machines and message sequence charts [23], asynchronous automata and Mazurkiewicz traces [41], and higher-dimensional automata (HDAs) with interval pomsets [14].

HDAs [36, 38] are general models of concurrency that extend traditional models like event structures and safe Petri nets [3, 39], asynchronous transition systems [5, 40], and Kleene automata. HDAs have gained significant attention in concurrency theory, offering an automata-like formalism that precisely captures non-interleaving concurrency. Initially explored through geometric and categorical approaches, the study of HDAs has shifted toward language theory, particularly since [14]. Key theoretical results include a Kleene theorem [15], a Myhill-Nerode theorem [18], and a Büchi theorem [2]. Higher-dimensional timed automata were introduced in [13], with their associated languages studied in [1]. These results demonstrate the robustness of the theory and establish a strong foundation for future developments, as seen in the (i)Po(m)set Project[1].

HDAs consist of a collection of cells representing concurrently running events, connected by face maps that model the start and termination of events. The language of an HDA is defined as a set of *interval pomsets* [21] with interfaces (interval ipomsets or *iipomsets*) [16]. Each event in an HDA execution P corresponds to a time interval of process activity, and the execution is constructed by joining elementary steps that represent segments of P. This gluing composition allows events to span across segments, linking one part to the next. Since any order extension of P remains a valid execution, HDA languages are inherently closed under *subsumption*, meaning that every possible interleaving of an execution is accepted. This property supports partial-order reduction and can improve state-space exploration when modeling systems 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 [39], asynchronous transition systems [5, 40], 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,

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

transfer arcs or generalized self-modifying net. These translations have been implemented in the prototype tool pn2hda[2]. For example, Fig. 1 illustrates Petri net and HDA models for a system with two events, labeled a and b, with the left side showing their interleaving execution ($a.b$ or $b.a$) and the right side showing their concurrent execution ($a \parallel b$), with a continuous path through the surface of a square.

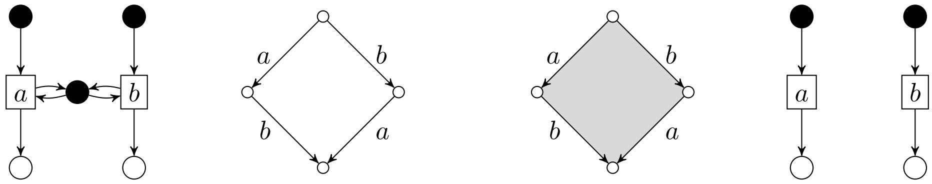

Figure 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b + b.a$; right: models for $a \parallel b$.

2 Research Objectives

To study non-terminating sequential machines and decision-related problems, Müller [33] and Büchi [6] defined automata that recognize words indexed by all natural numbers, known as ω-words. McNaughton [31] later proved the equivalence of these definitions and extended Kleene's theorem to ω-words using a non-nested ω-iteration. These automata have logical [6] and algebraic [7] characterizations, and, beyond their theoretical significance, they play a key role in specifying and verifying reactive systems [11]. This led to the extension of automata models for concurrency to the infinite case, with key results extending to ω-branching automata that admit Kleene-like and Büchi-like theorems [26]. Similar developments also apply to traces [9, 12, 22], resulting in decision procedures as corollaries. However, these extensions have not found widespread success in model checking due to the inherent complexity of the verification process, which becomes significantly more challenging when dealing with concurrency.

The goal of this thesis is to extend the theory of ω-HDAs (Higher-Dimensional Automata) to better model non-terminating concurrent systems. A theory of ω-HDAs could provide a suitable framework for modeling non-terminating concurrent systems with dependence and independence relations. In particular it could mitigate the issue of state-space explosion by considering executions as ω-ipomsets rather than treating all interleavings separately. For example, an infinite execution where event a must precede b is modeled as the subsumption closure of the pomset with $a < b$ and all other events occurring in parallel, instead of considering all interleavings separately.

Previous work [34] has laid the groundwork for ω-HDA theory by defining these automata in terms of ω-interval pomsets with interfaces, and by extending fundamental concepts to the infinite case. They demonstrated that isomorphisms of ω-ipomsets are unique and admit canonical decompositions. The study of HDAs with Muller and Büchi acceptance conditions revealed key differences from classical theory:

1. Unlike the finite case, languages of ω-HDAs are not closed under order extension (subsumption).
2. Muller acceptance is more expressive than Büchi acceptance, even in the non-deterministic case.

These differences led to adaptations of the original rational operations and the introduction of a non-nested ω-iteration to define ω-rational languages. However, not all ω-rational languages are (Muller) ω-regular.

Thus, significant open questions remain, especially in identifying classes of languages where the different definitions coincide. Exploring different types of acceptance conditions inspired by standard ω-automata or using variants of HDAs [10, 17] may address some of these challenges.

We also plan to consider *parity* acceptance conditions and parity *games*, given the important role they play in standard automata theory. Translations from LTL to parity automata and the use

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

of parity games for synthesis are now standard applications of formal methods [25, 37], and we will explore these venues for ω-HDAs.

Recent developments have established formal connections between HDAs and logical frameworks [2, 8], laying the groundwork for verification techniques that take advantage of their non-interleaving structure. Extending these connections to ω-HDAs, and exploring new ones, would significantly enhance model checking for non-terminating concurrent systems. Another important direction is infinite parallelism, as utilized in concurrent Kleene algebra [24] or the π-calculus [32]. Preliminary results in this area are presented in [4], and we aim to explore translations from Petri nets, especially unbounded ones, as well as concepts related to well-structured systems [20].

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] Thomas Baronner, Henning Basold, and Márton Hablicsek. Irrationality of process replication for higher-dimensional automata. In Uli Fahrenberg, Wesley Fussner, and Roland Glück, editors, *RAMiCS*, volume 14787 of *Lecture Notes in Computer Science*, pages 46–64. Springer-Verlag, 2024.

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

[6] J Richard Büchi. Symposium on decision problems: On a decision method in restricted second order arithmetic. In *Studies in Logic and the Foundations of Mathematics*, volume 44, pages 1–11. Elsevier, 1966.

[7] Olivier Carton, Dominique Perrin, and Jean-Eric Pin. Automata and semigroups recognizing infinite words. *Logic and automata*, 2:133–168, 2008.

[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] Volker Diekert and Anca Muscholl. Deterministic asynchronous automata for infinite traces. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 617–628. Springer, 1993.

[10] Jérémy Dubut. Trees in partial higher dimensional automata. In Mikołaj Bojańczyk and Alex Simpson, editors, *FOSSACS*, volume 11425 of *Lecture Notes in Computer Science*, pages 224–241. Springer-Verlag, 2019.

[11] Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. From Spot 2.0 to Spot 2.10: What's new? In *34th International Conference on Computer Aided Verification (CAV'22)*, volume 13372 of *Lecture Notes in Computer Science*, pages 174–187. Springer-Verlag, 2022.

[12] Werner Ebinger and Anca Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154(1):67–84, 1996.

[13] 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.

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

[15] 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.

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

[17] Uli Fahrenberg and Axel Legay. Partial higher-dimensional automata. In Lawrence S. Moss and Pawel Sobocinski, editors, *CALCO*, volume 35 of *LIPIcs*, pages 101–115. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[18] 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.

[19] Jean Fanchon and Rémi Morin. Pomset languages of finite step transition systems. In Giuliana Franceschinis and Karsten Wolf, editors, *PETRI NETS*, volume 5606 of *Lecture Notes in Computer Science*, pages 83–102. Springer-Verlag, 2009.

[20] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

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

[22] Paul Gastin, Antoine Petit, and Wiesław Zielonka. An extension of Kleene's and Ochmański's theorems to infinite traces. *Theoretical Computer Science*, 125(2):167–204, 1994.

[23] Blaise Genest, Dietrich Kuske, and Anca Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204(6):920–956, 2006.

[24] Tony Hoare, Stephan van Staden, Bernhard Möller, Georg Struth, and Huibiao Zhu. Developments in concurrent Kleene algebra. *Journal of Logic and Algebraic Methods in Programming*, 85(4):617–636, 2016.

[25] Swen Jacobs, Guillermo A. Pérez, Remco Abraham, Véronique Bruyère, Michaël Cadilhac, Maximilien Colange, Charly Delfosse, Tom van Dijk, Alexandre Duret-Lutz, Peter Faymonville, Bernd Finkbeiner, Ayrat Khalimov, Felix Klein, Michael Luttenberger, Klara J. Meyer, Thibaud Michaud, Adrien Pommellet, Florian Renkin, Philipp Schlehuber-Caissier, Mouhammad Sakr, Salomon Sickert, Gaëtan Staquet, Clément Tamines, Leander Tentrup, and Adam Walker. The reactive synthesis competition (SYNTCOMP): 2018-2021. *Int. J. Softw. Tools Technol. Transf.*, 26(5):551–567, 2024.

[26] Dietrich Kuske. Infinite series-parallel posets: Logic and languages. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, volume 1853 of *Lecture Notes in Computer Science*, pages 648–662. Springer-Verlag, 2000.

[27] K. Lodaya and P. Weil. A Kleene iteration for parallelism. In *Foundations of Software Technology and Theoretical Computer Science*, pages 355–366, 1998.

[28] K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages. In *STACS'98*, volume 1373 of *Lecture Notes in Computer Science*, pages 555–565. Springer-Verlag, 1998.

[29] K. Lodaya and P. Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1–2):347–380, 2000.

[30] K. Lodaya and P. Weil. Rationality in algebras with a series operation. *Information and Computation*, 171:269–293, 2001.

[31] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and control*, 9(5):521–530, 1966.

[32] Robin Milner. *Communicating and Mobile Systems - The Pi-Calculus*. Cambridge University Press, 1999.

[33] David E Muller. Infinite sequences and finite machines. In *Proceedings of the Fourth Annual Symposium on Switching Circuit Theory and Logical Design (swct 1963)*, pages 3–16. IEEE Computer Society, 1963.

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

[35] Vaughan R. Pratt. Modeling concurrency with partial orders. *Journal of Parallel Programming*, 15(1):33–71, 1986.

[36] Vaughan R. Pratt. Modeling concurrency with geometry. In David S. Wise, editor, *Conference Record of the Eighteenth Annual ACM Symposium on Principles of Programming Languages, Orlando, Florida, USA, January 21-23, 1991*, pages 311–322. ACM Press, 1991.

[37] Florian Renkin, Philipp Schlehuber-Caissier, Alexandre Duret-Lutz, and Adrien Pommellet. Dissecting ltlsynt. *Formal Methods in System Design*, 61(2):248–289, 2022.

[38] Rob J. van Glabbeek. Bisimulations for higher dimensional automata. Email message, June 1991. `http://theory.stanford.edu/~rvg/hda`.

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

[40] 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.

[41] Wiesław Zielonka. Notes on finite asynchronous automata. *RAIRO – Informatique Théorique et Applications*, 21(2):99–135, 1987.