Internship Proposal:

Petri Nets, Higher-Dimensional Automata and their Logics

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) [17] 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 [13–16], step transition systems and local trace languages [10], communicating finite-state machines and message sequence charts [12], asynchronous automata and Mazurkiewicz traces [22], and higher-dimensional automata (HDAs) with interval pomsets [6].

Another approach for modeling concurrent systems are Petri Nets [], with their different versions, semantics and interpretations. Petri Nets have a long standing history and are widely used in academia and industry. Despite being initially introduced as a model for true concurrency, most of the analysis techniques and tools for Petri Nets rely on interleaving semantics, such as the model checking competition (mcc).

HDAs [18, 19] are general models of concurrency that extend traditional models like event structures and safe Petri nets [3, 20], asynchronous transition systems [4, 21], 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 [6]. Key theoretical results include a Kleene theorem [7], a Myhill-Nerode theorem [9], and a Büchi theorem [2]. Higher-dimensional timed automata were introduced in [5], 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* [11] with interfaces (interval ipomsets or *iipomsets*) [8]. 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.

In [?, ?] it was shown how to translate safe Petri Nets and many of their extension (inhibitor arcs, self-modifying nets) into (partial) HDAs. This allows to in a way to recover the full concurrent semantics of the Petri Net by viewing it via its associated HDA. This is particularly interesting as

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

Petri Nets are very well established in industry and academia and provide a very generic modeling framework due to their notion of discrete actions mixed with the persistent notion of resources via the tokens. These translations from Petri Nets to HDAs 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.

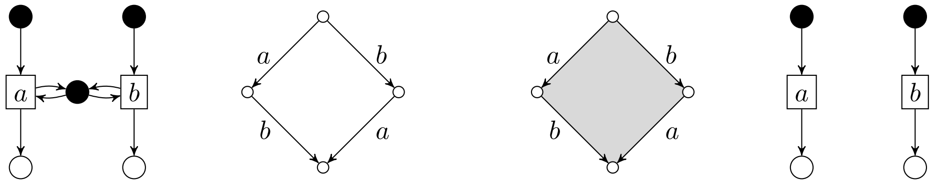

Figure 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b + b.a$; right: models for $a \parallel b$.

On the purely logical side of things, a LTL-like logic for pomsets has been introduced in [**?**]. It enriches standard LTL with additional operators allowing to distinguish and work with different concurrent events. The paper uses it in a purely theoretic fashion to prove a Kamp-like theorem.

2 Research Objectives

The objective of this internship is to build upon the existing results on HDAs, Petri Nets and their logics for efficient model checking and verification of concurrent systems, notably given in the from of Petri Nets. Indeed, the theory of HDAs has seen a rapid progression in recent years, however practical applications and *success stories* are still somewhat lacking. First steps in this direction have been taken by providing translations from Petri Nets to HDAs [] and seeking to improve the internal representation [**?**], however it has yet to be used for actual verification and model checking tasks.

We will therefore seek to derive a suitable version of an LTL-like logic specifically tailored for HDAs obtained from translating Petri Nets and which is hopefully directly verifiable over the max-cell representation of the HDA. The reason to tailor the logic specifically for HDAs obtained from Petri Nets is that such HDAs can be anonymous (i.e. events with the same label are indistinguishable) and are guaranteed to be full (i.e. not partial). Depending on the advances and the properties of the logic, we will either seek to apply it within the mcc or on interpreted Petri Nets used in manufacturing and automation.

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.

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

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

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

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

[14] 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.

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

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

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

[18] 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.

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

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

[21] 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.

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