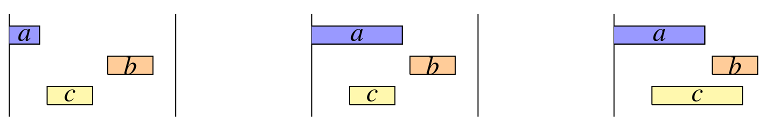
MSO for Higher-Dimensional Automata

Amazigh Amrane
Laboratoire de Recherche de L'EPITA (LRE)

In the study of concurrent systems, executions are often modeled using pomsets (partially ordered multisets) rather than sequences of events [22]. 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 [17, 18, 19, 20], step transition systems and local trace languages [12], communicating finite-state machines and message sequence charts [15], asynchronous automata and Mazurkiewicz traces [28], and higher-dimensional automata (HDAs) with interval pomsets [8].

HDAs [23, 25] are general models of concurrency that extend traditional models like event structures and safe Petri nets [4, 26], asynchronous transition systems [5, 27], 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 [8]. Key theoretical results include a Kleene theorem [9], a Myhill-Nerode theorem [11], and a Büchi-Elgot-Trakhtenbrot theorem [2, 3]. Higher-dimensional timed automata were introduced in [7], with their associated languages studied in [1]. These results demonstrate the robustness of the theory and establish a strong foundation.

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* [13] with interfaces (interval ipomsets) [10]. 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. HDA languages are inherently closed under *subsumption*, meaning that every possible order extension of an execution is accepted. This property supports partial-order reduction and can improve state-space exploration when modeling systems with HDAs. For example, activity intervals of events are depicted bellow. Here, we remove precedence order by prolonging intervals *i.e.*, by making the intervals overlap, from left-to-right, which creates a sequence of subsumptions.



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, Among such 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 [4], 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. 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.

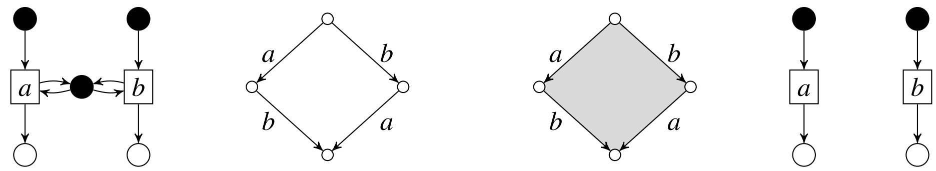
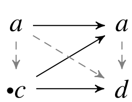


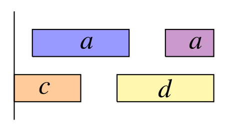
Figure 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b+b.a$; right: models for $a \parallel b$.

A *pomset with interfaces*, or *ipomset*, $(P, <_P, \dashrightarrow_P, S_P, T_P, \lambda_P)$ consists of a *partially ordered multiset* $(P, <_P, \dashrightarrow_P, \lambda_P)$, where λ_P is a labelling function, $<_P$ (resp $\dashrightarrow_P$) precedence (resp. event) order, together with subsets $S_P, T_P \subseteq P$ (*source* and *target interfaces*) such that elements of S_P are $<_P$-minimal and those of T_P are $<_P$-maximal. The *width* $\mathsf{wid}(P)$ of an ipomset P is the cardinality of its maximal $<$-antichain. An ipomset P is *interval* if $<_P$ is an interval order [14]; that is, if it admits an interval representation given by functions b and e from P to real numbers such that $b(x) \leq e(x)$ for all $x \in P$ and $x <_P y$ iff $e(x) < b(y)$ for all $x, y \in P$. In what follows, ipomset means interval ipomset.

In this work, we study ipomsets from a categorical and a logical point of views. We first show that ipomsets form a category isomorphic to a category of *step sequences*, which are equivalence classes of words on special discrete ipomsets called *starters* and *terminators* under a natural relation. For example, below, the ipomset $(\{x_1,\ldots,x_4\}, \{(x_1,x_2),(x_3,x_4),(x_3,x_2)\}, \{(x_1,x_3),(x_1,x_4),(x_2,x_4)\}, \{x_3\}, \emptyset, \{(x_1,a),(x_2,a),(x_3,c),(x_4,d)\})$ is depicted on the left, its corresponding step sequence on the middle composed of six steps: (1) a is started while c is running; (2) c is terminated while a is running, (3) d is started while a is running, (4) a is terminated while d is running, (5) another a is started while d is running and finally a and d are terminated, and an interval representation of this ipomset on the right.



$$\begin{bmatrix} a\bullet \\ \bullet c \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ d \bullet \end{bmatrix} \begin{bmatrix} \bullet a \\ \bullet d \bullet \end{bmatrix} \begin{bmatrix} a \bullet \\ \bullet d \bullet \end{bmatrix} \begin{bmatrix} \bullet a \\ \bullet d \end{bmatrix}$$



We extend the correspondence between step sequences and ipomsets to the logic side, by showing that a language of ipomsets with bounded width is MSO-definable if and only if the corresponding language of step sequences is MSO-definable. More precisely, we construct an MSO interpretation of ipomsets into step sequences (or rather into their representatives), and of canonical representatives of step sequences into ipomsets. This shows that, up to isomorphism, MSO has the same expressive power over step sequences and ipomsets when a width bound is fixed. As corollaries, a Büchi-Elgot-Trakhtenbrot theorem and an algorithm building an MSO sentence $\psi\!\downarrow$ satisfied by all the order extensions of the ipomsets satisfying an MSO sentence ψ. The latter is not decidable for general pomsets. Also, the satisfiability and model checking problems for HDAs are both decidable.

These corollaries has motivated further research into the expressive power of first-order logic over ipomsets. An initial step in this direction appears in [6]. Along similar lines, another operational model has begun to receive attention: ω-HDAs, that is, HDAs over infinite ipomsets [21] which is accepted and will be presented in FSCD2025. Still, in both areas, substantial work remains to be done. In particular, developing a logical characterization for the infinite case would also be of interest. One may also consider *parity* acceptance conditions and parity *games*, given the important role these play in standard automata theory. Translations from LTL to parity automata and the use of parity games for synthesis are now standard applications of formal methods [24, 16], and one may explore these venues for ω-HDAs.

This work is carried out in cooperation with Hugo Bazille (LRE & EPITA Rennes, France), Emily Clement (LIPN & Sorbonne University, France), Uli Fahrenberg (LRE & EPITA Rennes, France), Marie Fortin (IRIF & CNRS Paris, France) and Krzysztof Ziemiański (Warsaw U, Poland).

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, Emily Clement, Uli Fahrenberg, Marie Fortin, and Krzysztof Ziemiański. Büchi-elgot-trakhtenbrot theorem for higher-dimensional automata, 2025.

[3] 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.

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

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

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

[7] Uli Fahrenberg. Higher-dimensional timed automata. In *ADHS*, pages 109–114, 2018.

[8] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. 31(5):575–613, 2021.

[9] 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, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

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

[11] Uli Fahrenberg and Krzysztof Ziemiański. A Myhill-Nerode theorem for higher-dimensional automata. In Luís Gomes and Robert Lorenz, editors, *PETRI NETS*, volume 13929, pages 167–188, 2023.

[12] Jean Fanchon and Rémi Morin. Pomset languages of finite step transition systems. In *PETRI NETS*, pages 83–102, 2009.

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

[14] Peter C. Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. Wiley, 1985.

[15] 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.

[16] 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.

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

[18] K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages. In *STACS'98*, volume 1373, pages 555–565, 1998.

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

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

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

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

[23] 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.

[24] Florian Renkin, Philipp Schlehuber-Caissier, Alexandre Duret-Lutz, and Adrien Pommellet. Dissecting ltl-synt. 61(2):248–289, 2022.

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

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

[27] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 4.

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