LANGUAGES OF HIGHER-DIMENSIONAL AUTOMATA EXTENDED ABSTRACT

ULI FAHRENBERG

Higher-dimensional automata (HDAs) were introduced by Pratt [11] and van Glabbeek [12] as a geometrical model for non-interleaving concurrency. HDAs are general models of concurrency that subsume, for example, event structures and safe Petri nets [13]. Asynchronous transition systems [1,16] are two-dimensional HDAs; HDAs without concurrency, the 1-dimensional case, are standard automata.

Unlike for standard automata and many other models, there has until recently been no theory of *languages* of HDAs. Such notions are important for connecting models to behaviours, but HDAs have been developed first of all with a view on operational, topological and geometric aspects.

In two recent papers [4, 6] we have started to develop the language theory of HDAs. Like languages related to other formalisms for concurrency, languages of HDAs need to account for both the sequential and the concurrent nature of computations. Their elements are thus finite *pomsets* or *partial words*. As an example, Figure 1 displays an HDA consisting of two squares, with three events labeled a, c, and d. Here the a-labeled event is executed concurrently to the sequence $c.d$, so that the language of this HDA will contain the pomset

$$\left(\begin{smallmatrix} & a & \\ c & \longrightarrow & d \end{smallmatrix} \right).$$

Partial words and pomsets have a long history as semantics for concurrent systems [14, 15]. In this setting, it is usual to consider languages which are closed under *subsumption* [7], so that adding order to a behaviour does not make it invalid. The subclass of pomsets which are *interval orders* [8] has seen abundant attention in concurrency theory and distributed systems [9, 10, 14]. A pomset is an interval order precisely if it is 2+2-*free*, that is, does not contain an induced subpomset of the form

$$\mathsf{2{+}2} = \left(\begin{smallmatrix} \bullet & \longrightarrow & \bullet \\ \bullet & \longrightarrow & \bullet \end{smallmatrix} \right).$$

We model HDAs as presheaves on a novel notion of labelled precube category, so that events and labels are built into the base category. An execution of an HDA is a (higher-dimensional) path [13]: a sequence of cells connected with operations of starting and terminating events. With every path one can associate the set of events that occur during this execution and order these according to precedence. Labels of paths are then pomsets with *interfaces* (*ipomsets*) [3, 5], where the interfaces gather the events that are active in the initial and final cells.

The language of an HDA collects the ipomsets of all its accepting paths. We show in [4] that languages of HDAs are subsumption-closed sets of interval-order ipomsets, and that any interval ipomset may be generated by an HDA. We also show that languages of HDAs are closed under

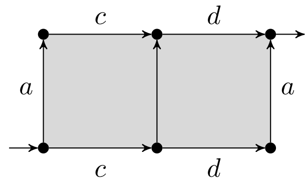

FIGURE 1. HDA which executes a in parallel with $c.d$. Initial and accepting cells marked with incoming and outgoing arrows.

binary union and parallel composition, and further that *bisimilarity* of HDAs [2, 13] implies language equivalence.

Subsumption-closed sets of interval ipomsets can be combined using union $\cup$, gluing (serial) composition $*$, parallel composition $\|$, and the (serial) Kleene plus $^+$. The languages generated from singletons using these operations yield a natural notion of rational language. In [6] we show a *Kleene*-type theorem that the rational languages are precisely the regular languages, *i.e.*, languages of finite HDAs:

Theorem [6]**.** A subsumption-closed set of interval ipomsets is rational iff it is regular.

Proving this theorem requires a gluing operation on HDAs which is rather intricate, because we must glue HDAs along higher-dimensional cells. We introduce two tools which may be of independent interest. Higher-dimensional automata with interfaces (iHDAs) are analogues of HDAs where events can be assigned to source or target interfaces. Any HDA can be converted to an equivalent iHDA (recognising the same language) and vice versa, using adjoint functors of resolution and closure.

Our second set of tools is motivated by algebraic topology. First, we introduce cylinder objects, allowing us to convert any cospan $f : Y \to X \leftarrow Z : g$ into an iHDA $C(f, g)$ together with an *initial inclusion* $\tilde{f} : Y \to C(f, g)$, a *final inclusion* $\tilde{g} : Z \to C(f, g)$, and a projection $p : C(f, g) \to X$. Using this we may decompose any map between iHDAs into an (initial or final) inclusion followed by a (future or past) path-lifting map, which in turn allows us to "pull apart" start cells and accept cells of iHDAs and to treat serial compositions and loops. It also hints at some notion of factorization systems or model categories for HDAs, but this we leave for future work.

Joint work with Christian Johansen (NTNU, Norway), Georg Struth (U of Sheffield, UK and Collegium de Lyon, France), and Krzysztof Ziemiański (Warsaw U, Poland) based on [4, 5, 6].

References

[1] M. A. Bednarczyk. *Categories of Asynchronous Systems.* PhD thesis, Univ. of Sussex, UK, 1987.
[2] U. Fahrenberg. A category of higher-dimensional automata. In *FoSSaCS*, vol. 3441 of *Lect. Notes Comput. Sci.* Springer, 2005.
[3] U. Fahrenberg, C. Johansen, G. Struth, and R. B. Thapa. Generating posets beyond N. In *RAMiCS*, vol. 12062 of *Lect. Notes Comput. Sci.* Springer, 2020. https://arxiv.org/abs/1910.06162.
[4] U. Fahrenberg, C. Johansen, G. Struth, and K. Ziemiański. Languages of higher-dimensional automata. *Math. Struct. Comput. Sci.*, 2021. https://arxiv.org/abs/2103.07557.
[5] U. Fahrenberg, C. Johansen, G. Struth, and K. Ziemiański. Posets with interfaces for concurrent Kleene algebra. *CoRR*, abs/2106.10895, 2021. https://arxiv.org/abs/2106.10895.
[6] U. Fahrenberg, C. Johansen, G. Struth, and K. Ziemiański. A Kleene theorem for higher-dimensional automata. *CoRR*, abs/2202.03791, 2022. https://arxiv.org/abs/2202.03791.
[7] J. Fanchon and R. Morin. Regular sets of pomsets with autoconcurrency. In *CONCUR*, vol. 2421 of *Lect. Notes Comput. Sci.* Springer, 2002.
[8] P. C. Fishburn. Intransitive indifference with unequal indifference intervals. *J. Math. Psych.*, 7(1):144–149, 1970.
[9] R. Janicki and M. Koutny. Structure of concurrency. *Theor. Comput. Sci.*, 112(1):5–52, 1993.
[10] L. Lamport. The mutual exclusion problem. *J. ACM*, 33(2):313–326, 1986.
[11] V. R. Pratt. Modeling concurrency with geometry. In *POPL*, New York City, 1991. ACM Press.
[12] R. J. van Glabbeek. Bisimulations for higher dimensional automata. Email message, June 1991. http://theory.stanford.edu/~rvg/hda.
[13] R. J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theor. Comput. Sci.*, 356(3):265–290, 2006.
[14] W. Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, vol. 625 of *Lect. Notes Comput. Sci.* Springer, 1992.
[15] J. Winkowski. An algebraic characterization of the behaviour of non-sequential systems. *Inf. Process. Lett.*, 6(4):105–109, 1977.
[16] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 4. Clarendon Press, Oxford, 1995.

EPITA Research and Development Laboratory (LRDE), Paris, France