Closure and Decision Properties
for Higher-Dimensional Automata

Amazigh Amrane[a], Hugo Bazille[a], Uli Fahrenberg[a], Krzysztof Ziemiański[b]

[a] *EPITA Research Laboratory (LRE), France*
[b] *University of Warsaw, Poland*

Abstract

We report some further developments regarding the language theory of higher-dimensional automata (HDAs). Regular languages of HDAs are sets of finite interval partially ordered multisets (pomsets) with interfaces. We show a pumping lemma which allows us to expose a class of non-regular languages. Concerning decision and closure properties, we show that inclusion of regular languages is decidable (hence is emptiness), and that intersections of regular languages are again regular. Yet complements of regular languages are not always regular. We introduce a width-bounded complement and show that width-bounded complements of regular languages are again regular.

We also study determinism and ambiguity. We show that it is decidable whether a regular language is accepted by a deterministic HDA and that there exists regular languages with unbounded ambiguity. Finally, we characterize one-letter deterministic languages in terms of ultimately periodic functions.

1. Introduction

Higher-dimensional automata (HDAs), introduced by Pratt and van Glabbeek [21, 23], are a general geometric model for non-interleaving concurrency which subsumes, for example, event structures and Petri nets [24]. HDAs of dimension one are standard automata, whereas HDAs of dimension two are isomorphic to asynchronous transition systems [5, 16, 22]. As an example, Figure 1 shows Petri net and HDA models for a system with two events, labeled a and b. The Petri

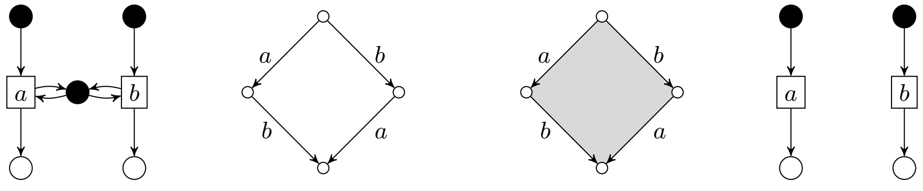


Figure 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b + b.a$; right: models for $a \parallel b$.

Preprint submitted to Theoretical Computer Science *February 23, 2025*

net and HDA on the left side model the (mutually exclusive) interleaving of a and b as either $a.b$ or $b.a$; those to the right model concurrent execution of a and b. In the HDA, this independence is indicated by a filled-in square.

Recent work defines languages of HDAs [9], which are sets of partially ordered multisets with interfaces (ipomsets) [11] that are closed under subsumptions. Follow-up papers introduce a language theory for HDAs, showing a Kleene theorem [10], which makes a connection between rational and regular ipomset languages (those accepted by finite HDAs), and a Myhill-Nerode theorem [13] stating that regular languages are precisely those that have finite prefix quotient. Here we continue to develop this nascent higher-dimensional automata theory.

Our first contribution, in Section 4, is a pumping lemma for HDAs, based on the fact that if an ipomset accepted by an HDA is long enough, then there is a cycle in the path that accepts it. As an application we can expose a class of non-regular ipomset languages. We also show that regular languages are closed under intersection, both using the Myhill-Nerode theorem and an explicit product construction.

In Section 5 we introduce a translation from HDAs to ordinary finite automata over an alphabet of discrete ipomsets, called ST-automata. The translation forgets some of the structure of the HDA; nevertheless, it allows us to show that inclusion of regular ipomset languages is decidable. This immediately implies that emptiness is decidable; universality is trivial given that the universal language is not regular.

The paper [13] introduces deterministic HDAs and shows that not all HDAs are determinizable. We show that the question whether a language is deterministic is decidable. As a weaker notion in-between determinism and nondeterminism, one may ask whether all regular languages may be recognized by finitely ambiguous HDAs, *i.e.*, those in which there is an upper bound for the number of accepting paths on any ipomset. We show in Section 6 that the answer to this question is negative and that there are regular languages of unbounded ambiguity.

In Section 7, we discuss one-letter languages. For regular languages, [6] develops a characterization based on ultimately periodic functions. We extend this to regular languages on pomsets with one letter. However, we show that also one-letter languages may be non-deterministic and of unbounded ambiguity.

Finally, in Section 8, we are interested in a notion of complement. This immediately raises two problems: first, complements of ipomset languages are generally not closed under subsumption; second, the complement of the empty language, which is regular, is the universal language, which is non-regular. The first problem is solved by taking subsumption closure, turning complement into a pseudocomplement in the sense of lattice theory.

As to the second problem, we can show that complements of regular languages are non-regular. Yet if we restrict the width of our languages, *i.e.*, the number of events which may occur concurrently, then the so-defined width-bounded complement has good properties: it is still a pseudocomplement; the languages for which double complement is identity have an easy characterization; and finally width-bounded complements of regular languages are again regular. The proof of that last property again uses ST-automata and the fact that the induced translation from ipomset languages to word languages over discrete ipomsets has good algebraic properties. We note that width-bounded languages and (pseudo)complements are found in other works on concurrent languages, for example [14, 19, 20].

Another goal of this work was to obtain the above results using automata-theoretic means as opposed to category-theoretic or topological ones. Indeed we do not use presheaves, track objects, cylinders, or any other of the categorical or topological constructions employed in [10, 13]. Categorical reasoning would have simplified proofs in several places, but no background in category theory or algebraic topology is necessary to understand this paper.

To sum up, our main contributions to higher-dimensional automata theory are as follows:

- a pumping lemma (Lemma 15);
- closure of regular languages under intersection (Proposition 19);
- decidability of inclusion of regular languages (Theorem 28);
- decidability of determinism of a regular language (Theorem 41);
- regular languages of unbounded ambiguity (Proposition 43);
- a characterization of one-letter deterministic languages (Proposition 52);
- closure of regular languages under width-bounded complement (Theorem 63).

This paper is an extended version of [4] which was presented at the 20th International Colloquium on Theoretical Aspects of Computing (ICTAC 2023). Compared to this previous work, we update the definition of ST-automata to the one introduced in [2], we describe one-letter languages, and we provide an effective procedure to decide whether a regular language is deterministic. The latter solves an open problem posed in [4]. Also in [4], we sought further development of the theory of timed HDAs and a Büchi-like characterization of HDA languages. This has been achieved in [1] and [3], respectively.

2. Pomsets with interfaces

HDAs model systems in which (labeled) events have duration and may happen concurrently. Notably, as seen in the introduction, concurrency of events is a more general notion than interleaving. Every event has an interval in time during which it is active: it starts at some point in time, then remains active until it terminates, and never appears again. Events may be concurrent, in which case their activity intervals overlap: one of the two events starts before the other terminates. Executions are thus isomorphism classes of partially ordered intervals. For reasons of compositionality we also consider executions in

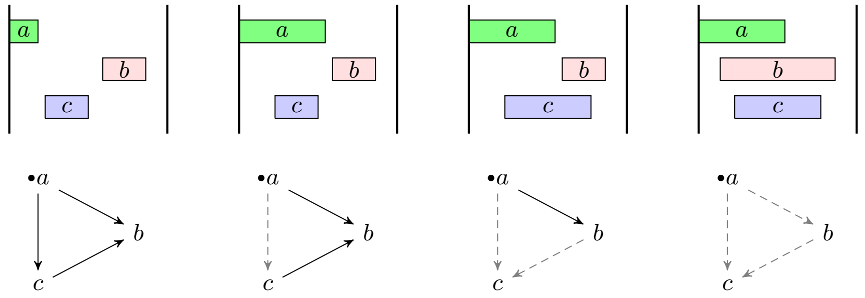


Figure 2: Activity intervals of events (top) and corresponding ipomsets (bottom), *cf.* Example 1. Full arrows indicate precedence order; dashed arrows indicate event order; bullets indicate interfaces.

which events may be active already at the beginning or remain active at the end; we will model these using *interfaces*.

Any time point of an execution defines a *concurrency list* (or *conclist*) of currently active events. The relative position of any two concurrent events on such lists does not change during passage of time; this equips events of an execution with a partial order which we call *event order*. The temporal order of non-concurrent events (one of two events terminating before the other starts) introduces another partial order which we call *precedence*. An execution is, then, a collection of labeled events together with two partial orders.

To make the above precise, let Σ be a finite alphabet. We define three notions, in increasing order of generality:

- A *concurrency list*, or *conclist*, $U = (U, \dashrightarrow_U, \lambda_U)$ consists of a finite set U, a strict total order $\dashrightarrow_U \subseteq U \times U$ (the event order),[1] and a labeling $\lambda_U : U \to \Sigma$.

- A *partially ordered multiset*, or *pomset*, $P = (P, <_P, \dashrightarrow_P, \lambda_P)$ consists of a finite set P, two strict partial orders $<_P, \dashrightarrow_P \subseteq P \times P$ (precedence and event order), and a labeling $\lambda_P : P \to \Sigma$, such that for each $x \neq y$ in P, at least one of $x <_P y$, $y <_P x$, $x \dashrightarrow_P y$, or $y \dashrightarrow_P x$ holds.

- A *pomset with interfaces*, or *ipomset*, $(P, <_P, \dashrightarrow_P, S_P, T_P, \lambda_P)$ consists of a pomset $(P, <_P, \dashrightarrow_P, \lambda_P)$ 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.

We will omit the subscripts $_U$ and $_P$ whenever possible. We will also often use the notation $_SP_T$ instead of $(P, <, \dashrightarrow, S, T, \lambda)$ if no confusion may arise.

Conclists may be regarded as pomsets with empty precedence (*discrete* pomsets). Pomsets are ipomsets with empty interfaces, and in any ipomset P, the

[1] A strict *partial* order is a relation which is irreflexive and transitive; a strict *total* order is a relation which is irreflexive, transitive, and total. We may omit the "strict".

subpomsets induced by S_P and T_P are conclists. Note that different events of ipomsets may carry the same label; in particular we do *not* exclude autoconcurrency. Figure 2 shows some simple examples. Source and target events are marked by "•" at the left or right side, and if the event order is not shown, we assume that it goes downwards, as in Figure 2.

An ipomset P is *interval* if $<_P$ is an interval order [15], that is, if it admits an interval representation given by functions b and e from P to real numbers such that $b(x) \le e(x)$ for all $x \in P$ and $x <_P y$ iff $e(x) < b(y)$ for all $x, y \in P$. Given that our ipomsets represent activity intervals of events, any of the ipomsets we will encounter will be interval, and we omit the qualification "interval". We emphasise that this is *not* a restriction, but rather induced by the semantics, see also [26]. We let iiPoms denote the set of (interval) ipomsets.

Ipomsets may be *refined* by shortening activity intervals, potentially removing concurrency and expanding precedence. The inverse to refinement is called *subsumption* and defined as follows. For ipomsets P and Q we say that Q subsumes P and write $P \sqsubseteq Q$ if there is a bijection $f : P \to Q$ for which

(1) $f(S_P) = S_Q$, $f(T_P) = T_Q$, and $\lambda_Q \circ f = \lambda_P$;

(2) $f(x) <_Q f(y)$ implies $x <_P y$;

(3) $x \not<_P y$, $y \not<_P x$ and $x \dashrightarrow_P y$ imply $f(x) \dashrightarrow_Q f(y)$.

That is, f respects interfaces and labels, reflects precedence, and preserves essential event order. (Event order is essential for concurrent events, but by transitivity, it also appears between non-concurrent events. Subsumptions ignore such non-essential event order.) This definition adapts the one of [17] to event orders and interfaces. Intuitively, P has more order and less concurrency than Q.

Example 1. In Figure 2 there is a sequence of subsumptions from left to right:

$$\bullet a \to c \to b \sqsubseteq \begin{bmatrix} \bullet a & \searrow & \\ & & b \\ c & \nearrow & \end{bmatrix} \sqsubseteq \left[\begin{smallmatrix} \bullet a \to b \\ c \end{smallmatrix}\right] \sqsubseteq \begin{bmatrix} \bullet a \\ b \\ c \end{bmatrix}$$

An event e_1 is smaller than e_2 in the precedence order if e_1 is terminated before e_2 is started in the interval representation; e_1 is smaller than e_2 in the event order if they are concurrent and e_1 is above e_2 in the respective conclist.

Isomorphisms of ipomsets are invertible subsumptions, *i.e.*, bijections f for which items (2) and (3) above are strengthened to

(2′) $f(x) <_Q f(y)$ iff $x <_P y$;

(3′) $x \not<_P y$ and $y \not<_P x$ imply that $x \dashrightarrow_P y$ iff $f(x) \dashrightarrow_Q f(y)$.

Due to the requirement that all elements are ordered by $<$ or $\dashrightarrow$, there is at most one isomorphism between any two ipomsets. Hence we may switch freely between ipomsets and their isomorphism classes. We will also call these equivalence classes ipomsets and often conflate equality and isomorphism.

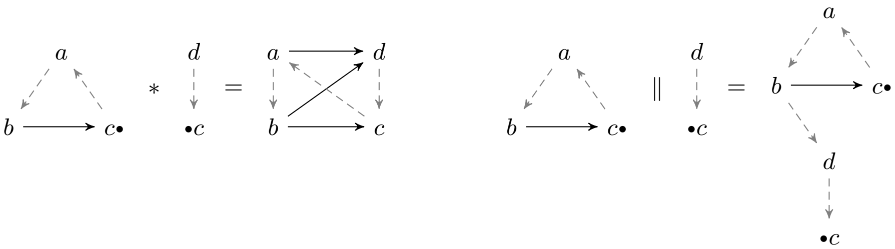


Figure 3: Gluing and parallel composition of ipomsets.

2.1. Compositions

The standard serial and parallel compositions of pomsets [17] extend to ipomsets. The *parallel* composition of ipomsets P and Q is

$$P \parallel Q = (P \sqcup Q, <, \dashrightarrow, S, T, \lambda),$$

where $P \sqcup Q$ denotes disjoint union and

- $x < y$ if $x <_P y$ or $x <_Q y$;
- $x \dashrightarrow y$ if $x \dashrightarrow_P y$, $x \dashrightarrow_Q y$, or $x \in P$ and $y \in Q$;
- $S = S_P \cup S_Q$ and $T = T_P \cup T_Q$;
- $\lambda(x) = \lambda_P(x)$ if $x \in P$ and $\lambda(x) = \lambda_Q(x)$ if $x \in Q$.

Note that parallel composition of ipomsets is generally not commutative, see [11] or Example 58 below for details.

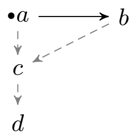
Serial composition generalises to a *gluing* composition which continues interface events across compositions and is defined as follows. Let P and Q be ipomsets such that

- $T_P = S_Q$ and $(P - T_P) \cap (Q - S_Q) = \emptyset$,[2]
- $x \dashrightarrow_P y$ iff $x \dashrightarrow_Q y$ for all $x, y \in T_P = S_Q$, and
- the restrictions $\lambda_P \restriction T_P = \lambda_Q \restriction S_Q$,

then $P * Q = (P \cup Q, <, \dashrightarrow, S_P, T_Q, \lambda)$, where

- $x < y$ if $x <_P y$, $x <_Q y$, or $x \in P - T_P$ and $y \in Q - S_Q$;
- $\dashrightarrow$ is the transitive closure of $\dashrightarrow_P \cup \dashrightarrow_Q$;
- $\lambda(x) = \lambda_P(x)$ if $x \in P$ and $\lambda(x) = \lambda_Q(x)$ if $x \in Q$.

[2] We use "$-$" for set difference instead of the perhaps more common "\".



$$\text{Sparse: } \begin{bmatrix} \bullet a \bullet \\ c \bullet \\ d \bullet \end{bmatrix} * \begin{bmatrix} \bullet a \\ \bullet c \bullet \\ \bullet d \bullet \end{bmatrix} * \begin{bmatrix} b \bullet \\ \bullet c \bullet \\ \bullet d \bullet \end{bmatrix} * \begin{bmatrix} \bullet b \\ \bullet c \\ \bullet d \end{bmatrix}$$

$$\text{Dense: } \left[\begin{smallmatrix} \bullet a \bullet \\ c \bullet \end{smallmatrix}\right] * \begin{bmatrix} \bullet a \bullet \\ \bullet c \bullet \\ d \bullet \end{bmatrix} * \begin{bmatrix} \bullet a \\ \bullet c \bullet \\ \bullet d \bullet \end{bmatrix} * \begin{bmatrix} b \bullet \\ \bullet c \bullet \\ \bullet d \bullet \end{bmatrix} * \begin{bmatrix} \bullet b \\ \bullet c \bullet \\ \bullet d \bullet \end{bmatrix} * \left[\begin{smallmatrix} \bullet c \bullet \\ \bullet d \end{smallmatrix}\right] * \bullet c$$

Figure 4: Ipomset of size 3.5 and two of its step decompositions.

Gluing is, thus, only defined if the targets of P are equal to the sources of Q *as conclists*. If we would not conflate equality and isomorphism, we would have to define the carrier set of $P * Q$ to be the disjoint union of P and Q quotiented out by the unique isomorphism $T_P \to S_Q$. We will often omit the "$*$" in gluing compositions. Figure 3 shows some examples.

An ipomset P is a *word* (with interfaces) if $<_P$ is total. Conversely, P is *discrete* if $<_P$ is empty (hence $\dashrightarrow_P$ is total). Conclists are discrete ipomsets without interfaces. The relation $\sqsubseteq$ is a partial order on iiPoms with minimal elements words and maximal elements discrete ipomsets. Further, gluing and parallel compositions respect $\sqsubseteq$.

2.2. Special ipomsets

A *starter* is a discrete ipomset U with $T_U = U$, a *terminator* one with $S_U = U$. The intuition is that a starter does nothing but start the events in $A = U - S_U$, and a terminator terminates the events in $B = U - T_U$. These will be so important later that we introduce special notation, writing ${}_A{\uparrow}U = {}_{U \setminus A}U_U$ and $U{\downarrow}_B = {}_UU_{U \setminus B}$ for the above. Starter ${}_A{\uparrow}U$ is *elementary* if A is a singleton, similarly for $U{\downarrow}_B$. Discrete ipomsets U with $S_U = T_U = U$ are identities for the gluing composition and written id_U. Note that $\mathsf{id}_U = {}_\emptyset{\uparrow}U = U{\downarrow}_\emptyset = {}_UU_U$.

The *width* $\mathsf{wid}(P)$ of an ipomset P is the cardinality of a maximal $<$-antichain. For $k \geq 0$, we let $\mathsf{iiPoms}_{\leq k} \subseteq \mathsf{iiPoms}$ denote the set of ipomsets of width at most k. The *size* of an ipomset P is $\mathsf{size}(P) = |P| - \frac{1}{2}(|S_P| + |T_P|)$. Identities are exactly the ipomsets of size 0. Elementary starters and terminators are exactly the ipomsets of size $\frac{1}{2}$.

Any ipomset can be decomposed as a gluing of starters and terminators [11], see also [18]. Such a presentation we call a *step decomposition.* If starters and terminators are alternating, the step decomposition is called *sparse*; if they are all elementary, then it is *dense.* The algebra of these representations is studied more in depth in [2], along with how to represent subsumptions in this formalism.

Example 2. Figure 4 illustrates two step decompositions. The sparse one first starts c and d, then terminates a, starts b, and terminates b, c and d together. The dense one first starts c, then starts d, terminates a, starts b, and finally terminates b, d, and c in order.

Lemma 3 ([13]). *Every ipomset P has a unique sparse step decomposition.*

Dense step decompositions are generally not unique, but they all have the same length.

Lemma 4. *Every dense step decomposition of ipomset P has length $2\,\mathsf{size}(P)$.*

Proof. Every element of a dense step decomposition of P starts precisely one event or terminates precisely one event. Thus every event in $P - (S_P \cup T_P)$ gives rise to two elements in the step decomposition and every event in $S_P \cup T_P - (S_P \cap T_P)$ to one element. The length of the step decomposition is, thus, $2|P| - 2|S_P \cup T_P| + |S_P \cup T_P| - |S_P \cap T_P| = 2|P| - (|S_P| + |T_P| - |S_P \cap T_P|) - |S_P \cap T_P| = 2\,\mathsf{size}(P)$.

2.3. Rational languages

For $A \subseteq \mathsf{iiPoms}$ we let

$$A{\downarrow} = \{P \in \mathsf{iiPoms} \mid \exists Q \in A : P \sqsubseteq Q\}$$

be its closure under subsumption. Note that $(A \cup B){\downarrow} = A{\downarrow} \cup B{\downarrow}$ for all $A, B \subseteq \mathsf{iiPoms}$, but for intersection this does *not* hold. For example it may happen that $A \cap B = \emptyset$ but $A{\downarrow} \cap B{\downarrow} \neq \emptyset$. A *language* is a subset $L \subseteq \mathsf{iiPoms}$ for which $L{\downarrow} = L$. The set of all languages is denoted $\mathscr{L} \subseteq 2^{\mathsf{iiPoms}}$.

The *width* of a language L is $\mathsf{wid}(L) = \sup\{\mathsf{wid}(P) \mid P \in L\}$. For $k \geq 0$ and $L \in \mathscr{L}$, denote $L_{\leq k} = \{P \in L \mid \mathsf{wid}(P) \leq k\}$. L is *k-dimensional* if $L = L_{\leq k}$. We let $\mathscr{L}_{\leq k} = \mathscr{L} \cap \mathsf{iiPoms}_{\leq k}$ denote the set of k-dimensional languages.

The *singleton ipomsets* are $[a]$ $[\bullet a]$, $[a\bullet]$ and $[\bullet a\bullet]$, for all $a \in \Sigma$. The *rational operations* $\cup$, $*$, $\|$ and (Kleene plus) $^+$ for languages are defined as follows.

$$\begin{aligned}
L * M &= \{P * Q \mid P \in L,\ Q \in M,\ T_P = S_Q\}{\downarrow},\\
L \parallel M &= \{P \parallel Q \mid P \in L,\ Q \in M\}{\downarrow},\\
L^+ &= \bigcup\nolimits_{n \geq 1} L^n, \qquad \text{for } L^1 = L, L^{n+1} = L * L^n.
\end{aligned}$$

The class of *rational languages* is the smallest subset of $\mathscr{L}$ that contains

$$\big\{\emptyset, \{\varepsilon\}, \{[a]\}, \{[\bullet a]\}, \{[a\bullet]\}, \{[\bullet a\bullet]\} \mid a \in \Sigma\big\}$$

(ε denotes the empty ipomset) and is closed under the rational operations.

Lemma 5 ([10]). *Any rational language has finite width.*

It immediately follows that the universal language iiPoms is *not* rational.

The *prefix quotient* of a language $L \in \mathscr{L}$ by an ipomset P is $P\backslash L = \{Q \in \mathsf{iiPoms} \mid PQ \in L\}$. Similarly, the *suffix quotient* of L by P is $L/P = \{Q \in \mathsf{iiPoms} \mid QP \in L\}$. Denoting

$$\mathsf{suff}(L) = \{P\backslash L \mid P \in \mathsf{iiPoms}\}, \qquad \mathsf{pref}(L) = \{L/P \mid P \in \mathsf{iiPoms}\},$$

we may now state the central result of [13].

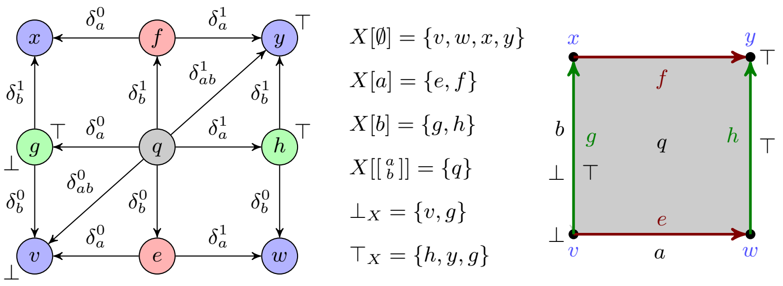


Figure 5: A two-dimensional HDA X on $\Sigma = \{a, b\}$, see Example 8.

Theorem 6 ([13]). *Let $L \in \mathscr{L}$ be a language. The following are equivalent:*

1. *L is rational;*
2. *$\mathsf{suff}(L)$ is finite;*
3. *$\mathsf{pref}(L)$ is finite.*

3. Higher-dimensional automata

An HDA is a collection of *cells* which are connected by *face maps*. Each cell contains a conclist of events which are active in it, and the face maps may terminate some events (*upper* faces) or "unstart" some events (*lower* faces), *i.e.*, map a cell to another in which the indicated events are not yet active.

To make this precise, let $\square$ denote the set of conclists. A *precubical set*[3]

$$X = (X, \mathsf{ev}, \{\delta^0_{A;U}, \delta^1_{A;U} \mid U \in \square, A \subseteq U\})$$

consists of a set of cells X together with a function $\mathsf{ev} : X \to \square$. For a conclist U we write $X[U] = \{x \in X \mid \mathsf{ev}(x) = U\}$ for the cells of type U. Further, for every $U \in \square$ and $A \subseteq U$ there are face maps $\delta^0_A, \delta^1_A : X[U] \to X[U - A]$ (omitting the extra subscript "U" from here on) which satisfy $\delta^\nu_A \delta^\mu_B = \delta^\mu_B \delta^\nu_A$ for $A \cap B = \emptyset$ and $\nu, \mu \in \{0, 1\}$. The upper face maps δ^1_A transform a cell x into one in which the events in A have terminated, whereas the lower face maps δ^0_A transform x into a cell where the events in A have not yet started. The *precubical identity* above expresses the fact that these transformations commute for disjoint sets of events.

A *higher-dimensional automaton* (*HDA*) $X = (X, \bot_X, \top_X)$ is a precubical set together with subsets $\bot_X, \top_X \subseteq X$ of *start* and *accept* cells. While HDAs may have an infinite number of cells, we will mostly be interested in finite HDAs. Thus, in the following we will omit the word "finite" and will be explicit when talking about infinite HDAs. The *dimension* of an HDA X is $\dim(X) = \sup\{|\mathsf{ev}(x)| \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$.

[3] We often refer to algebraic structures only by their underlying set.

Remark 7. *Precubical sets are presheaves over a category on objects $\square$, and then HDAs form a category with the induced morphisms, see [10].*

For a precubical set X and $k \geq 0$ we write $X_{\leq k} = \{x \in X \mid |\mathsf{ev}(x)| \leq k\}$ for its *k-skeleton*, *i.e.*, its restriction to cells of dimension at most k. It is clear that this is again a precubical set, and we use the same notation for HDAs.

A standard automaton is the same as a one-dimensional HDA $X = X_{\leq 1}$ with the property that for all $x \in \bot_X \cup \top_X$, $\mathsf{ev}(x) = \emptyset$: cells in $X[\emptyset]$ are states, cells in $X[\{a\}]$ for $a \in \Sigma$ are a-labeled transitions,[4] and face maps $\delta^0_{\{a\}}$ and $\delta^1_{\{a\}}$ attach source and target states to transitions. In contrast to standard automata, one-dimensional HDAs may have start and accept *transitions* instead of merely states, so languages of one-dimensional HDAs may contain words with interfaces.

Example 8. Figure 5 shows a two-dimensional HDA as a combinatorial object (left) and in a geometric realisation (right). It consists of nine cells: the corner cells $X_0 = \{x, y, v, w\}$ (depicted in blue) in which no event is active (for all $z \in X_0$, $\mathsf{ev}(z) = \emptyset$), the transition cells $X_1 = \{g, h, f, e\}$ (shown in red and green) in which one event is active ($\mathsf{ev}(f) = \mathsf{ev}(e) = a$ and $\mathsf{ev}(g) = \mathsf{ev}(h) = b$), and the (gray) square cell q where $\mathsf{ev}(q) = [\begin{smallmatrix} a \\ b \end{smallmatrix}]$.

The arrows between the cells on the left representation correspond to the face maps connecting them. For example, the upper face map δ^1_{ab} maps q to y because the latter is the cell in which the active events a and b of q have been terminated. On the right, face maps are used to glue cells together, so that for example $\delta^1_{ab}(q)$ is glued to the top right of q. In this and other geometric realisations, when we have two concurrent events a and b with $a \dashrightarrow b$, we will draw a horizontally and b vertically.

3.1. Regular languages

Computations of HDAs are *paths*, *i.e.*, sequences

$$\alpha = (x_0, \varphi_1, x_1, \ldots, x_{n-1}, \varphi_n, x_n)$$

consisting of cells x_i of X and symbols φ_i which indicate which type of step is taken: for every $i \in \{1, \ldots, n\}$, $(x_{i-1}, \varphi_i, x_i)$ is either

- $(\delta^0_A(x_i), \nearrow^A, x_i)$ for $A \subseteq \mathsf{ev}(x_i)$ (an *upstep*)
- or $(x_{i-1}, \searrow_A, \delta^1_A(x_{i-1}))$ for $A \subseteq \mathsf{ev}(x_{i-1})$ (a *downstep*).

Downsteps terminate events, following upper face maps, whereas upsteps start events by following inverses of lower face maps. Both types of steps may be empty.

The *source* and *target* of α as above are $\mathsf{src}(\alpha) = x_0$ and $\mathsf{tgt}(\alpha) = x_n$. The set of all paths in X starting at $Y \subseteq X$ and terminating in $Z \subseteq X$ is denoted

[4] We often denote conclists by only referring to the events' labels.

by $\mathsf{Path}(X)_Y^Z$. A path α is *accepting* if $\mathsf{src}(\alpha) \in \bot_X$ and $\mathsf{tgt}(\alpha) \in \top_X$. Paths α and β may be concatenated if $\mathsf{tgt}(\alpha) = \mathsf{src}(\beta)$. Their concatenation is written $\alpha * \beta$ or simply $\alpha\beta$.

Path equivalence is the congruence $\simeq$ generated by $(z \nearrow^A y \nearrow^B x) \simeq (z \nearrow^{A\cup B} x)$, $(x \searrow_A y \searrow_B z) \simeq (x \searrow_{A\cup B} z)$, and $\gamma\alpha\delta \simeq \gamma\beta\delta$ whenever $\alpha \simeq \beta$. Intuitively, this relation allows to assemble subsequent upsteps or downsteps into one bigger step. A path is *sparse* if its upsteps and downsteps are alternating, so that no more such assembling may take place. Every equivalence class of paths contains a unique sparse path.

The observable content or *event ipomset* $\mathsf{ev}(\alpha)$ of a path α is defined recursively as follows:

- if $\alpha = (x)$, then $\mathsf{ev}(\alpha) = \mathsf{id}_{\mathsf{ev}(x)}$;
- if $\alpha = (y \nearrow^A x)$, then $\mathsf{ev}(\alpha) = {}_A{\uparrow}\mathsf{ev}(x)$;
- if $\alpha = (x \searrow_A y)$, then $\mathsf{ev}(\alpha) = \mathsf{ev}(x){\downarrow}_A$;
- if $\alpha = \alpha_1 * \cdots * \alpha_n$ is a concatenation, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \cdots * \mathsf{ev}(\alpha_n)$.

Note that upsteps in α correspond to starters in $\mathsf{ev}(\alpha)$ and downsteps correspond to terminators. Path equivalence $\alpha \simeq \beta$ implies $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$ [10]. Further, if $\alpha = \alpha_1 * \cdots * \alpha_n$ is a sparse path, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \cdots * \mathsf{ev}(\alpha_n)$ is a sparse step decomposition.

The *language* of an HDA X is $\mathsf{L}(X) = \{\mathsf{ev}(\alpha) \mid \alpha \text{ accepting path in } X\}$.

Example 9. The HDA X of Figure 5 admits several accepting paths with target h, for example $v \nearrow^{ab} q \searrow_a h$. This is a sparse path and equivalent to the non-sparse paths $v \nearrow^a e \nearrow^b q \searrow_a h$ and $v \nearrow^b g \nearrow^a q \searrow_a h$. Their event ipomset is $\left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right]$. In addition, since g is both a start and accept cell, we have also g and $v \nearrow^b g$ as accepting paths, with event ipomsets $\bullet b\bullet$ and $b\bullet$, respectively. We have $\mathsf{L}(X) = \{b\bullet, \bullet b\bullet, \left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ \bullet b\bullet \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}\right]\}{\downarrow}$.

The following property, that languages of skeleta are width restrictions of languages, is clear.

Lemma 10. *For any HDA X and $k \geq 0$, $\mathsf{L}(X_{\leq k}) = \mathsf{L}(X)_{\leq k}$.*

In the lemma below, we write $\mathsf{Path}(X)_Y = \mathsf{Path}(X)_Y^X$, $\mathsf{Path}(X)^Z = \mathsf{Path}(X)_X^Z$, and $\mathsf{Path}(X) = \mathsf{Path}(X)_X^X$.

Lemma 11 ([13]). *Let X be an HDA, $x, y \in X$ and $\gamma \in \mathsf{Path}(X)_x^y$. Assume that $\mathsf{ev}(\gamma) = P * Q$ for ipomsets P and Q. Then there exist paths $\alpha \in \mathsf{Path}(X)_x$ and $\beta \in \mathsf{Path}(X)^y$ such that $\mathsf{ev}(\alpha) = P$, $\mathsf{ev}(\beta) = Q$ and $\mathsf{tgt}(\alpha) = \mathsf{src}(\beta)$.*

Lemma 12. *Let X be an HDA, $P \in \mathsf{L}(X)$ and $P = P_1 * \cdots * P_n$ be any decomposition (not necessarily a step decomposition). Then there exists an accepting path $\alpha = \alpha_1 * \cdots * \alpha_n$ in X such that $\mathsf{ev}(\alpha_i) = P_i$ for all i. If $P = P_1 * \cdots * P_n$ is a sparse step decomposition, then $\alpha = \alpha_1 * \cdots * \alpha_n$ is sparse.*

Proof. The first claim follows from Lemma 11 by induction. As to the second, if starters and terminators are alternating in $P_1 * \dots * P_n$, then upsteps and downsteps are alternating in $\alpha_1 * \dots * \alpha_n$.

Languages of HDAs are sets of (interval) ipomsets which are closed under subsumption [10], *i.e.*, languages in our sense. A language is *regular* if it is the language of a finite HDA.

Theorem 13 ([10]). *A language is regular iff it is rational.*

Remark 14. *Every ipomset P may be converted into a* track object $\square^P$*, see [10], which is an HDA with the property that for any HDA X, $P \in \mathsf{L}(X)$ iff there is a morphism $\square^P \to X$. This is notably useful in proofs, for example of Lemma 11 above.*

4. Regular and non-regular languages

We exhibit a property, similar to the pumping lemma for word languages, which allows to show that some languages are non-regular. Afterwards we show that intersections of regular languages are again regular.

4.1. Pumping lemma

Lemma 15. *Let L be a regular language. There exists $k \in \mathbb{N}$ such that for any $P \in L$, any decomposition $P = Q_1 * \dots * Q_n$ with $n > k$ and any $0 \leq m \leq n - k$ there exist i, j such that $m \leq i < j \leq m + k$ and $Q_1 * \dots * Q_i * (Q_{i+1} * \dots * Q_j)^r * Q_{j+1} * \dots * Q_n \in L$ for every $r \geq 1$.*

Proof. Let X be an HDA accepting L and $k > |X|$. By Lemma 12 there exists an accepting path $\alpha = \alpha_1 * \dots * \alpha_n$ such that $\mathsf{ev}(\alpha_i) = Q_i$ for all i, and $\mathsf{ev}(\alpha) = P$. Denote $x_i = \mathsf{tgt}(\alpha_i) = \mathsf{src}(\alpha_{i+1})$. Amongst the cells $x_m, \dots, x_{m+k}$ there are at least two equal, say $x_i = x_j$, $m \leq i < j \leq m + k$. As a consequence, $\mathsf{src}(\alpha_{i+1}) = \mathsf{tgt}(\alpha_j)$, and for every $r \geq 1$,

$$\alpha_1 * \dots * \alpha_i * (\alpha_{i+1} * \dots * \alpha_j)^r * \alpha_{j+1} * \dots * \alpha_n$$

is an accepting path that recognizes

$$Q_1 * \dots * Q_i * (Q_{i+1} * \dots * Q_j)^r * Q_{j+1} * \dots * Q_n.$$

□

Corollary 16. *Let L be a regular language. There exists $k \in \mathbb{N}$ such that any $P \in L$ with $\mathsf{size}(P) > k$ can be decomposed into $P = Q_1 * Q_2 * Q_3$ such that Q_2 is not an identity and $Q_1 * Q_2^+ * Q_3 \subseteq L$.*

The proof follows by applying Lemma 15 to a dense step decomposition $P = Q_1 * \dots * Q_{2\,\mathsf{size}(P)}$, *cf.* Lemma 4. We may now expose a language which is not regular.

Proposition 17. *The language* $L = \{[\begin{smallmatrix} a \\ a \end{smallmatrix}]^n * a^n \mid n \geq 1\}{\downarrow}$ *is not regular.*

Note that the restriction $L_{\leq 1} = (aaa)^+$ *is* regular, showing that regularity of languages may not be decided by restricting to their one-dimensional parts.

Proof (of Proposition 17). We give two proofs. The first uses Theorem 6: for every $k \geq 1$, $[\begin{smallmatrix} a \\ a \end{smallmatrix}]^k \backslash L = \{[\begin{smallmatrix} a \\ a \end{smallmatrix}]^n * a^{n+k} \mid n \geq 0\}{\downarrow}$, and these are different for different k, so $\mathsf{suff}(L)$ is infinite.

The second proof uses Lemma 15. Assume L to be regular, let k be the constant from the lemma, and take $P = [\begin{smallmatrix} a \\ a \end{smallmatrix}]^k * a^k = Q_1 * \cdots * Q_k * Q_{k+1}$, where $Q_1 = \cdots = Q_k = [\begin{smallmatrix} a \\ a \end{smallmatrix}]$ and $Q_{k+1} = a^k$. For $m = 0$ we obtain that $[\begin{smallmatrix} a \\ a \end{smallmatrix}]^{k+(j-i)r}\, a^k \in L$ for all r and some $j - i > 0$: a contradiction. □

We may strengthen the above result to show that regularity of languages may not be decided by restricting to their k-dimensional parts for any $k \geq 1$. For $a \in \Sigma$ let $a^{\|_1} = a$ and $a^{\|_k} = a \parallel a^{\|_{k-1}}$ for $k \geq 2$: the k-fold parallel product of a with itself. Now let $k \geq 1$ and

$$L = \big\{(a^{\|_{k+1}})^n * P^n \;\big|\; n \geq 0, P \in \{a^{\|_{k+1}}\}{\downarrow} - \{a^{\|_{k+1}}\}\big\}{\downarrow}.$$

The idea of the condition on P is to remove from $\{a^{\|_{k+1}}\}{\downarrow}$ precisely the only ipomset of width $k+1$. Using the same arguments as above one can show that L is not regular, but $L_{\leq k} = ((\{a^{\|_{k+1}}\}{\downarrow} - \{a^{\|_{k+1}}\})^2)^+$ is.

Yet the k-restrictions of any regular language remain regular:

Proposition 18. *Let* $k \geq 0$. *If* $L \in \mathscr{L}$ *is regular, then so is* $L_{\leq k}$.

Proof. By Lemma 10. □

4.2. Intersection of regular languages

By definition, the regular languages are closed under union, parallel composition, gluing composition, and Kleene plus. Here we show that they are also closed under intersection. (For complement this is more complicated, as we will see later.)

Proposition 19. *The regular languages are closed under* $\cap$.

Proof. We again give two proofs, one algebraic using Theorem 6 and another, constructive proof using Theorem 13. For the first proof, let L_1 and L_2 be regular, then $\mathsf{suff}(L_1)$ and $\mathsf{suff}(L_2)$ are both finite. Now

$$\begin{aligned}
&\mathsf{suff}(L_1 \cap L_2)\\
&\qquad = \{P\backslash(L_1 \cap L_2) \mid P \in \mathsf{iiPoms}\}\\
&\qquad = \big\{\{Q \in \mathsf{iiPoms} \mid PQ \in L_1 \cap L_2\} \;\big|\; P \in \mathsf{iiPoms}\big\}\\
&\qquad = \big\{\{Q \in \mathsf{iiPoms} \mid PQ \in L_1\} \cap \{Q \in \mathsf{iiPoms} \mid PQ \in L_2\} \;\big|\; P \in \mathsf{iiPoms}\big\}\\
&\qquad = \{P\backslash L_1 \cap P\backslash L_2 \mid P \in \mathsf{iiPoms}\}\\
&\qquad \subseteq \{M_1 \cap M_2 \;\big|\; M_1 \in \mathsf{suff}(L_1),\ M_2 \in \mathsf{suff}(L_2)\}
\end{aligned}$$

which is thus finite.

For the second, constructive proof, let X_1 and X_2 be HDAs. We construct an HDA X with $\mathsf{L}(X) = \mathsf{L}(X_1) \cap \mathsf{L}(X_2)$:

$$\begin{aligned} X = \{(x_1, x_2) \in X_1 \times X_2 \mid \mathsf{ev}_1(x_1) &= \mathsf{ev}_2(x_2)\}, \\ \delta^\nu_A(x_1, x_2) &= (\delta^\nu_A(x_1), \delta^\nu_A(x_2)), \\ \mathsf{ev}((x_1, x_2)) &= \mathsf{ev}_1(x_1) = \mathsf{ev}_2(x_2), \\ \bot &= \bot_1 \times \bot_2, \qquad \top = \top_1 \times \top_2. \end{aligned} \tag{1}$$

For the inclusion $\mathsf{L}(X) \subseteq \mathsf{L}(X_1) \cap \mathsf{L}(X_2)$, any accepting path α in X projects to accepting paths β in X_1 and γ in X_2, and then $\mathsf{ev}(\beta) = \mathsf{ev}(\gamma) = \mathsf{ev}(\alpha)$. For the reverse inclusion, we need to be slightly more careful to ensure that accepting paths in X_1 and X_2 may be assembled to an accepting path in X.

Let $P \in \mathsf{L}(X_1) \cap \mathsf{L}(X_2)$ and $P = P_1 * \cdots * P_n$ the sparse step decomposition. Let $\beta = \beta_1 * \cdots * \beta_n$ and $\gamma = \gamma_1 * \cdots * \gamma_n$ be sparse accepting paths for P in X_1 and X_2, respectively, such that $\mathsf{ev}(\alpha_i) = \mathsf{ev}(\beta_i) = P_i$ for all i, *cf.* Lemma 12.

Let $i \in \{1, \dots, n\}$ and assume that $P_i = {}_A{\uparrow}U$ is a starter, then $\beta_i = (\delta^0_A x_1, \nearrow^A, x_1)$ and $\gamma_i = (\delta^0_A x_2, \nearrow^A, x_2)$ for $x_1 \in X_1$ and $x_2 \in X_2$ such that $\mathsf{ev}(x_1) = \mathsf{ev}(x_2) = U$. Hence we may define a step $\alpha_i = (\delta^0_A(x_1, x_2), \nearrow^A, (x_1, x_2))$ in X. If P_i is a terminator, the argument is similar. By construction, $\mathsf{tgt}(\alpha_i) = \mathsf{src}(\alpha_{i+1})$, so the steps α_i assemble to an accepting path $\alpha = \alpha_1 * \cdots * \alpha_n \in \mathsf{Path}(X)^\top_\bot$, and $\mathsf{ev}(\alpha) = P$. $\square$

Remark 20. *The HDA X constructed in* (1) *above is the product in the category of HDAs. This gives a third and more high-level proof of Proposition 19, using track objects and the universal property of the product.*

5. ST-automata

In the following, we define *ST-automata* whose languages are sets of words over an alphabet of starters and terminators. We use here the most recent definition of these objects, introduced in [2], which subsumes other variants that have been used in [1, 4, 7, 8, 10]. We use a construction from [2] which translates HDAs into ST-automata. This will be useful for showing properties of HDA languages.

Let us first introduce some notation. Let $\Omega = \{{}_A{\uparrow}U \mid U \in \square, A \subseteq U\} \cup \{U{\downarrow}_B \mid U \in \square, B \subseteq U\}$ be the (infinite) set of starters and terminators over Σ and $\mathsf{Id} = \{\mathsf{id}_U \mid U \in \square\} \subseteq \Omega$ the (infinite) set of identities. Further, for any $k \geq 0$, let $\Omega_{\leq k} = \Omega \cap \mathsf{iiPoms}_{\leq k}$ and $\mathsf{Id}_{\leq k} = \mathsf{Id} \cap \mathsf{iiPoms}_{\leq k}$. Note that both these sets are finite and $\mathsf{Id}_{\leq k} \subseteq \Omega_{\leq k}$.

An *ST-automaton* is a structure $A = (Q, E, I, F, \lambda)$ consisting of a set of states Q, subsets $I, F \subseteq Q$ of initial and accepting states, transitions $E \subseteq Q \times \Omega \times Q$, and a function $\lambda : Q \to \square$ such that for all $(q, {}_S U_T, r) \in E$, $\lambda(q) = S$ and $\lambda(r) = T$. Thus, an ST-automaton is a standard automaton over Ω where the states are also labeled by $\square$ consistently with the labeling of incoming and outgoing edges.

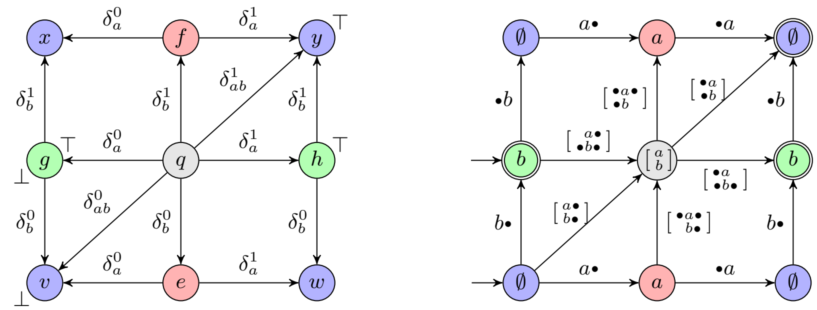


Figure 6: HDA of Figure 5 (left) and its ST-automaton (right; identity loops not displayed). In the HDA, names of cells are shown; in the ST-automaton, we show labels instead.

A *path* in an ST-automaton A is defined as an alternating sequence $\pi = (q_0, e_1, q_1, \dots, e_n, q_n)$ of states q_i and transitions e_i such that $e_i = (q_{i-1}, P_i, q_i)$ for every $i = 1, \dots, n$ and some $P_1, \dots, P_n \in \Omega$. The path is *accepting* if $q_0 \in I$ and $q_n \in F$. The *label* of π is $\ell(\pi) = \mathsf{id}_{\lambda(q_0)} P_1 \mathsf{id}_{\lambda(q_1)} \dots P_n \mathsf{id}_{\lambda(q_n)}$. The *language* of an ST-automaton A is

$$\mathsf{W}(A) = \{\ell(\pi) \mid \pi \text{ accepting path in } A\};$$

we use the symbol W to distinguish such languages (over Ω) from languages of HDAs (denoted by L) and will use the term “word language” if confusion may arise.

Note that languages of ST-automata are subsets of $\mathsf{Id}_{\le k}\, (\Omega_{\le k} \mathsf{Id}_{\le k})^*$. In particular, the labeling of states and their consideration in the path labels forbid to have the empty word ε in the language of an ST-automata. (The empty *ipomset* $\mathsf{id}_\emptyset$, however, may well be contained in such a language.)

To a given HDA $X = (X, \bot, \top)$ we associate an ST-automaton $\mathsf{ST}(X) = (Q, E, I, F, \lambda)$ as follows:

- $Q = X$, $I = \bot$, $F = \top$, $\lambda = \mathsf{ev}$, and
- $E = \{(\delta_A^0(q), {}_A{\uparrow}\mathsf{ev}(q), q) \mid A \subseteq \mathsf{ev}(q)\} \cup \{(q, \mathsf{ev}(q){\downarrow}_A, \delta_A^1(q)) \mid A \subseteq \mathsf{ev}(q)\}$.

That is, the transitions of $\mathsf{ST}(X)$ precisely mimic the starting and terminating of events in X. Note that given an HDA X of dimension $\dim(X) = k \in \mathbb{N}$, $\mathsf{ST}(X)$ is over the finite alphabet $\Omega_{\le k}$. Note also that lower faces in X are inverted to get the starting transitions.

In what follows, we consider languages of nonempty words over Ω, which we denote by W etc. and the class of such languages by $\mathscr{W}$.

Example 21. Figure 6 displays the ST-automaton $\mathsf{ST}(X)$ pertaining to the HDA X in Figure 5, with the identity loops $(z, \mathsf{id}_{\mathsf{ev}(z)}, z)$ for all states z omitted. Notice that the transitions between a cell and its lower face are opposite to the face maps in X. The smallest accepting path of $\mathsf{ST}(X)$ is $\omega = (g)$ where $\lambda(g) = b$. Its labeling is $\ell(\omega) = {\bullet}b{\bullet}$. More generally, $\mathsf{W}(\mathsf{ST}(X)) = \{{\bullet}b{\bullet}, \mathsf{id}_\emptyset b{\bullet}{\bullet}b{\bullet}, \mathsf{id}_\emptyset \left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a\bullet \\ \bullet b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a \\ \bullet b\bullet \end{smallmatrix}\right] {\bullet}b{\bullet}, {\bullet}b{\bullet} \left[\begin{smallmatrix} a\bullet \\ \bullet b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a\bullet \\ \bullet b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a \\ \bullet b\bullet \end{smallmatrix}\right], \mathsf{id}_\emptyset \left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a\bullet \\ \bullet b\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet a \\ \bullet b \end{smallmatrix}\right] \mathsf{id}_\emptyset, \dots\}$.

Define functions $\Phi : \mathscr{L} \to \mathscr{W}$ and $\Psi : \mathscr{W} \to \mathscr{L}$ by

$$\begin{aligned}\Phi(L) &= \{P_1 \cdots P_n \in \Omega^* \mid P_1 * \cdots * P_n \in L,\ n \geq 1,\ \forall i : P_i \in \Omega\},\\ \Psi(W) &= \{P_1 * \cdots * P_n \in \mathsf{iiPoms} \mid P_1 \cdots P_n \in W,\ n \geq 1,\ \forall i : T_{P_i} = S_{P_{i+1}}\}{\downarrow}.\end{aligned}$$

Φ translates ipomsets into concatenations of their step decompositions, and Ψ translates words of composable starters and terminators into their ipomset composition (and takes subsumption closure). Hence Φ creates "coherent" words, *i.e.*, nonempty concatenations of starters and terminators with matching interfaces. Conversely, Ψ disregards all words which are not coherent in that sense. Every ipomset is mapped by Φ to infinitely many words over Ω (because ipomsets $\mathsf{id}_U \in \Omega$ are not units in $\mathscr{W}$). This will not be a problem for us later. It is clear that $\Psi(\Phi(L)) = L$ for all $L \in \mathscr{L}$, since every ipomset has a step decomposition. For the other composition, neither $\Phi(\Psi(W)) \subseteq W$ nor $W \subseteq \Phi(\Psi(W))$ hold:

Example 22. If $W = \{a{\bullet}\, {\bullet}b\}$ (the word language containing the concatenation of $a\bullet$ and $\bullet b$), then $\Psi(W) = \emptyset$ and thus $\Phi(\Psi(W)) = \emptyset \not\supseteq W$. If $W = \{[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}][\begin{smallmatrix} \bullet a \\ \bullet b \end{smallmatrix}]\}$, then $\Psi(W) = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}], ab, ba\}$ and $\Phi(\Psi(W)) \not\subseteq W$.

Lemma 23. *For all $A_1, A_2 \subseteq \mathsf{iiPoms}$, $A_1{\downarrow} * A_2{\downarrow} = \{P_1 * P_2 \mid P_1 \in A_1,\ P_2 \in A_2\}{\downarrow}$.*

Proof. Let $R \in A_1{\downarrow} * A_2{\downarrow}$. By definition, there exists $P_i' \in A_i{\downarrow}$ such that $R \sqsubseteq P_1' * P_2'$. Let $P_i \in A_i$ such that $P_i' \sqsubseteq P_i$. Then $R \sqsubseteq P_1 * P_2$. The other inclusion follows from the facts that $A_i \subseteq A_i{\downarrow}$ and that the gluing composition preserves subsumption. □

As trivial consequences of the definitions, we have that Φ respects boolean operations:

Lemma 24. *For all $L_1, L_2 \in \mathscr{L}$, $\Phi(L_1 \cap L_2) = \Phi(L_1) \cap \Phi(L_2)$ and $\Phi(L_1 \cup L_2) = \Phi(L_1) \cup \Phi(L_2)$.*

Yet Φ does *not* respect concatenations: only inclusion $\Phi(L * L') \subseteq \Phi(L)\Phi(L')$ holds, given that $\Phi(L)\Phi(L')$ also may contain words in Ω^* that are not composable in iiPoms.

As to Ψ, we show that it respects regular operations:

Lemma 25. *For all $W_1, W_2 \in \mathscr{W}$, $\Psi(W_1 \cup W_2) = \Psi(W_1) \cup \Psi(W_2)$, $\Psi(W_1 W_2) = \Psi(W_1) * \Psi(W_2)$, and $\Psi(W_1^+) = \Psi(W_1)^+$.*

Proof. The first claim follows easily using the fact that $(A \cup B){\downarrow} = A{\downarrow} \cup B{\downarrow}$. For the second, we have

$$\begin{aligned}\Psi(W_1) * \Psi(W_2) &= \{P_1 * \cdots * P_n \mid P_1 \cdots P_n \in W_1,\ \forall i : T_{P_i} = S_{P_{i+1}}\}{\downarrow}\\ &\qquad * \{Q_1 * \cdots * Q_m \mid Q_1 \cdots Q_m \in W_2,\ \forall i : T_{Q_i} = S_{Q_{i+1}}\}{\downarrow}\\ &= \{P_1 * \cdots * P_n * P_{n+1} * \cdots * P_{n+m} \mid P_1 \cdots P_n \in W_1,\\ &\qquad P_{n+1} \cdots P_{n+m} \in W_2,\ \forall i : T_{P_i} = S_{P_{i+1}}\}{\downarrow} \text{ (by Lemma 23)}\\ &= \Psi(W_1 W_2).\end{aligned}$$

The equality $\Psi(W_1^+) = \Psi(W_1)^+$ then follows by trivial recurrence, using the equalities for binary union and gluing composition. □

On the other hand, Ψ does not respect intersections, given that $(A \cap B){\downarrow} = A{\downarrow} \cap B{\downarrow}$ does not always hold.

Let $\mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^* \subseteq \Omega_{\le k}^*$ (which is a regular word language) denote the set of all words over $\Omega_{\le k}$ starting and ending with an identity and where identities and starters/terminators alternate. We have the following:

Lemma 26. *For any HDA X and $k \ge \dim(X)$,*

$$\mathsf{W}(\mathsf{ST}(X)) = \Phi(\mathsf{L}(X)) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^*.$$

Proof. By construction, there is a one-to-one correspondence between the accepting paths in X and $\mathsf{ST}(X)$:

$$\alpha = (x_0, \varphi_1, x_1, \varphi_2, \dots, \varphi_n, x_n) \mapsto \big(x_0 \xrightarrow{\psi_1} x_1 \xrightarrow{\psi_2} \cdots \xrightarrow{\psi_n} x_n\big) = \omega$$

where ψ_i is the starter or terminator corresponding to the step φ_i and $\lambda(x_i) = \mathsf{id}_{\mathsf{ev}(x_i)}$. Note also that the path (x) is accepting in X if and only if it is accepting in $\mathsf{ST}(X)$. In addition $\mathsf{ev}(x) = \ell(x)$.

Now let $\mathsf{id}_0 P_1 \mathsf{id}_1 \cdots P_n \mathsf{id}_n \in \mathsf{W}(\mathsf{ST}(X))$ with $n > 1$. Then there is an accepting path ω in $\mathsf{ST}(X)$ such that $P_i = \mathsf{ev}(x_{i-1}, \varphi_i, x_i) = \psi_i$, $\mathsf{id}_0 = \mathsf{id}_{\mathsf{ev}(x_{i-1})}$ and $\mathsf{id}_i = \mathsf{id}_{\mathsf{ev}(x_i)}$ for $1 \le i \le n$. The corresponding path α in X is accepting. Hence $P_1 * \cdots * P_n = \mathsf{id}_0 * P_1 * \mathsf{id}_1 * \cdots * P_n * \mathsf{id}_n = \mathsf{ev}(\alpha) \in \mathsf{L}(X)$, and $\mathsf{id}_0 P_1 \mathsf{id}_1 \cdots P_n \mathsf{id}_n \in \Phi(\mathsf{L}(X))$. This shows the inclusion $\subseteq$.

Now let $\mathsf{id}_0 P_1 \mathsf{id}_1 \cdots P_n \mathsf{id}_n \in \Phi(\mathsf{L}(X)) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^*$. Thus $\mathsf{id}_0 * P_1 * \mathsf{id}_1 * \cdots * P_n * \mathsf{id}_n \in \mathsf{L}(X)$. Using Lemma 12 we conclude that there exists an accepting path $\alpha = \mathsf{src}(\beta_1) * \beta_1 * \mathsf{tgt}(\beta_1) * \cdots * \mathsf{src}(\beta_n) * \beta_n * \mathsf{tgt}(\beta_n)$ in X such that $\mathsf{ev}(\beta_i) = P_i$, $\mathsf{ev}(\mathsf{src}(\beta_i)) = \mathsf{id}_{i-1}$ and $\mathsf{ev}(\mathsf{tgt}(\beta_i) = \mathsf{id}_i$. The path ω corresponding to α recognizes $\mathsf{id}_0 P_1 \mathsf{id}_1 \cdots P_n \mathsf{id}_n$, which shows the inclusion $\supseteq$. □

Lemma 27. *Let $k \ge 0$. For all languages $L_1, L_2 \in \mathscr{L}_{\le k}$, $L_1 \subseteq L_2$ if and only if $\Phi(L_1) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^* \subseteq \Phi(L_2) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^*$.*

Proof. The forward implication is immediate from Lemma 24. Now if $L_1 \not\subseteq L_2$, then also $\Phi(L_1) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^* \not\subseteq \Phi(L_2) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^*$, since every ipomset admits a step decomposition starting with an identity. □

Theorem 28. *Inclusion of regular languages is decidable.*

Proof. Let L_1 and L_2 be regular and recognized by HDAs X_1 and X_2, respectively, and let $k = \max(\dim(X_1), \dim(X_2))$. By Lemmas 26 and 27,

$$\begin{aligned} L_1 \subseteq L_2 &\iff \Phi(L_1) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^* \subseteq \Phi(L_2) \cap \mathsf{Id}_{\le k}\,(\Omega_{\le k}\mathsf{Id}_{\le k})^* \\ &\iff \mathsf{W}(\mathsf{ST}(X_1)) \subseteq \mathsf{W}(\mathsf{ST}(X_2)). \end{aligned}$$

Given that these are finite automata, the latter inclusion is decidable. □

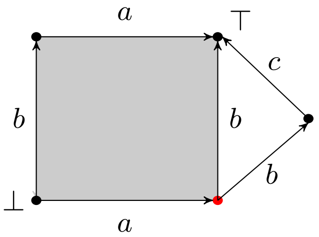


Figure 7: Non-deterministic HDA accepting the language of Example 32.

6. Determinism and ambiguity

It is shown in [13] that not all regular languages may be recognized by deterministic HDAs. We recall the definition from [13]. A cell $x \in X$ in an HDA X is *essential* if there exists an accepting path in X that contains x. A path is essential if all its cells are essential.

Definition 29. An HDA X is *deterministic* if

- it has at most one start cell of every type: for all $U \in \square$, $|\bot_X \cap X[U]| \leq 1$;
- for every $U \in \square$, $A \subseteq U$, and essential cell $x \in X[U - A]$, there exists at most one cell $y \in X[U]$ such that $\delta_A(y) = x$.

We also say that a regular language is *deterministic* if it is recognized by a finite deterministic HDA. The following is obvious.

Lemma 30. *If X is a deterministic HDA, then its 1-skeleton $X_{\leq 1}$ is a deterministic finite automaton.*

Lemma 31 ([13]). *There exist regular languages which are not deterministic.*

Hence there exist HDAs which may not be determinized. By Lemma 30 it is clear that any example of such a non-determinizable HDA must be at least two-dimensional. The example below, taken from [13], provides such an HDA.

Example 32. Let $L = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}], abc\}{\downarrow} = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}], ab, ba, abc\}$. Figure 7 shows an HDA X which recognizes L. Note that X is not deterministic since its bottom-right cell (the red one) has *two* outgoing b-labeled edges. In Example 36 below we will see that L itself is non-deterministic: it is not recognized by any deterministic HDA.

We will below prove a strengthening of Lemma 31, introducing a notion of *ambiguity* and showing that there exist regular languages with unbounded ambiguity.

6.1. Decidability of determinism

First we show that determinism is decidable. To do so, we use a criterion developed in [13] characterizing deterministic languages.

Definition 33. A language L is *swap-invariant* if it holds for all $P, Q, P', Q' \in \mathsf{iiPoms}$ that $PP' \in L, QQ' \in L$ and $P \sqsubseteq Q$ imply $QP' \in L$.

Lemma 34 ([13]). *A language L is swap-invariant if and only if $P \sqsubseteq Q$ implies $P\backslash L = Q\backslash L$ for all $P, Q \in \mathsf{iiPoms}$ for which $Q\backslash L = \emptyset$.*

Theorem 35 ([13]). *A language L is deterministic if and only if it is swap-invariant.*

Example 36. Continuing Example 32, note that $ab \sqsubseteq \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ but $ab\backslash L = \{\varepsilon, c\} \neq \{\varepsilon\} = \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]\backslash L$. Thus L is not swap-invariant and hence not a deterministic regular language.

Lemma 37. *Determinism is preserved by intersection, but neither by union nor Kleene plus.*

Proof. To show preservation by intersection, let L_1 and L_2 be deterministic languages and $P, P', Q, Q' \in L = L_1 \cap L_2$ such that $PP' \in L$, $QQ' \in L$ and $P \sqsupseteq Q$. Then, as L_1 and L_2 are swap-invariant, $QP' \in L_1$ and $QP' \in L_2$ and thus $QP' \in L$. We have shown that L is swap invariant and thus deterministic.

For union, consider $L_1 = \{\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right] aa\}$ and $L_2 = \{aa \left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]\}$. Both L_1 and L_2 are deterministic, but $L_1 \cup L_2$ is not.

As to Kleene plus, consider $L = \{b\bullet, \left[\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}\right] cd, ab \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right]\}$, then L is deterministic, yet Proposition 43 below implies that L^+ is not. □

In the following let L be a regular language accepted by some HDA X. Let

$$\mathsf{Pre}(X) = \{\mathsf{ev}(\alpha) \mid \alpha = (x_0, \varphi_1, \dots, \varphi_n, x_n) \in \mathsf{Path}(X)_\perp, \forall i \neq j : x_i \neq x_j\},$$

that is the event ipomsets of all paths of X starting at any initial cell and that do not loop back onto themselves.

Lemma 38. *Let $P \in \mathsf{iiPoms}$. Then $P\backslash L \in \mathsf{suff}(L)$ if and only if there exists $Q \in \mathsf{Pre}(X)$ such that $P\backslash L = Q\backslash L$.*

Proof. The proof is immediate if $P \in \mathsf{Pre}(X)$. For the opposite it suffices to take $Q \in \mathsf{Pre}(X)$ whose path has the same target as the one of P.

As corollaries we have:

Corollary 39. $\mathsf{suff}(L) = \{P\backslash L \mid P \in \mathsf{Pre}(X)\}$.

Corollary 40. *For all $P \in \mathsf{iiPoms}$, either $P\backslash L = \emptyset$ or there exists $Q \in \mathsf{Pre}(X)$ such that $P\backslash L = Q\backslash L$.*

Theorem 41. *It is decidable whether a regular language L is deterministic.*

Proof. Let X be an HDA that recognizes L and notice that the set $\mathsf{Pre}(X)$ is finite and computable. In addition, for all $P \in \mathsf{Pre}(X)$, $P\backslash L$ is accepted by the HDA Y built as follows. For all $x \in X$ such that there exists a non-looping $\alpha \in \mathsf{Path}(X)_{\perp}^{x}$ and $P = \mathsf{ev}(\alpha)$, let $(X_x, \{x\}, \top_X)$ be a copy of X. Then Y is the union of such X_x's. We have now an HDA for each element of $\mathsf{suff}(L)$.

Now, L is deterministic if and only if it is swap-invariant (Theorem 35), if and only if $P \sqsubseteq Q$ implies $P\backslash L = Q\backslash L$ for all $P, Q \in \mathsf{iiPoms}$ for which $Q\backslash L = \emptyset$ (Lemma 34). But, by Lemma 38 and Corollary 40, it suffices to check Lemma 34 for $P, Q \in \mathsf{Pre}(X)$, that is check for all $P, Q \in \mathsf{Pre}(X)$ such that $P \sqsubseteq Q$ and $Q\backslash L \neq \emptyset$, whether $P\backslash L = Q\backslash L$. Since we can easily check subsumption, the theorem follows from the decidability of inclusion (Theorem 28). □

6.2. Ambiguity

We now introduce a notion of ambiguity for HDAs and show that languages of unbounded ambiguity exist.

Definition 42. Let $k \geq 1$. An HDA X is *k-ambiguous* if every $P \in \mathsf{L}(X)$ is the event ipomset of at most k sparse accepting paths in X.

A language L is said to be *of bounded ambiguity* if it is recognized by a k-ambiguous HDA for some k. Deterministic HDAs are 1-ambiguous, and deterministic regular languages are of bounded ambiguity.

Proposition 43. *The regular language $L = (\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right] cd + ab \left[\begin{smallmatrix}c\\d\end{smallmatrix}\right])^{+}$ is of unbounded ambiguity.*

Before the proof, we need a lemma about the structure of accepting paths in any HDA which accepts L.

Lemma 44. *Let X be an HDA with $\mathsf{L}(X) = L$. Let α and β be essential sparse paths in X with $\mathsf{ev}(\alpha) = \left[\begin{smallmatrix}a\\b\end{smallmatrix}\right] cd$ and $\mathsf{ev}(\beta) = ab \left[\begin{smallmatrix}c\\d\end{smallmatrix}\right]$. Then*

$$\begin{aligned}\alpha &= (v \nearrow^{ab} q \searrow_{ab} x \nearrow^{c} e \searrow_{c} y \nearrow^{d} f \searrow_{d} z),\\ \beta &= (v' \nearrow^{a} g \searrow_{a} w' \nearrow^{b} h' \searrow_{b} x' \nearrow^{cd} r' \searrow_{cd} z')\end{aligned}$$

for some $v, x, y, z, v', w', x', z' \in X[\varepsilon]$, $e \in X[c]$, $f \in X[d]$, $g' \in X[a]$, $h' \in X[b]$, $q \in X[[\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right]]]$, $r' \in X[[\left[\begin{smallmatrix}c\\d\end{smallmatrix}\right]]]$. Furthermore, $x \neq x'$, and for

$$\begin{aligned}\bar{\alpha} &= (v \nearrow^{a} \delta_b^0(q) \searrow_{a} \delta_a^0\delta_a^1(q) \nearrow^{b} \delta_a^1(q) \searrow_{b} x \nearrow^{c} e \searrow_{c} y \nearrow^{d} f \searrow_{d} z),\\ \bar{\beta} &= (v' \nearrow^{a} g \searrow_{a} w' \nearrow^{b} h' \searrow_{b} x' \nearrow^{c} \delta_d^0(r') \searrow_{c} \delta_d^0\delta_c^1(r') \nearrow^{d} \delta_c^1(r') \searrow_{d} z')\end{aligned}$$

we have $\mathsf{ev}(\bar{\alpha}) = \mathsf{ev}(\bar{\beta}) = abcd$ and $\bar{\alpha} \neq \bar{\beta}$.

Proof. The unique sparse step decomposition of $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] cd$ is

$$\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] cd = \left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right] * \left[\begin{smallmatrix} \bullet a \\ \bullet b \end{smallmatrix}\right] * [c\bullet] * [\bullet c] * [d\bullet] * [\bullet d].$$

Thus, α must be as described above. A similar argument applies for β.

Now assume that $x = x'$. Then

$$\gamma = (v \nearrow^{ab} q \searrow_{ab} x = x' \nearrow^{cd} r' \searrow_{cd} z')$$

is a path on X for which $\mathsf{ev}(\gamma) = \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] * \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right]$. Since γ is essential, there are paths $\gamma' \in \mathsf{Path}(X)_{\perp}^{v}$ and $\gamma'' \in \mathsf{Path}(X)_{z'}^{\top}$. The composition $\omega = \gamma'\gamma\gamma''$ is an accepting path. Thus, $\mathsf{ev}(\gamma') * \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] * \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right] * \mathsf{ev}(\gamma'') \in L$: a contradiction.

Calculation of $\mathsf{ev}(\bar{\alpha})$ and $\mathsf{ev}(\bar{\beta})$ is elementary, and $\bar{\alpha} \neq \bar{\beta}$ because $x \neq x'$. □

Proof (of Proposition 43). Let X be an HDA with $\mathsf{L}(X) = L$. We will show that there exist at least 2^n different sparse accepting paths accepting $(abcd)^n$. Let $P = \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] cd$ and $Q = ab \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right]$. For every sequence $\mathbf{R} = (R_1, \dots, R_n) \in \{P, Q\}^n$ let $\omega_{\mathbf{R}}$ be an accepting path such that $\mathsf{ev}(\omega_{\mathbf{R}}) = R_1 * \cdots * R_n$. By Lemma 12, there exist paths $\omega_{\mathbf{R}}^1, \dots, \omega_{\mathbf{R}}^n$ such that $\mathsf{ev}(\omega_{\mathbf{R}}^k) = R_k$ and $\omega'_{\mathbf{R}} = \omega_{\mathbf{R}}^1 * \cdots * \omega_{\mathbf{R}}^n$ is an accepting path. Let $\bar{\omega}_{\mathbf{R}}^k$ be the path defined as in Lemma 44 (*i.e.*, like $\bar{\alpha}$ if $R_k = P$ and $\bar{\beta}$ if $R_k = Q$). Finally, put $\bar{\omega}_{\mathbf{R}} = \bar{\omega}_{\mathbf{R}}^1 * \cdots * \bar{\omega}_{\mathbf{R}}^n$.

Now choose $\mathbf{R} \neq \mathbf{S} \in \{P, Q\}^n$. Assume that $\bar{\omega}_{\mathbf{R}} = \bar{\omega}_{\mathbf{S}}$. This implies that $\bar{\omega}_{\mathbf{R}}^k = \bar{\omega}_{\mathbf{S}}^k$ for all k (all segments have the same length). But there exists k such that $R_k \neq S_k$ (say $R_k = P$ and $S_k = Q$), and, by Lemma 44 again, applied to $\alpha = \bar{\omega}_{\mathbf{R}}^k$ and $\beta = \bar{\omega}_{\mathbf{S}}^k$, we get $\bar{\omega}_{\mathbf{R}}^k \neq \bar{\omega}_{\mathbf{S}}^k$: a contradiction.

As a consequence, the paths $\{\bar{\omega}_{\mathbf{R}}\}_{\mathbf{R} \in \{P,Q\}^n}$ are sparse and pairwise different, and $\mathsf{ev}(\bar{\omega}_{\mathbf{R}}) = (abcd)^n$ for all $\mathbf{R}$. □

7. One-letter languages

In this section we restrict our attention to languages on one letter only, with $\Sigma = \{a\}$. We develop a characterization for deterministic regular one-letter languages using ultimately periodic functions, similar to the one of Büchi [6]. Yet first we show that also in the one-letter case, not all languages are deterministic.

Proposition 45. *The regular language* $(\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right] a + a \left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right])^+$ *is of unbounded ambiguity and hence non-deterministic.*

Proof. A simple modification of the proof of Proposition 43, replacing the letters b, c and d by a. □

Now let X be a deterministic HDA on $\Sigma = \{a\}$. Noting that $\square = \{a^{\parallel k} \mid k \geq 0\}$, we write $X[k]$ for $X[\{a^{\parallel k}\}]$.

We assume that $\perp_X = \{v_0\} \subseteq X_0$ contains only a vertex; the more general case when X also has higher-dimensional start cells is similar and will not be treated below. We also assume that X is *accessible*, that is, every cell is a target of a path starting at v_0.

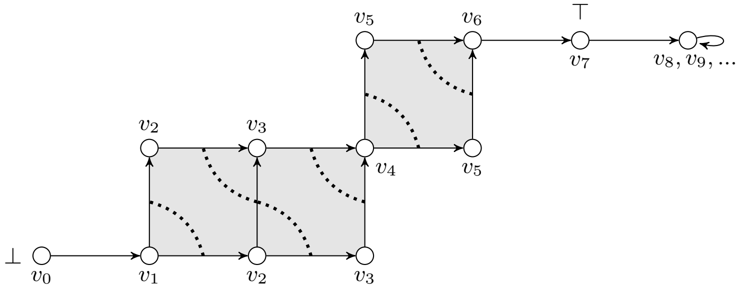


Figure 8: Deterministic HDA on one-letter alphabet (all edges labeled a). Edges connected with dotted lines are identified. Vertex labels indicate the sequence $(v_0, v_1, \dots)$ given by the succ function.

By Lemma 30, $X_{\leq 1}$ is a deterministic finite automaton. Potentially adding an extra sink vertex, we may further assume that $X_{\leq 1}$ is *complete* in the sense that very vertex has an outgoing transition.

Define a function $\mathsf{succ} : X[0] \to X[0]$ by $\mathsf{succ} = \{(v, v') \mid \exists e \in X[1] : \delta_1^0(e) = v, \delta_1^1(e) = v'\}$. This is indeed a function as $X_{\leq 1}$ is deterministic and complete; it is the same as the transition function of $X_{\leq 1}$ seen as a deterministic finite automaton.

Define an infinite sequence $(v_0, v_1, \dots)$ by $v_{n+1} = \mathsf{succ}(v_n)$, then $X[0] = \{v_0, v_1, \dots\}$ by accessibility. The following is easy to see.

Lemma 46. *There exist unique $r_X \geq 1$ and $s_X \geq 0$ such that*

- $v_k = v_{k+r_X}$ *for all* $k \geq s_X$,
- $v_k \neq v_l$ *for all* $k < s_X \leq l$. □

Example 47. Figure 8 shows a simple deterministic HDA. Edges connected with dotted lines are identified, as are their corresponding end points. Also shown is the sequence $(v_0, v_1, \dots)$; in this case, $r = 1$ and $s = 8$.

Now define a function $f_X : \mathbb{N} \to \mathbb{N}$ by

$$f_X(n) = \max\{k \mid \exists x \in X[k] : \delta^0_{\mathsf{ev}(x)}(x) = v_n\}.$$

This function marks the maximal number of concurrent events that can be started in each vertex. That is, $f_X(n) = k$ if there is a k-cell starting at v_n and no higher dimensional cells start at v_n. We collect some properties of f_X.

Lemma 48. *For all $n \geq 0$,*

(1) $f_X(n) \geq 1$,

(2) $f_X(n+1) \geq f_X(n) - 1$, *and*

(3) $f_X(n) = f_X(n + r_X)$ *for* $n \geq s_X$.

Proof. Every v_n has a successor, which implies (1). If $x \in X[f_X(n)]$ is a cell such that $\delta^0_{\mathsf{ev}(x)}(x) = v_n$, then for every $b \in \mathsf{ev}(x)$ we have $\delta^0_{\mathsf{ev}(x)}(\delta^1_b(x)) = v_{n+1}$, which implies (2). Finally, (3) follows from Lemma 46. □

Example 49. For the HDA of Example 47, the values of f_X are

$$(1, 2, 2, 1, 2, 1, 1, 1, 1, \dots).$$

Proposition 50. *Let* $r \geq 1$, $s \geq 0$ *and* $f : \mathbb{N} \to \mathbb{N}$ *be a function satisfying the conditions of Lemma 48. Then there exists a deterministic accessible HDA* X *with a single start vertex such that* $f = f_X$. *Moreover, the underlying precubical set is unique.*

Proof. For $k \geq 0$ define

$$X[k] = \{x^k_n \mid k \leq f(n), n < s + r\},$$
$$\delta^0_A(x^k_n) = x^{k-|A|}_n, \qquad \delta^1_A(x^k_n) = x^{k-|A|}_{n+|A|},$$

with the convention that $x^k_n = x^k_{n-r\lceil (n-s-r+1)/r\rceil}$ for $n \geq s+r$. Let $\bot_X = \{x^0_0\}$.

- Determinism: If $A \subseteq U \in \square$ and $x^k_n \in X[U - A]$, then $k = |U - A|$ and $x^{|U|}_n$ is the only cell such that $\delta^0_A(x^{|U|}_n) = x^k_n$.
- Accessibility: All vertices x^0_n are accessible from x^0_0 and thus all cells x^k_n.
- Uniqueness: $X_{\leq 1}$ is defined by s and r and hence unique. By determinism, for every n there is at most one cell x of a given dimension such that $\delta^0_{\mathsf{ev}(x)}(x) = x^0_n$: uniqueness of X follows. □

Finally we obtain a classification of single-letter languages.

Definition 51. Let S be a set. A function $f : \mathbb{N} \to S$ is *ultimately periodic* if there exists a pair $(r, s) \in \mathbb{N}_{\geq 1} \times \mathbb{N}$, called a *period* of f, such that $f(n+r) = f(n)$ for every $n \geq s$.

If (r, s) is a period of f, then for every $k \geq 0$ and $c \geq 1$, $(cr, s+k)$ is a period as well. Every ultimately periodic function admits a unique minimal period.

Proposition 52. *Let* $\mathsf{TD} = \{(k, T) \mid k \in \mathbb{N}_{\geq 1}, T \subseteq \{0, \dots, k\}\}$. *There is a 1-1 correspondence between*

- *deterministic languages on one letter containing ipomsets with empty start interface,*
- *ultimately periodic functions* $\varphi = (f, \tau) : \mathbb{N} \to \mathsf{TD}$ *such that* $f(n + 1) \geq f(n) - 1$.

Proof. Let L be a deterministic language and X a deterministic HDA with $L = \mathsf{L}(X)$. By Proposition 50, X regarded as a precubical set is determined by an ultimately periodic function $f_X : \mathbb{N} \to \mathbb{N}_{\geq 1}$ such that $f_X(n+1) \geq f_X(n) - 1$. The sets $\tau(n)$ are defined by the condition that $k \in \tau(n)$ iff x_n^k is accepting.

For the converse, let $\varphi : \mathbb{N} \to \mathsf{TD}$ be ultimately periodic and choose a period (r, s). Let X be the HDA defined in Proposition 50, with $\top_X = \{x_n^k \mid k \in \tau(n)\}$, then X is deterministic. □

8. Complement

The *complement* of a language $L \subseteq \mathsf{iiPoms}$, *i.e.*, $\mathsf{iiPoms} - L$, is generally not down-closed and thus not a language. If we define $\overline{L} = (\mathsf{iiPoms} - L){\downarrow}$, then $\overline{L}$ *is* a language, but a *pseudocomplement* rather than a complement: because of down-closure, $L \cap \overline{L} = \emptyset$ is now false in general. The following additional problem poses itself.

Proposition 53. *If L is regular, then $\overline{L}$ has infinite width, hence is not regular.*

Proof. By Lemma 5, $\mathsf{wid}(L)$ is finite. For any $k > \mathsf{wid}(L)$, $\mathsf{iiPoms} - L$ contains all ipomsets of width k, hence $\{\mathsf{wid}(P) \mid P \in \overline{L}\}$ is unbounded. □

To remedy this problem, we introduce a width-bounded version of (pseudo) complement. We fix an integer $k \geq 0$ for the rest of the section.

Definition 54. The *k-bounded complement* of $L \in \mathscr{L}$ is $\overline{L}^k = (\mathsf{iiPoms}_{\leq k} - L){\downarrow}$.

Lemma 55. *Let L and M be languages.*

1. $\overline{L}^0 = \{\mathsf{id}_\emptyset\} - L$.
2. $L \subseteq M$ *implies* $\overline{M}^k \subseteq \overline{L}^k$.
3. $\overline{\overline{L}^k}^k \subseteq L_{\leq k} \subseteq L$.
4. $\overline{L}^k = \overline{L_{\leq k}}^k$

Proof.

1. $\overline{L}^0 = (\mathsf{iiPoms}_{\leq 0} - L){\downarrow} = \{\varepsilon\} - L$.
2. $L \subseteq M$ implies $\mathsf{iiPoms}_k - M \subseteq \mathsf{iiPoms}_k - L$, thus $(\mathsf{iiPoms}_k - M){\downarrow} \subseteq (\mathsf{iiPoms}_k - L){\downarrow}$.
3. We have $\mathsf{iiPoms}_{\leq k} - L_{\leq k} = \mathsf{iiPoms}_{\leq k} - L \subseteq \overline{L}^k$, so by the previous item, $\mathsf{iiPoms}_{\leq k} - \overline{L}^k \subseteq \mathsf{iiPoms}_{\leq k} - (\mathsf{iiPoms}_{\leq k} - L_{\leq k}) = L_{\leq k}$. Thus, $\overline{\overline{L}^k}^k \subseteq L_{\leq k}{\downarrow} = L_{\leq k}$.
4. $\overline{L}^k = (\mathsf{iiPoms}_{\leq k} - L){\downarrow} = (\mathsf{iiPoms}_{\leq k} - L_{\leq k}){\downarrow} = \overline{L_{\leq k}}^k$.

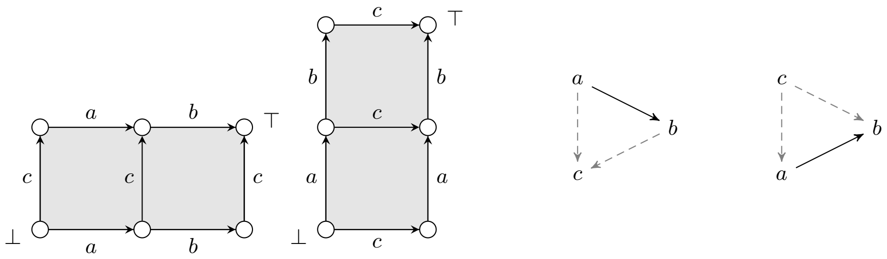


Figure 9: HDA X (disconnected) which accepts language L of Example 58 and the two generating ipomsets in L.

Proposition 56. *For any $k \ge 0$, $\overline{\cdot}^k$ is a pseudocomplement on the lattice $(\mathscr{L}_{\le k}, \supseteq)$, that is, for any $L, M \in \mathscr{L}_{\le k}$, $L \cup M = \mathsf{iiPoms}_k$ iff $\overline{L}^k \subseteq M$.*

The order on the lattice $\mathscr{L}_{\le k}$ is intentionally reversed: $\overline{\cdot}^k$ is a pseudocomplement for $\supseteq$, not for $\subseteq$.

Proof. Let $L, M \in \mathscr{L}_{\le k}$ such that $L \cup M = \mathsf{iiPoms}_k$ and $P \in \overline{L}^k$. There exists $Q \in \mathsf{iiPoms}_{\le k}$ such that $P \sqsubseteq Q$ and $Q \notin L$. Thus, $Q \in M$ and since M is closed by subsumption, $P \in M$.

Conversely, let $L, M \in \mathscr{L}_{\le k}$ such that $\overline{L}^k \subseteq M$ and $P \in \mathsf{iiPoms}_{\le k} - M$. Then $P \in \overline{M}^k$, and we have that $\overline{M}^k \subseteq \overline{\overline{L}^k}^k \subseteq L$ by Lemma 55(3). Thus, $P \in L$ and then $L \cup M = \mathsf{iiPoms}_k$. □

The pseudocomplement property immediately gets us the following.

Corollary 57. *Let $k \ge 0$ and $L, M \in \mathscr{L}_{\le k}$. Then $L \cup \overline{L}^k = \mathsf{iiPoms}_{\le k}$, $\overline{\overline{\overline{L}^k}^k}^k = \overline{L}^k$, $\overline{L \cap M}^k = \overline{L}^k \cup \overline{M}^k$, $\overline{L \cup M}^k \subseteq \overline{L}^k \cap \overline{M}^k$, and $\overline{\overline{L \cup M}^k}^k = \overline{\overline{L}^k}^k \cup \overline{\overline{M}^k}^k$. Further, $\overline{L}^k = \emptyset$ iff $L = \mathsf{iiPoms}_{\le k}$.*

For $k = 0$ and $k = 1$, $\overline{\cdot}^k$ is a complement on $\mathsf{iiPoms}_{\le k}$, but for $k \ge 2$ it is not: in general, neither $L = \overline{\overline{L}^k}^k$, $L \cap \overline{L}^k = \emptyset$, nor $\overline{L \cup M}^k = \overline{L}^k \cap \overline{M}^k$ hold:

Example 58. Let $A = \{P \in \mathsf{iiPoms}_{\le 2} \mid abc \sqsubseteq P\}$, $L = \{\bigl[\begin{smallmatrix} a \to b \\ c \end{smallmatrix}\bigr], \bigl[\begin{smallmatrix} c \\ a \to b \end{smallmatrix}\bigr]\}{\downarrow}$ and $M = (A - L){\downarrow}$. The HDA X in Figure 9 accepts L. Notice that due to the non-commutativity of parallel composition (because of event order), X consists of two parts, one a "transposition" of the other. The left part accepts $\bigl[\begin{smallmatrix} a \to b \\ c \end{smallmatrix}\bigr]$, while the right part accepts $\bigl[\begin{smallmatrix} c \\ a \to b \end{smallmatrix}\bigr]$.

Now $abc \sqsubseteq \bigl[\begin{smallmatrix} a \to c \\ b \end{smallmatrix}\bigr]$ which is not in L, so that $abc \in \overline{L}^2$. Similarly, $abc \sqsubseteq \bigl[\begin{smallmatrix} a \to b \\ c \end{smallmatrix}\bigr] \notin M$, so $abc \in \overline{M}^2$. Thus, $abc \in \overline{L}^2 \cap \overline{M}^2$. On the other hand, for any P such that $\mathsf{wid}(P) \le 2$ and $abc \sqsubseteq P$, we have $P \in L \cup M = A{\downarrow}$. Hence $abc \notin \overline{L \cup M}^2$.

Finally, $\overline{L}^3$ contains every ipomset of width 3, hence $\overline{L}^3 = \mathsf{iiPoms}_{\le 3}$, so that $L \cap \overline{L}^3 = L \neq \emptyset$ and $\overline{\overline{L}^3}^3 = \emptyset \neq L$. This may be generalised to the fact that $\overline{\overline{L}^k}^k = \emptyset$ as soon as $\mathsf{wid}(L) < k$.

For $k \geq 0$, let $\mathscr{S}_k$ be the set of all languages L for which $L = \overline{\overline{L}^k}^k$. We characterise $\mathscr{S}_k$ in the following. By $\overline{\overline{\overline{L}^k}^k}^k = \overline{L}^k$ (Cor. 57), $\mathscr{S}_k = \{\overline{L}^k \mid L \in \mathscr{L}\}$, *i.e.*, $\mathscr{S}_k$ is the image of $\mathscr{L}$ under $\overline{}^k$. (This is a general property of pseudocomplements.)

Define $\mathsf{M}_k = \{P \in \mathsf{iiPoms}_{\leq k} \mid \forall Q \in \mathsf{iiPoms}_{\leq k} : Q \neq P \implies P \not\sqsubseteq Q\}$, the set of all $\sqsubseteq$-maximal elements of $\mathsf{iiPoms}_{\leq k}$. In particular, $\mathsf{M}_k{\downarrow} = \mathsf{iiPoms}_{\leq k}$. Note that $P \in \mathsf{M}_k$ does not imply $\mathsf{wid}(P) = k$: for example, $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] \in \mathsf{M}_3$.

Lemma 59. *For any $L \in \mathscr{L}$, $\overline{L}^k = (\mathsf{M}_k - L){\downarrow}$.*

Proof. We have

$$\begin{aligned} Q \in \overline{L}^k &\iff \exists P \in (\mathsf{iiPoms}_{\leq k} - L) :\ Q \sqsubseteq P \\ &\iff \exists P \in (\mathsf{iiPoms}_{\leq k} - L) \cap \mathsf{M}_k :\ Q \sqsubseteq P \\ &\iff \exists P \in \mathsf{M}_k - L :\ Q \sqsubseteq P \iff Q \in (\mathsf{M}_k - L){\downarrow}. \end{aligned}$$

□

Corollary 60. *Let $L \in \mathscr{L}$ and $k \geq 0$, then $\overline{L}^k = \mathsf{iiPoms}_{\leq k}$ iff $L \cap \mathsf{M}_k = \emptyset$.*

Proposition 61. *$\mathscr{S}_k = \{A{\downarrow} \mid A \subseteq \mathsf{M}_k\}$.*

Proof. Inclusion $\subseteq$ follows from Lemma 59. For the other direction, $A \subseteq \mathsf{M}_k$ implies

$$\overline{\overline{A{\downarrow}}^k}^k = \overline{(\mathsf{M}_k - A{\downarrow}){\downarrow}}^k = \overline{(\mathsf{M}_k - A){\downarrow}}^k = (\mathsf{M}_k - (\mathsf{M}_k - A)){\downarrow} = A{\downarrow}.$$

If $A \neq B \subseteq \mathsf{M}_k$, then also $A{\downarrow} \neq B{\downarrow}$, since all elements of M_k are $\sqsubseteq$-maximal. As a consequence, $\mathscr{S}_k$ and the powerset $\mathcal{P}(\mathsf{M}_k)$ are isomorphic lattices, hence $\mathscr{S}_k$ is a distributive lattice with join $L \vee M = L \cup M$ and meet $L \wedge M = (L \cap M \cap \mathsf{M}_k){\downarrow}$.

Corollary 62. *For $L, M \in \mathscr{L}$, $\overline{L}^k = \overline{M}^k$ iff $L \cap \mathsf{M}_k = M \cap \mathsf{M}_k$.*

We can now show that bounded complement preserves regularity.

Theorem 63. *If $L \in \mathscr{L}$ is regular, then for all $k \geq 0$ so is $\overline{L}^k$.*

Proof. By Proposition 18, $L_{\leq k}$ is regular. Let X be an HDA such that $\mathsf{L}(X) = L_{\leq k}$ and $k = \dim(X)$. The $\Omega_{\leq k}$-language $\mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* \cap \Phi(\mathsf{L}(X))$ is regular by Lemma 26, hence so is $\mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* - \Phi(\mathsf{L}(X))$. By Lemma 25, Ψ preserves regularity, so $\Psi(\mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* - \Phi(\mathsf{L}(X)))$ is a regular ipomset language. Now for $P \in \mathsf{iiPoms}_{\leq k}$ we have

$$\begin{aligned} &P \in \Psi(\mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* - \Phi(L_{\leq k})) \\ &\quad\iff \exists Q \sqsupseteq P, \exists Q_1 \cdots Q_n \in \mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* - \Phi(L_{\leq k}) : Q = Q_1 * \cdots * Q_n \\ &\quad\iff \exists Q_1 \cdots Q_n \in \mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* : P \sqsubseteq Q_1 * \cdots * Q_n \notin L_{\leq k} \\ &\quad\iff P \in \overline{L_{\leq k}}^k, \end{aligned}$$

hence $\overline{L_{\leq k}}^k = \Psi(\mathsf{Id}_{\leq k}\,(\Omega_{\leq k}\mathsf{Id}_{\leq k})^* - \Phi(L_{\leq k}))$. Lemma 55(4) allows us to conclude. □

Corollary 64. *$\mathsf{iiPoms}_{\leq k}$ is regular for every $k \geq 0$.*

9. Conclusion and further work

We have advanced the theory of higher-dimensional automata (HDAs) along several lines: we have shown a pumping lemma, exposed a regular language of unbounded ambiguity, introduced width-bounded complement, shown that regular languages are closed under intersection and width-bounded complement, and shown that determinism of a regular language and inclusion of regular languages are decidable.

A question which is still open is if it is decidable whether a regular language is of bounded ambiguity. On a more general level, a notion of recognizability is still missing. This is complicated by the fact that ipomsets do not form a monoid but rather a 2-category with lax tensor [11].

Even more generally, a theory of weighted and/or timed HDAs would be called for, with a corresponding Kleene-Schützenberger theorem. For timed HDAs, some work is available in [1, 8]. For weighted HDAs, the convolution algebras of [12] may provide a useful framework. [2] shows a close relation between ipomsets and non-free category generated by starters and terminators, and we are wondering whether this could provide a path both to a treatment of recognizability and to weighted HDAs.

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, *PETRI NETS*, volume 14628 of *Lecture Notes in Computer Science*, pages 197–219. Springer, 2024.

[2] Amazigh Amrane, Hugo Bazille, Emily Clement, Uli Fahrenberg, and Krzysztof Ziemiański. Presenting interval pomsets with interfaces. In *RAMiCS*, 2024. Accepted.

[3] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Marie Fortin. Logic and languages of higher-dimensional automata. In *DLT*, 2024. Accepted.

[4] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. In Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa, editors, *ICTAC*, volume 14446 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2023.

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

[6] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6(1-6):66–92, 1960.

[7] 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.

[8] Uli Fahrenberg. Higher-dimensional timed and hybrid automata. *Leibniz Transactions on Embedded Systems*, 8(2):03:1–03:16, 2022.

[9] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021. `https://arxiv.org/abs/2103.07557`.

[10] 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 *Leibniz International Proceedings in Informatics*, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `https://doi.org/10.4230/LIPIcs.CONCUR.2022.29`.

[11] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285(B):104914, 2022. `https://arxiv.org/abs/2106.10895`.

[12] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Catoids and modal convolution algebras. *Algebra Universalis*, 84(10), 2023.

[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, 2023. `https://arxiv.org/abs/2210.08298`.

[14] Jean Fanchon and Rémi Morin. Regular sets of pomsets with autoconcurrency. In Luboš Brim, Petr Jančar, Mojmír Křetínský, and Antonín Kučera, editors, *CONCUR*, volume 2421 of *Lecture Notes in Computer Science*, pages 402–417. Springer, 2002.

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

[16] Eric Goubault. Labelled cubical sets and asynchronous transition systems: an adjunction. In *CMCIM*, 2002. `http://www.lix.polytechnique.fr/~goubault/papers/cmcim02.ps.gz`.

[17] Jan Grabowski. On partial languages. *Fundamentae Informatica*, 4(2):427, 1981.

[18] Ryszard Janicki and Maciej Koutny. Operational semantics, interval orders and sequences of antichains. *Fundamentae Informatica*, 169(1-2):31–55, 2019.

[19] Peter Jipsen and M. Andrew Moshier. Concurrent Kleene algebra with tests and branching automata. *Journal of Logic and Algebraic Methods in Programming*, 85(4):637–652, 2016.

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

[21] Vaughan R. Pratt. Modeling concurrency with geometry. In *POPL*, pages 311–322, New York City, 1991. ACM Press.

[22] Mike W. Shields. Concurrent machines. *The Computer Journal*, 28(5):449–465, 1985.

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

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

[25] Rob J. van Glabbeek. Erratum to "On the expressiveness of higher dimensional automata". *Theoretical Computer Science*, 368(1-2):168–194, 2006.

[26] Norbert Wiener. A contribution to the theory of relative position. *Proceedings of the Cambridge Philosophical Society*, 17:441–449, 1914.