Closure and Decision Properties for Higher-Dimensional Automata

Amazigh Amrane[1], Hugo Bazille[1], Uli Fahrenberg[1], and Krzysztof Ziemiański[2]

[1] EPITA Research Laboratory (LRE), Paris, France
[2] University of Warsaw, Poland

Abstract. In this paper we develop the language theory of higher-dimensional automata (HDAs). Regular languages of HDAs are sets of finite interval partially ordered multisets (pomsets) with interfaces (iiPoms). We first show a pumping lemma which allows us to expose a class of non-regular languages. We also give an example of a regular language with unbounded ambiguity. 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. On the other hand, complements of regular languages are not regular. We introduce a width-bounded complement and show that width-bounded complements of regular languages are again regular.

1 Introduction

Higher-dimensional automata (HDAs), introduced by Pratt and van Glabbeek [16, 18], are a general geometric model for non-interleaving concurrency which subsumes, for example, event structures and Petri nets [19]. HDAs of dimension one are standard automata, whereas HDAs of dimension two are isomorphic to asynchronous transition systems [2, 11, 17]. As an example, Fig. 1 shows Petri net and HDA models for a system with two events, labelled a and b. The Petri 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 [4], which are sets of partially ordered multisets with interfaces (ipomsets) [6] that are closed under subsumptions. Follow-up papers introduce a language theory for HDAs, showing a Kleene

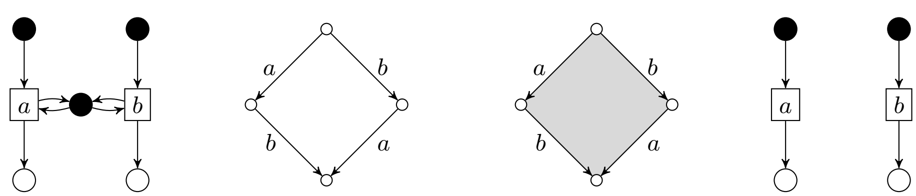


Fig. 1: Petri net and HDA models distinguishing interleaving (left) from non-interleaving (right) concurrency. Left: models for $a.b{+}b.a$; right: models for $a \parallel b$.

theorem [5], which makes a connection between rational and regular ipomset languages (those accepted by finite HDAs), and a Myhill-Nerode theorem [8] 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 Sect. 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.

The paper [8] introduces deterministic HDAs and shows that not all HDAs are determinizable. As a weaker notion in-between determinism and non-determinism, one may ask whether all regular languages may be recognized by finitely ambiguous HDAs, *i.e.*, HDAs in which there is an upper bound for the number of accepting paths on any ipomset. We show that the answer to this question is negative and that there are regular languages of unbounded ambiguity.

In Sect. 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, and we leave open the question if, and in what sense, it would be invertible. Nevertheless, this translation 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.

Finally, in Sect. 6, 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; its skeletal elements (the ones for which double complement is identity) have an easy characterisation; 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 [9, 14, 15].

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 [5, 8]. Categorical reasoning would have simplified proofs in several places, and we do make note of this in several footnotes, 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 (Lem. 11);
- regular languages of unbounded ambiguity (Prop. 16);
- closure of regular languages under intersection (Prop. 15);
- closure of regular languages under width-bounded complement (Thm. 33);
- decidability of inclusion of regular languages (Thm. 22).

Due to space constraints, some proofs had to be omitted from this paper. These can be found in the long version [1].

2 Pomsets with interfaces

HDAs model systems in which (labelled) 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 which events may be active already at the beginning or remain active at the end.

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 labelled 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),[3] and a labelling $\lambda_U : U \to \Sigma$.
- 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 labelling $\lambda_P : P \to \Sigma$, 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.

[3] 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".

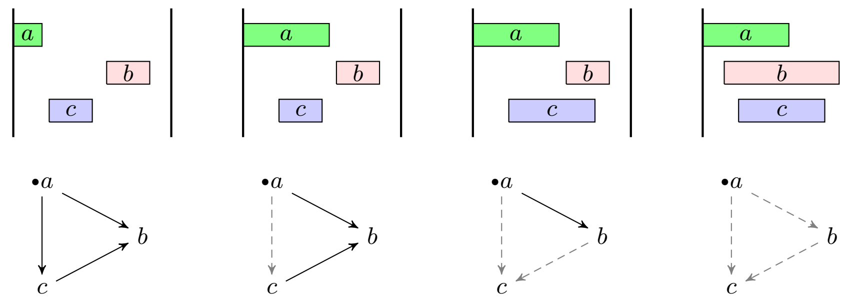


Fig. 2: Activity intervals of events (top) and corresponding ipomsets (bottom), *cf.* Ex. 1. Full arrows indicate precedence order; dashed arrows indicate event order; bullets indicate interfaces.

– 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.

Conclists may be regarded as pomsets with empty precedence (*discrete* pomsets); the last condition above enforces that $\dashrightarrow$ is then total. Pomsets are ipomsets with empty interfaces, and in any ipomset P, the substructures 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.

An ipomset P is *interval* if $<_P$ is an interval order [10], 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 [21]. 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

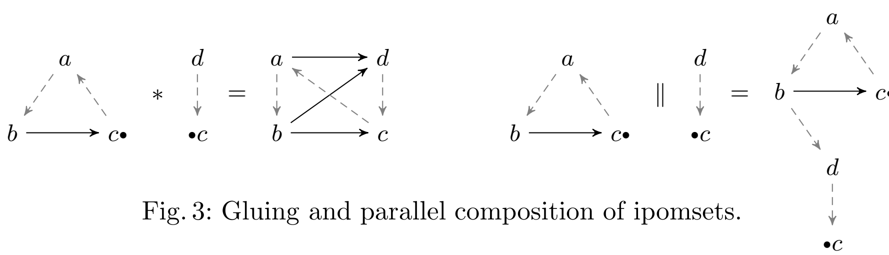


Fig. 3: Gluing and parallel composition of ipomsets.

non-essential event order.) This definition adapts the one of [12] to event orders and interfaces. Intuitively, P has more order and less concurrency than Q.

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

$$\bullet acb \sqsubseteq \begin{bmatrix} \bullet a \searrow \\ \qquad b \\ c \nearrow \end{bmatrix} \sqsubseteq \begin{bmatrix} \bullet a \to b \\ c \end{bmatrix} \sqsubseteq \begin{bmatrix} \bullet a \\ b \\ c \end{bmatrix}$$

An event e_1 is smaller than e_2 in the precedence order if e_1 is terminated before e_2 is started; 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.

Compositions. The standard serial and parallel compositions of pomsets [12] 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

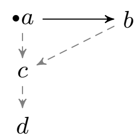
– $x < y$ if $x <_P y$ or $x <_Q y$;
– $x \dashrightarrow y$ if $x \dashrightarrow_P y$, $x \dashrightarrow_Q y$, or $x \in P$ and $y \in Q$;
– $S = S_P \cup S_Q$ and $T = T_P \cup T_Q$;
– $\lambda(x) = \lambda_P(x)$ if $x \in P$ and $\lambda(x) = \lambda_Q(x)$ if $x \in Q$.

Note that parallel composition of ipomsets is generally not commutative, see [6] or Ex. 28 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 $T_P = S_Q$, $x \dashrightarrow_P y$ iff $x \dashrightarrow_Q y$ for all $x, y \in T_P = S_Q$, and the restrictions $\lambda_{P \upharpoonright T_P} = \lambda_{Q \upharpoonright S_Q}$, 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 $y \in Q - S_Q$;[4]

[4] We use "–" for set difference instead of the perhaps more common "\".



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

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

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

– $\dashrightarrow$ 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 $x \in Q$.

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. Fig 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 $\sqsubseteq$.

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 ${}_A{\uparrow}U$ and $U{\downarrow}_B$ 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 $\mathsf{id}_U = {}_\emptyset{\uparrow}U = U{\downarrow}_\emptyset$.

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 [6], see also [13]. 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*.

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 ([8]). *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 $2\,\mathsf{size}(P)$.*

Rational languages. For $A \subseteq \mathsf{iiPoms}$ we let

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

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

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

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

Lemma 5 ([5]). *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 [8].

Theorem 6 ([8]). *A language $L \in \mathscr{L}$ is rational iff $\mathsf{suff}(L)$ is finite, iff $\mathsf{pref}(L)$ 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

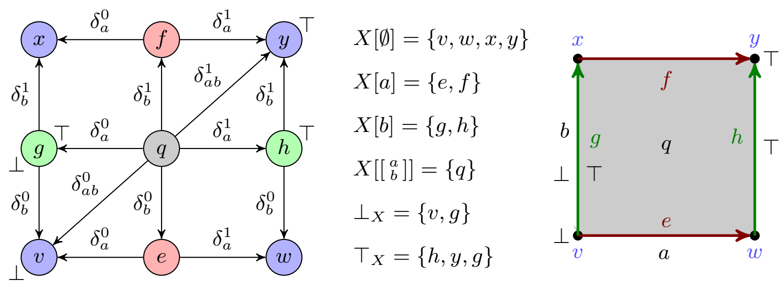


Fig. 5: A two-dimensional HDA X on $\Sigma = \{a, b\}$, see Ex. 7.

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*

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

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 $\delta^0_A, \delta^1_A : X[U] \to X[U - A]$ 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 $\dim(X) = \sup\{|\mathsf{ev}(x)| \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$.[5]

A standard automaton is the same as a one-dimensional HDA X 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-labelled transitions, and face maps $\delta^0_{\{a\}}$ and $\delta^1_{\{a\}}$ attach source and target states to transitions. In contrast to ordinary automata we allow start and accept *transitions* instead of merely states, so languages of one-dimensional HDAs may contain words with interfaces.

Example 7. 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 $X_0 = \{x, y, v, w\}$ in which no event is active (for all $z \in X_0$, $\mathsf{ev}(z) = \emptyset$), the transition cells $X_1 = \{g, h, f, e\}$ in which one event is active ($\mathsf{ev}(f) = \mathsf{ev}(e) = a$ and $\mathsf{ev}(g) = \mathsf{ev}(h) = b$), and the square cell q where $\mathsf{ev}(q) = [\,^a_b\,]$.

[5] Precubical sets are presheaves over a category on objects $\square$, and then HDAs form a category with the induced morphisms, see [5].

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.

Regular languages. *Computations* of HDAs are *paths*, *i.e.*, sequences $\alpha = (x_0, \phi_1, x_1, \dots, x_{n-1}, \phi_n, x_n)$ consisting of cells x_i of X and symbols ϕ_i which indicate face map types: for every $i \in \{1, \dots, n\}$, (x_{i-1}, ϕ_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, and $\nearrow^\emptyset = \searrow_\emptyset$.

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 by $\mathsf{Path}(X)^Z_Y$. 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 $\alpha = \alpha_1 * \dots * \alpha_n$ is a concatenation, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \dots * \mathsf{ev}(\alpha_n)$.

Note that upsteps in α correspond to starters in $\mathsf{ev}(\alpha)$ and downsteps correspond to terminators. Path equivalence $\alpha \simeq \beta$ implies $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$ [5]. Further, if $\alpha = \alpha_1 * \dots * \alpha_n$ is a sparse path, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \dots * \mathsf{ev}(\alpha_n)$ is a sparse step decomposition.

The *language* of an HDA X is $\mathsf{L}(X) = \{\mathsf{ev}(\alpha) \mid \alpha \text{ accepting path in } X\}$.[6]

Example 8. The HDA X of Fig. 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

[6] Every ipomset P may be converted into a *track object* $\square^P$, see [5], which is an HDA with the property that for any HDA X, $P \in \mathsf{L}(X)$ iff there is a morphism $\square^P \to X$.

is $[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}]$. 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 $\mathsf{L}(X) = \{b\bullet, \bullet b\bullet, [\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}], [\begin{smallmatrix} a \\ \bullet b\bullet \end{smallmatrix}], [\begin{smallmatrix} a \\ b \end{smallmatrix}], [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}]\}{\downarrow}$.

Lemma 9. *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.*

Languages of HDAs are sets of (interval) ipomsets which are closed under subsumption [5], *i.e.*, languages in our sense. A language is *regular* if it is the language of a finite HDA.

Theorem 10 ([5]). *A language is regular iff it is rational.*

4 Regular and non-regular languages

Pumping lemma. The next lemma is similar to the pumping lemma for word languages.

Lemma 11. *Let L be a regular language. There exists $k \in \mathbb{N}$ such that for any $P \in L$, any decomposition $P = Q_1 * \cdots * Q_n$ 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 $Q_1 * \cdots * Q_i * (Q_{i+1} * \cdots * Q_j)^+ * Q_{j+1} * \cdots * Q_n \subseteq L$.*

Proof. Let X be an HDA accepting L and $k > |X|$. By Lem. 9 there exists an accepting path $\alpha = \alpha_1 * \cdots * \alpha_n$ 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 $r \geq 1$

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

is an accepting path that recognises $Q_1 * \cdots * Q_i * (Q_{i+1} * \cdots * Q_j)^r * Q_{j+1} * \cdots * Q_n$. ☐

Corollary 12. *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 Lem. 11 to a dense step decomposition $P = Q_1 * \cdots * Q_{2\,\mathsf{size}(P)}$, *cf.* Lem. 4. We may now expose a language which is not regular.

Proposition 13. *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 (Proof of Prop. 13). We give two proofs. The first uses Thm. 6: for every $k \geq 1$, $\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]^k \backslash L = \{\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]^n * a^{n+k} \mid n \geq 0\}{\downarrow}$, and these are different for different k, so $\mathsf{suff}(L)$ is infinite.

The second proof uses Lem. 11. Assume L to be regular, let k be the constant from the lemma, and take $P = \left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]^k * a^k = Q_1 * \cdots * Q_k * Q_{k+1}$, where $Q_1 = \cdots = Q_k = \left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]$ and $Q_{k+1} = a^k$. For $m = 0$ we obtain that $\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]^{k+(j-i)r} a^k \in L$ 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

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

The idea is to remove from the right-hand part of the expression 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-restriction of any regular language remains regular:

Proposition 14. *Let $k \geq 0$. If $L \in \mathscr{L}$ is regular, then so is $L_{\leq k}$.*

Intersection. 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 15. *The regular languages are closed under $\cap$.*

Proof. We again give two proofs, one algebraic using Thm. 6 and another, constructive proof using Thm. 10. 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

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

which is thus finite.

For the second, constructive proof, let X_1 and X_2 be HDAs. We construct an HDA X with $\mathsf{L}(X) = \mathsf{L}(X_1) \cap \mathsf{L}(X_2)$:[7]

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

[7] This is the product in the category of HDAs. Using track objects, the lemma follows immediately.

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.* Lem. 9.

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 $\mathsf{ev}(\alpha) = P$. □

Ambiguity. It is shown in [8] that not all languages are determinizable, that is, there exist regular languages which cannot be recognised by deterministic HDAs. We have not introduced deterministic HDAs here and will not need them in what follows, instead we prove a strengthening of that result. Say that an HDA X is *k-ambiguous*, for $k \geq 1$, if every $P \in \mathsf{L}(X)$ is the event ipomset of at most k sparse accepting paths in X. (Deterministic HDAs are 1-ambiguous.) A language L is said to be *of bounded ambiguity* if it is recognised by a k-ambiguous HDA for some k.

Proposition 16. *The regular language $L = ([\begin{smallmatrix} a \\ b \end{smallmatrix}]\, cd + ab\, [\begin{smallmatrix} c \\ d \end{smallmatrix}])^+$ is of unbounded ambiguity.*

5 ST-automata

We define a variant of a construction from [5] which translates HDAs into finite automata over an alphabet of starters and terminators. This will be useful for showing properties of HDA languages. Let $\Omega = \{{}_A{\uparrow}U, U{\downarrow}_A \mid U \in \square, A \subseteq U\}$ be the (infinite) set of starters and terminators over Σ and, for any $k \geq 0$, $\Omega_{\leq k} = \Omega \cap \mathsf{iiPoms}_{\leq k}$. Note that the sets $\Omega_{\leq k}$ are all finite.

Let X be an HDA and $k \geq \dim(X)$. The *ST_k-automaton* pertaining to X is the finite automaton $G_k(X) = (\Omega_{\leq k}, Q, I, F, E)$ with $Q = X \cup \{x_\bot \mid x \in \bot_X\}$, $I = \{x_\bot \mid x \in \bot_X\}$, $F = \top_X$, and

$$E = \{(\delta^0_A(x), {}_A{\uparrow}U, x) \mid x \in X[U], A \subseteq U\} \cup \{(x_\bot, \mathsf{id}_U, x) \mid x \in \bot_X \cap X[U]\} \\ \cup \{(x, U{\downarrow}_A, \delta^1_A(x)) \mid x \in X[U], A \subseteq U\}.$$

We add extra copies of start cells in $G_k(X)$ in order to avoid runs on the empty word ϵ. Note that only the alphabet of $G_k(X)$ changes for different k.

In what follows, we consider languages of nonempty words over Ω, which we denote by W etc. and the class of such languages by $\mathscr{W}$. Further, $\mathsf{W}(\mathcal{A})$ denotes the set of words accepted by a finite automaton $\mathcal{A}$.

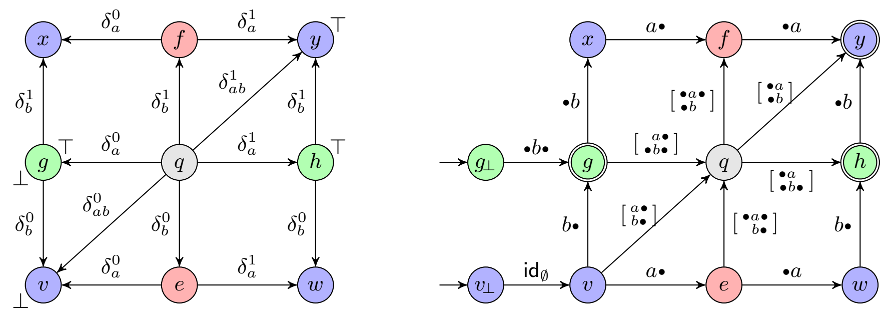


Fig. 6: HDA of Fig. 5 and its ST-automaton (identity loops not displayed).

Example 17. Figure 6 displays the ST-automaton $G_2(X)$ pertaining to the HDA X in Fig. 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. Further, this example illustrates the necessity to duplicate initial states: without that, the empty word would be accepted by $G_2(X)$, while the empty ipomset is *not* in $\mathsf{L}(X)$ (see Ex. 8). We have $\mathsf{W}(G_2(X)) = \{\mathsf{id}_\emptyset b\bullet, \bullet b\bullet, \mathsf{id}_\emptyset \left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right]\left[\begin{smallmatrix} \bullet a \\ \bullet b\bullet \end{smallmatrix}\right], \bullet b\bullet \left[\begin{smallmatrix} a\bullet \\ \bullet b\bullet \end{smallmatrix}\right]\left[\begin{smallmatrix} \bullet a \\ \bullet b\bullet \end{smallmatrix}\right], \dots\}$.

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

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

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 18. 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 $W = \{\left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right]\left[\begin{smallmatrix} \bullet a \\ \bullet b \end{smallmatrix}\right]\}$, then $\Psi(W) = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], ab, ba\}$ and $\Phi(\Psi(W)) \not\subseteq W$.

Lemma 19. *Φ respects boolean operations: for all $L_1, L_2 \in \mathscr{L}$, $\Phi(L_1 \cap L_2) = \Phi(L_1) \cap \Phi(L_2)$ and $\Phi(L_1 \cup L_2) = \Phi(L_1) \cup \Phi(L_2)$. Ψ respects regular operations: 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 $\Psi(W_1^+) = \Psi(W_1)^+$.*

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

Let $\mathsf{Id} = \{\mathsf{id}_U \mid U \in \square\} \subseteq \Omega$ and, for any $k \geq 0$, $\mathsf{Id}_{\leq k} = \mathsf{Id} \cap \mathsf{iiPoms}_{\leq k} \subseteq \Omega_k$. Then $\mathsf{Id}_{\leq k}\, \Omega^*_{\leq k} \subseteq \Omega^*_{\leq k}$ (which is a regular word language) denotes the set of all words starting with an identity.

Lemma 20. *For any HDA X and $k \geq \dim(X)$, $\mathsf{W}(G_k(X)) = \Phi(\mathsf{L}(X)) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k}$.*

Proof. There is a one-to-one correspondence between the accepting paths in X and $G_k(X)$:

$$\alpha = (x_0, \phi_1, x_1, \phi_2, \dots, \phi_n, x_n) \mapsto \big((x_0)_\perp \to x_0 \xrightarrow{\psi_1} x_1 \xrightarrow{\psi_2} \cdots \xrightarrow{\psi_n} x_n\big) = \omega$$

where ψ_i is the starter or terminator corresponding to the step ϕ_i. If $P_0 P_1 \cdots P_n \in \mathsf{W}(G_k(X))$, then there is an accepting path ω such that $P_0 = \mathsf{id}_{\mathsf{ev}(x_0)}$ and $P_i = \mathsf{ev}(x_{i-1}, \varphi_i, x_i)$. The corresponding path α in X is accepting. Hence $P_0 * P_1 * \cdots * P_n = P_1 * \cdots * P_n = \mathsf{ev}(\alpha) \in \mathsf{L}(X)$, and $P_0 P_1 \cdots P_n \in \Phi(\mathsf{L}(X))$. Further, P_0 is an identity, which shows the inclusion $\subseteq$.

Now let $P_0 P_1 \cdots P_n \in \Phi(\mathsf{L}(X)) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k}$. Thus P_0 is an identity and $P_0 * P_1 * \cdots * P_n \in \mathsf{L}(X)$. Using Lem. 9 we conclude that the exists an accepting path $\alpha = \beta_1 * \cdots * \beta_n$ in X such that $\mathsf{ev}(\beta_i) = P_i$. The path ω corresponding to α recognises $P_0 P_1 \cdots P_n$, which shows the inclusion $\supseteq$. ☐

Lemma 21. *Let $k \geq 0$. For all $L_1, L_2 \in \mathscr{L}_{\leq k}$, $L_1 \subseteq L_2$ iff $\Phi(L_1) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k} \subseteq \Phi(L_2) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k}$.*

Proof. The forward implication is immediate from Lem. 19. Now if $L_1 \not\subseteq L_2$, then also $\Phi(L_1) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k} \not\subseteq \Phi(L_2) \cap \mathsf{Id}_{\leq k}\, \Omega^*_{\leq k}$, since every ipomset admits a step decomposition starting with an identity. ☐

Theorem 22. *Inclusion of regular languages is decidable.*

Proof. Let L_1 and L_2 be regular and recognised respectively by X_1 and X_2, and let $k = \max(\dim(X_1), \dim(X_2))$. By Lemmas 20 and 21,

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

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

6 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.

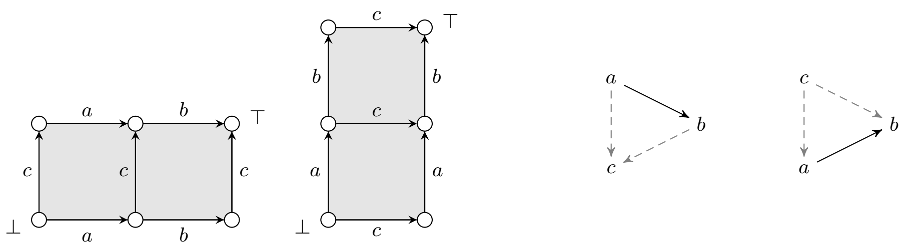


Fig. 7: HDA X which accepts language L of Ex. 28 and the two generating ipomsets in L.

Proposition 23. *If L is regular, then $\overline{L}$ has infinite width, hence is not regular.*

Proof. By Lem. 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 24. *The* k-bounded complement *of $L \in \mathscr{L}$ is $\overline{L}^k = (\mathsf{iiPoms}_{\leq k} - L)\downarrow$.*

Lemma 25. *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. $\overline{L}^k = \overline{L_{\leq k}}^k$

Proposition 26. *For any $k \geq 0$, $\overline{\cdot}^k$ is a pseudocomplement on the lattice $(\mathscr{L}_{\leq k}, \supseteq)$, that is, for any $L, M \in \mathscr{L}_{\leq k}$, $L \cup M = \mathsf{iiPoms}_k$ iff $\overline{L}^k \subseteq M$.*

Proof. Let $L, M \in \mathscr{L}_{\leq k}$ such that $L \cup M = \mathsf{iiPoms}_k$ and $P \in \overline{L}^k$. There exists $Q \in \mathsf{iiPoms}_{\leq k}$ 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 $L, M \in \mathscr{L}_{\leq k}$ such that $\overline{L}^k \subseteq M$ and $P \in \mathsf{iiPoms}_{\leq k} - M$. Then $P \in \overline{M}^k$, and we have that $\overline{M}^k \subseteq \overline{\overline{L}^k}^k \subseteq L$ by Lem. 25(3). Thus, $P \in L$ and then $L \cup M = \mathsf{iiPoms}_k$. □

The pseudocomplement property immediately gets us the following.

Corollary 27. *Let $k \geq 0$ and $L, M \in \mathscr{L}_{\leq k}$. Then $L \cup \overline{L}^k = \mathsf{iiPoms}_{\leq k}$, $\overline{\overline{\overline{L}^k}^k}^k = \overline{L}^k$, $\overline{L \cap M}^k = \overline{L}^k \cup \overline{M}^k$, $\overline{L \cup M}^k \subseteq \overline{L}^k \cap \overline{M}^k$, 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 $L = \mathsf{iiPoms}_{\leq k}$.*

For $k = 0$ and $k = 1$, $\overline{}^k$ is a complement on $\mathsf{iiPoms}_{\leq k}$, but for $k \geq 2$ 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 28. Let $A = \{P \in \mathsf{iiPoms}_{\leq 2} \mid abc \sqsubseteq P\}$, $L = \{[\begin{smallmatrix} a \to b \\ c \end{smallmatrix}], [\begin{smallmatrix} c \\ a \to b \end{smallmatrix}]\}\downarrow$ and $M = (A - L)\downarrow$. The HDA X in Fig. 7 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 $[\begin{smallmatrix} a \to b \\ c \end{smallmatrix}]$, while the right part accepts $[\begin{smallmatrix} c \\ a \to b \end{smallmatrix}]$.

Now $abc \sqsubseteq [\begin{smallmatrix} a \to c \\ b \end{smallmatrix}]$ which is not in L, so that $abc \in \overline{L}^2$. Similarly, $abc \sqsubseteq [\begin{smallmatrix} a \to b \\ c \end{smallmatrix}] \notin M$, 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 $\mathsf{wid}(P) \leq 2$ 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 $\overline{L}^3 = \mathsf{iiPoms}_{\leq 3}$, 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 $\mathsf{wid}(L) < k$.

We say that $L \in \mathscr{L}$ is *k-skeletal* if $L = \overline{\overline{L}^k}^k$. Let $\mathscr{S}_k$ be the set of all k-skeletal languages. We characterise $\mathscr{S}_k$ in the following. By $\overline{\overline{\overline{L}^k}^k}^k = \overline{L}^k$ (Cor. 27), $\mathscr{S}_k = \{\overline{L}^k \mid L \in \mathscr{L}\}$, *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, $[\begin{smallmatrix} a \\ b \end{smallmatrix}] \in \mathsf{M}_3$.

Lemma 29. *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 30. *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 31. $\mathscr{S}_k = \{A\downarrow \mid A \subseteq \mathsf{M}_k\}$.

Proof. Inclusion $\subseteq$ follows from Lem. 29. 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 32. *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 33. *If $L \in \mathscr{L}$ is regular, then for all $k \geq 0$ so is $\overline{L}^k$.*

Proof. By Prop. 14, $L_{\le k}$ is regular. Let X be an HDA such that $\mathsf{L}(X) = L_{\le k}$ and $k = \dim(X)$. The $\Omega_{\le k}$-language $\mathsf{Id}_{\le k}\, \Omega^*_{\le k} \cap \Phi(\mathsf{L}(X))$ is regular by Lem. 20, hence so is $\mathsf{Id}_{\le k}\, \Omega^*_{\le k} - \Phi(\mathsf{L}(X))$. Ψ preserves regularity, so $\Psi(\mathsf{Id}_{\le k}\, \Omega^*_{\le k} - \Phi(\mathsf{L}(X)))$ is a regular ipomset language. Now for $P \in \mathsf{iiPoms}_{\le k}$ we have

$$\begin{aligned} &P \in \Psi(\mathsf{Id}_{\le k}\, \Omega^*_{\le k} - \Phi(L_{\le k})) \\ &\quad\iff \exists Q \sqsupseteq P, \exists Q_0 Q_1 \cdots Q_n \in \mathsf{Id}_{\le k}\, \Omega^*_{\le k} - \Phi(L_{\le k}) : Q = Q_0 * Q_1 * \cdots * Q_n \\ &\quad\iff \exists Q_0 Q_1 \cdots Q_n \in \mathsf{Id}_{\le k}\, \Omega^*_{\le k} : P \sqsubseteq Q_0 * Q_1 * \cdots * Q_n \notin L_{\le k} \\ &\quad\iff P \in \overline{L_{\le k}}^k, \end{aligned}$$

hence $\overline{L_{\le k}}^k = \Psi(\mathsf{Id}_{\le k}\, \Omega^*_{\le k} - \Phi(L_{\le k}))$. Lemma 25(4) allows us to conclude. □

Corollary 34. $\mathsf{iiPoms}_{\le k}$ *is regular for every* $k \ge 0$.

7 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 inclusion of regular languages is decidable.

A question which is still open is if it is decidable whether a regular language is deterministic or of bounded ambiguity and, related to that, whether HDAs are learnable. On a more general level, two things which are missing are a Büchi-type theorem on a logical characterisation of regular languages and a notion of recognizability. The latter is complicated by the fact that ipomsets do not form a monoid but rather a 2-category with lax tensor [6].

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 initial work is available in [3]. For weighted HDAs, the convolution algebras of [7] provide a useful framework.

Acknowledgement. We are in debt to Emily Clement, Thomas Colcombet, Christian Johansen, Georg Struth, and Safa Zouari for numerous discussions regarding the subjects of this paper. Any errors, however, are exclusively ours.

References

1. Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. *CoRR*, abs/2305.02873, 2023.
2. Marek A. Bednarczyk. *Categories of Asynchronous Systems.* PhD thesis, University of Sussex, UK, 1987.
3. Uli Fahrenberg. Higher-dimensional timed and hybrid automata. *Leibniz Transactions on Embedded Systems*, 8(2):03:1–03:16, 2022.

4. 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.
5. 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.
6. 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.
7. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Catoids and modal convolution algebras. *Algebra Universalis*, 84(10), 2023.
8. 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.
9. 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.
10. Peter C. Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. Wiley, 1985.
11. Eric Goubault. Labelled cubical sets and asynchronous transition systems: an adjunction. In *CMCIM*, 2002. http://www.lix.polytechnique.fr/~goubault/papers/cmcim02.ps.gz.
12. Jan Grabowski. On partial languages. *Fundamentae Informatica*, 4(2):427, 1981.
13. Ryszard Janicki and Maciej Koutny. Operational semantics, interval orders and sequences of antichains. *Fundamentae Informatica*, 169(1-2):31–55, 2019.
14. 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.
15. Kamal Lodaya and Pascal Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1-2):347–380, 2000.
16. Vaughan R. Pratt. Modeling concurrency with geometry. In *POPL*, pages 311–322, New York City, 1991. ACM Press.
17. Mike W. Shields. Concurrent machines. *The Computer Journal*, 28(5):449–465, 1985.
18. Rob J. van Glabbeek. Bisimulations for higher dimensional automata. Email message, June 1991. http://theory.stanford.edu/~rvg/hda.
19. Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. See also [20].
20. Rob J. van Glabbeek. Erratum to "On the expressiveness of higher dimensional automata". *Theoretical Computer Science*, 368(1-2):168–194, 2006.
21. Norbert Wiener. A contribution to the theory of relative position. *Proceedings of the Cambridge Philosophical Society*, 17:441–449, 1914.

Appendix: Proofs

Preliminaries. We prove here Lem. 4 and Lem. 9.

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, $2|P| - 2|S_P \cup T_P| + |S_P \cup T_P| - |S_P \cap T_P| = 2|P| - (|S_P| + |T_P| - |S_P \cap T_P|) - |S_P \cap T_P| = 2\,\mathsf{size}(P)$.

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

Lemma A.1 ([8]). *Let X be an HDA, $x, y \in X$ and $\gamma \in \mathit{Path}(X)^y_x$. Assume that $\mathsf{ev}(\gamma) = P * Q$ for ipomsets P and Q. Then there exist paths $\alpha \in \mathit{Path}(X)_x$ and $\beta \in \mathit{Path}(X)^y$ such that $\mathsf{ev}(\alpha) = P$, $\mathsf{ev}(\beta) = Q$ and $\mathit{tgt}(\alpha) = \mathit{src}(\beta)$.*

Lemma 9. *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 Lem. A.1 by induction. As to the second, if starters and terminators are alternating in $P_1 * \cdots * P_n$, then upsteps and downsteps are alternating in $\alpha_1 * \cdots * \alpha_n$.

Regular and non-regular languages. We prove here Prop. 14 and Prop. 16.

Proposition 14. *Let $k \geq 0$. If $L \in \mathscr{L}$ is regular, then so is $L_{\leq k}$.*

Proof. It suffices to remove from the HDA accepting L every cell x where $|\mathsf{ev}(x)| > k$.

Proposition 16. *The regular language $L = ([\begin{smallmatrix} a \\ b \end{smallmatrix}]\, cd + ab\, [\begin{smallmatrix} c \\ d \end{smallmatrix}])^+$ is of unbounded ambiguity.*

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

A cell $x \in X$ is *essential* if there exists an accepting path in X that contains x. A path is essential if all its cells are essential.

Lemma A.2. *Let X be an HDA with $\mathsf{L}(X) = L$. Let α and β be essential sparse paths in X with $\mathsf{ev}(\alpha) = [\begin{smallmatrix} a \\ b \end{smallmatrix}]\, cd$ and $\mathsf{ev}(\beta) = ab\, [\begin{smallmatrix} c \\ d \end{smallmatrix}]$. Then*

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

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

$$\bar\alpha = (v \nearrow^a \delta_b^0(q) \searrow_a \delta_a^0\delta_a^1(q) \nearrow^b \delta_a^1(q) \searrow_b x \nearrow^c e \searrow_c y \nearrow^d f \searrow_d z),$$
$$\bar\beta = (v' \nearrow^a g \searrow_a w' \nearrow^b h' \searrow_b x' \nearrow^c \delta_d^0(r') \searrow_c \delta_d^0\delta_c^1(r') \nearrow^d \delta_c^1(r') \searrow_d z')$$

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

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

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

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 $\mathsf{ev}(\gamma) = [\begin{smallmatrix} a \\ b \end{smallmatrix}] * [\begin{smallmatrix} c \\ d \end{smallmatrix}]$. 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, $\mathsf{ev}(\gamma') * [\begin{smallmatrix} a \\ b \end{smallmatrix}] * [\begin{smallmatrix} c \\ d \end{smallmatrix}] * \mathsf{ev}(\gamma'') \in L$: 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 (Proof of Prop. 16). 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 $P = [\begin{smallmatrix} a \\ b \end{smallmatrix}]\, cd$, $Q = ab\, [\begin{smallmatrix} c \\ d \end{smallmatrix}]$. For every sequence $\mathbf{R} = (R_1, \dots, R_n) \in \{P, Q\}^n$ let $\omega_{\mathbf{R}}$ be an accepting path such that $\mathsf{ev}(\omega_{\mathbf{R}}) = R_1 * \dots * R_n$. By Lem. 9, there exist paths $\omega^1_{\mathbf{R}}, \dots, \omega^n_{\mathbf{R}}$ such that $\mathsf{ev}(\omega^k_{\mathbf{R}}) = R_k$ and $\omega'_{\mathbf{R}} = \omega^1_{\mathbf{R}} * \dots * \omega^n_{\mathbf{R}}$ is an accepting path. Let $\bar\omega^k_{\mathbf{R}}$ be the path defined as in Lem. A.2 (*i.e.*, like $\bar\alpha$ if $R_k = P$ and $\bar\beta$ if $R_k = Q$). Finally, put $\bar\omega_{\mathbf{R}} = \bar\omega^1_{\mathbf{R}} * \dots * \bar\omega^n_{\mathbf{R}}$.

Now choose $\mathbf{R} \neq \mathbf{S} \in \{P, Q\}^n$. Assume that $\bar\omega_{\mathbf{R}} = \bar\omega_{\mathbf{S}}$. This implies that $\bar\omega^k_{\mathbf{R}} = \bar\omega^k_{\mathbf{S}}$ 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 Lem A.2 again, applied to $\alpha = \bar\omega^k_{\mathbf{R}}$ and $\beta = \bar\omega^k_{\mathbf{S}}$, we get $\bar\omega^k_{\mathbf{R}} \neq \bar\omega^k_{\mathbf{S}}$: 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 $\mathbf{R}$.

ST-automata. We prove here Lem. 19.

Lemma A.3. *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.

Lemma 19. Φ *respects boolean operations: for all* $L_1, L_2 \in \mathscr{L}$, $\Phi(L_1 \cap L_2) = \Phi(L_1) \cap \Phi(L_2)$ *and* $\Phi(L_1 \cup L_2) = \Phi(L_1) \cup \Phi(L_2)$. Ψ *respects regular operations: 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 claims for Φ are trivial consequences of the definitions. Regarding Ψ, the first claim follows easily using the fact that $(A \cup B){\downarrow} = A{\downarrow} \cup B{\downarrow}$. For the second, we have

$$\begin{aligned}\Psi(W_1) * \Psi(W_2) &= \{P_1 * \cdots * P_n \mid P_1 \cdots P_n \in W_1,\ \forall i : T_{P_i} = S_{P_{i+1}}\}{\downarrow} \\ &\quad * \{Q_1 * \cdots * Q_m \mid Q_1 \cdots Q_m \in W_2,\ \forall i : T_{Q_i} = S_{Q_{i+1}}\}{\downarrow} \\ &= \{P_1 * \cdots * P_n * P_{n+1} * \cdots * P_{n+m} \\ &\quad \mid P_1 \cdots P_n \in W_1, P_{n+1} \cdots P_{n+m} \in W_2,\ \forall i : T_{P_i} = S_{P_{i+1}}\}{\downarrow} \text{ by Lem A.3} \\ &= \Psi(W_1 W_2).\end{aligned}$$

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

Complement. We prove here Lem. 25.

Lemma 25. *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. $\overline{L}^0 = (\mathsf{iiPoms}_{\leq 0} - L){\downarrow} = \{\epsilon\} - L$.
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. $\overline{L}^k = (\mathsf{iiPoms}_{\leq k} - L){\downarrow} = (\mathsf{iiPoms}_{\leq k} - L_{\leq k}){\downarrow} = \overline{L_{\leq k}}^k$.

□