

Developments in Higher-Dimensional Automata Theory

Amazigh Amrane

EPITA Research Laboratory (LRE), Paris, France

Hugo Bazille

EPITA Research Laboratory (LRE), Paris, France

Uli Fahrenberg

EPITA Research Laboratory (LRE), Paris, France

Krzysztof Ziemiański

University of Warsaw, Poland

Abstract

We develop the language theory of higher-dimensional automata (HDAs). We show a pumping lemma which allows us to expose a class of non-regular ipomset languages. We also give an example of a regular language with unbounded ambiguity. Then we pass to decision and closure properties of regular languages. 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, no universal finite HDA exists, so 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.

2012 ACM Subject Classification Theory of computation → Automata extensions

Keywords and phrases higher-dimensional automaton, pomset with interfaces, automata theory, concurrency theory, pumping lemma, ambiguity

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Higher-dimensional automata (HDAs), introduced by Pratt and van Glabbeek [12, 14], are a general geometric model for non-interleaving concurrency which subsumes, for example, event structures and (safe) Petri nets [15]. HDAs of dimension one correspond to standard automata, whereas HDAs of dimension two are isomorphic to asynchronous transition systems [1, 9, 13]. 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 [3], which are sets of partially ordered multisets with interfaces (ipomsets) [5] that are closed under subsumptions. Follow-up papers introduce a language theory for HDAs, showing a Kleene theorem [4], which makes a connection between rational and regular ipomset languages (those accepted by finite HDAs), and a Myhill-Nerode theorem [7] 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. 3, 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. We also show that there is no universal HDA which accepts the language of all ipomsets, and that regular languages of unbounded ambiguity

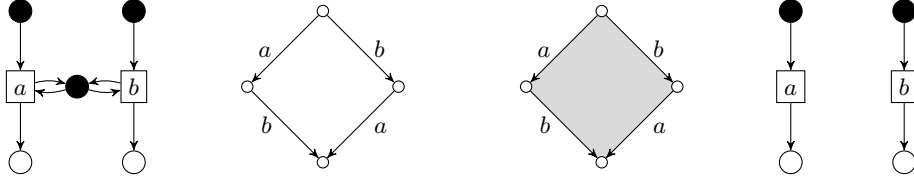


© A. Amrane, H. Bazille, U. Fahrenberg, K. Ziemiański;
licensed under Creative Commons License CC-BY 4.0



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



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

exist. The latter is a continuation of a theme from [7] which showed that not all HDAs are determinizable but left open the question of ambiguity.

In Sect. 4 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; but 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.

Finally, in Sect. 5, 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 in general, 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.

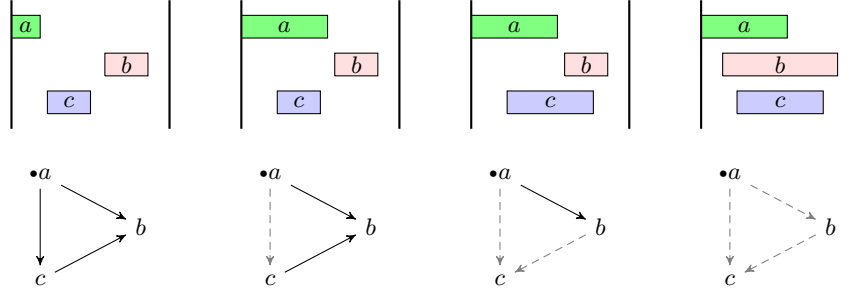
Another goal of this work was to obtain the above results using only “elementary” means as opposed to category-theoretic or topological ones. Indeed we do not use presheaves, track objects, cylinders, or any other of the categorical or topological constructions employed in [4, 7]. Categorical reasoning 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. 12);
- regular languages of unbounded ambiguity (Prop. 17);
- closure of regular languages under intersection (Prop. 16);
- closure of regular languages under width-bounded complement (Thm. 33);
- decidability of inclusion of regular languages (Thm. 22).

2 Preliminaries

By reason of exposition we introduce the *languages* of higher-dimensional automata before we introduce HDAs themselves.



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

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. For reasons of compositionality we also consider executions in which events may be active already at the beginning or remain active at the end of the execution.

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.

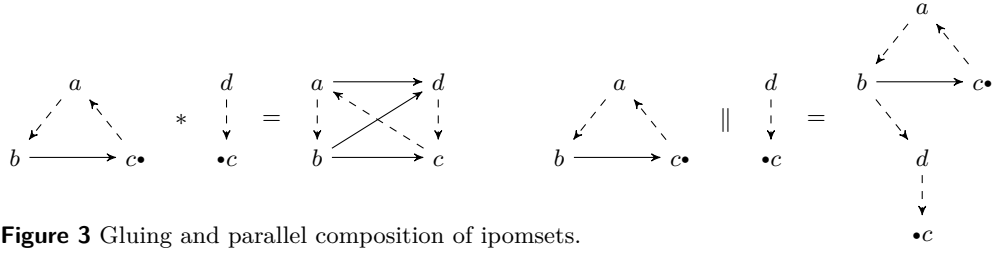
To make the above precise, let Σ be a finite alphabet. We define three notions, in increasing order of generality:

- 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),¹ and a labelling $\lambda_U : U \rightarrow \Sigma$.
- 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 $\lambda_P : P \rightarrow \Sigma$, such that for each $x \neq y$ in P , $x <_P y$, $y <_P x$, $x \dashrightarrow_P y$, or $y \dashrightarrow_P x$.
- 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 for pomsets 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.

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 qualifier “strict”.



■ **Figure 3** Gluing and parallel composition of ipomsets.

An ipomset P is *interval* if $<_P$ is an interval order [8]; that is, if it admits an interval representation given by functions b and e from P to real numbers such that $b(x) \leq e(x)$ for all $x \in P$ and $x <_P y$ iff $e(x) < b(y)$ for all $x, y \in P$. 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 [17]. We let iiPoms denote the set of (interval) ipomsets.

Ipomsets may be *refined* by shortening the corresponding 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 $f : P \rightarrow Q$ 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 may ignore such non-essential event order.)

This definition adapts the standard one [10] to event orders and interfaces. Intuitively, P has more order and less concurrency than Q . In Fig. 2 there is a sequence of subsumptions from left to right. *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 $f(x) \dashrightarrow_Q f(y)$.

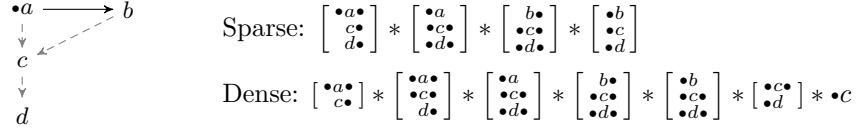
Due to the requirement that all elements are ordered by $<$ or \dashrightarrow , there is at most one isomorphism between any two ipomsets. Hence we may switch freely between ipomsets and their isomorphism classes. We will also call these equivalence classes ipomsets and often confuse equality and isomorphism.

Compositions. The standard serial and parallel compositions of pomsets [10] 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 [5] 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 with $T_P = S_Q$ and such that $x \dashrightarrow_P y$ iff $x \dashrightarrow_Q y$ for all $x, y \in T_P = S_Q$ and the restrictions $\lambda_P|_{T_P} = \lambda_Q|_{S_Q}$, then $P * Q = (P \cup Q, <, \dashrightarrow, S_P, T_Q, \lambda)$, where



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

- $x < y$ if $x <_P y$, $x <_Q y$, or $x \in P - T_P$ and $y \in Q - S_Q$;²
- \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 confuse 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 $T_P \rightarrow S_Q$. 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 $S_U = U$. These will be so important later that we introduce special notation, writing $_A \uparrow U$ and $U \downarrow_B$ for the above. The intuition is that a starter does nothing but start the events in $A = U - S_U$, and a terminator terminates the events in $B = U - T_U$. Starter $_A \uparrow U$ is *elementary* if $U - A$ 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 $\text{id}_U = \emptyset \uparrow U = U \downarrow \emptyset$.

The *width* $\text{wid}(P)$ of an ipomset P is the cardinality of a maximal $<$ -antichain; its *size* is $\text{size}(P) = |P| - \frac{1}{2}(|S_P| + |T_P|)$. Identities are exactly the ipomsets of size 0, while elementary starters and terminators are exactly the ipomsets of size $\frac{1}{2}$.

► **Lemma 1** ([4]). *Let P and Q be ipomsets. Then*

1. $\text{wid}(P \parallel Q) = \text{wid}(P) + \text{wid}(Q)$ and $\text{size}(P \parallel Q) = \text{size}(P) + \text{size}(Q)$;
2. if $T_P = S_Q$, then $\text{wid}(PQ) = \max(\text{wid}(P), \text{wid}(Q))$ and $\text{size}(PQ) = \text{size}(P) + \text{size}(Q)$;
3. if $P \sqsubseteq Q$, then $\text{wid}(P) \leq \text{wid}(Q)$ and $\text{size}(P) = \text{size}(Q)$.

Any ipomset can be decomposed as a gluing of starters and terminators [5], see also [11]. Such a presentation we call a *step decomposition*. If starters and terminators are alternating, the decomposition is called *sparse*; if they are all elementary, then it is *dense*. The differences between dense and sparse step decompositions are illustrated in Fig. 4.

► **Lemma 2** ([7]). *Every ipomset P has a unique sparse step decomposition.*

Dense decompositions are generally not unique, but they all have the same length. (Due to space constraints, the proof of this and many of the later lemmas is shown in appendix.)

► **Lemma 3.** *Every dense step decomposition of ipomset P has length $2 \text{size}(P)$.*

² We use “ $-$ ” for set difference instead of the perhaps more common “ \setminus ”.

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

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

Note that $(A \cup B)\downarrow = A\downarrow \cup B\downarrow$ for all $A, B \subseteq \text{iiPoms}$, 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 $L \subseteq \text{iiPoms}$ for which $L\downarrow = L$. The set of all languages is denoted $\mathcal{L} \subseteq 2^{\text{iiPoms}}$.

The *width* of a language L is $\text{wid}(L) = \sup\{\text{wid}(P) \mid P \in L\}$. For $k \geq 0$ and $L \in \mathcal{L}$, denote $L_{\leq k} = \{P \in L \mid \text{wid}(P) \leq k\}$. L is *k-dimensional* if $L = L_{\leq k}$. We let $\mathcal{L}_{\leq k} = \mathcal{L} \cap \text{iiPoms}_{\leq k}$ 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 , $*$, \parallel and (Kleene plus) $^+$ for languages are defined as follows.

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

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

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

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

► **Lemma 4.** For all $A_1, A_2 \subseteq \text{iiPoms}$, $A_1\downarrow * A_2\downarrow = \{P_1 * P_2 \mid P_1 \in A_1, P_2 \in A_2\}\downarrow$.

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

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

we may now state the central result of [7].

► **Theorem 5 ([7]).** A language $L \in \mathcal{L}$ is rational iff $\text{suff}(L)$ is finite, iff $\text{pref}(L)$ is finite.

Lemma 1 also implies the following.

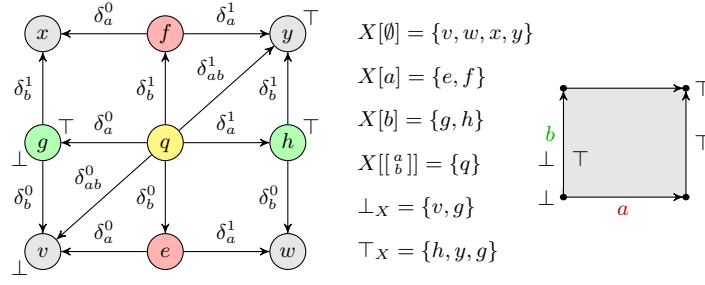
► **Corollary 6.** Any rational language has finite width.

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*

$$X = (X, \text{ev}, \{\delta_{A,U}^0, \delta_{A,U}^1 \mid U \in \square, A \subseteq U\})$$

consists of a set of cells X together with a function $\text{ev} : X \rightarrow \square$. For a conclist U we write $X[U] = \{x \in X \mid \text{ev}(x) = U\}$ for the cells of type U . Further, for every $U \in \square$ and $A \subseteq U$ there are face maps $\delta_{A,U}^0, \delta_{A,U}^1 : X[U] \rightarrow X[U - A]$ (we will omit the extra subscript U from now) which satisfy $\delta_A^\nu \delta_B^\mu = \delta_B^\mu \delta_A^\nu$ for $A \cap B = \emptyset$ and $\nu, \mu \in \{0, 1\}$. The upper face maps δ_A^1 transform a cell x into one in which the events in A have terminated, whereas the lower face maps δ_A^0 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.



■ **Figure 5** A two-dimensional HDA X on $\Sigma = \{a, b\}$, as a combinatorial object (left) and in a geometric realisation (right).

A *higher-dimensional automaton* (HDA) $X = (X, \perp_X, \top_X)$ is a precubical set together with subsets $\perp_X, \top_X \subseteq X$ 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 $\dim(X) = \sup\{|\text{ev}(x)| \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$.³

A standard automaton is the same as a one-dimensional HDA X with the property that for all $x \in \perp_X \cup \top_X$, $\text{ev}(x) = \emptyset$: cells in $X[\emptyset]$ are states, cells in $X[\{a\}]$ for $a \in \Sigma$ are a -labelled transitions, and face maps $\delta_{\{a\}}^0$ and $\delta_{\{a\}}^1$ 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. Figure 5 shows a two-dimensional HDA. In this and other HDA examples, 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 $\alpha = (x_0, \varphi_1, x_1, \dots, x_{n-1}, \varphi_n, x_n)$ consisting of cells x_i of X and symbols φ_i which indicate face map types: for every $i \in \{1, \dots, n\}$, $(x_{i-1}, \varphi_i, x_i)$ is either

- $(\delta_A^0(x_i), \nearrow^A, x_i)$ for $A \subseteq \text{ev}(x_i)$ (an *upstep*)
- or $(x_{i-1}, \searrow_A, \delta_A^1(x_{i-1}))$ for $A \subseteq \text{ev}(x_{i-1})$ (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 $\text{src}(\alpha) = x_0$ and $\text{tgt}(\alpha) = x_n$. The set of all paths in X starting at $Y \subseteq X$ and terminating in $Z \subseteq X$ is denoted by $\text{Path}(X)_Y^Z$; we write $\text{Path}(X)_Y = \text{Path}(X)_Y^X$, $\text{Path}(X)^Z = \text{Path}(X)_X^Z$, and $\text{Path}(X) = \text{Path}(X)_X^X$. A path α is *accepting* if $\text{src}(\alpha) \in \perp_X$ and $\text{tgt}(\alpha) \in \top_X$. Paths α and β may be concatenated if $\text{tgt}(\alpha) = \text{src}(\beta)$; 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* $\text{ev}(\alpha)$ of a path α is defined recursively as follows:

- if $\alpha = (x)$, then $\text{ev}(\alpha) = \text{id}_{\text{ev}(x)}$;
- if $\alpha = (y \nearrow^A x)$, then $\text{ev}(\alpha) = \uparrow_A \text{ev}(x)$;

³ Precubical sets are presheaves over a category on objects \square , and then HDAs form a category with the induced morphisms, see [4] for details. (We will not use these facts.)

- if $\alpha = (x \searrow_A y)$, then $\text{ev}(\alpha) = \text{ev}(x) \downarrow_A$;
- if $\alpha = \alpha_1 * \dots * \alpha_n$ is a concatenation, then $\text{ev}(\alpha) = \text{ev}(\alpha_1) * \dots * \text{ev}(\alpha_n)$.

Note that downsteps in α correspond to starters in $\text{ev}(\alpha)$ and upsteps correspond to terminators. Path equivalence $\alpha \simeq \beta$ implies $\text{ev}(\alpha) = \text{ev}(\beta)$ [4]. Further, if $\alpha = \alpha_1 * \dots * \alpha_n$ is a sparse path, then $\text{ev}(\alpha) = \text{ev}(\alpha_1) * \dots * \text{ev}(\alpha_n)$ is a sparse step decomposition.

The *language* of an HDA X is $\mathbf{L}(X) = \{\text{ev}(\alpha) \mid \alpha \text{ accepting path in } X\}$.⁴

Languages of HDAs are sets of (interval) ipomsets which are closed under subsumption [4], *i.e.*, languages in our sense. A language is *regular* if it is the language of a finite HDA.

► **Theorem 7** ([4]). *A language is regular iff it is rational.*

We finish the section by two lemmas.

► **Lemma 8** ([7]). *Let X be an HDA, $x, y \in X$ and $\gamma \in \text{Path}(X)_x^y$. Assume that $\text{ev}(\gamma) = P * Q$ for ipomsets P and Q . Then there exist paths $\alpha \in \text{Path}(X)_x$ and $\beta \in \text{Path}(X)_y$ such that $\text{ev}(\alpha) = P$, $\text{ev}(\beta) = Q$ and $\text{tgt}(\alpha) = \text{src}(\beta)$.*

► **Lemma 9.** *Let X be an HDA, $P \in \mathbf{L}(X)$ and $P = P_1 * \dots * P_n$ be any decomposition. Then there exists an accepting path $\alpha = \alpha_1 * \dots * \alpha_n$ in X such that $\text{ev}(\alpha_i) = P_i$ for all i . If $P = P_1 * \dots * P_n$ is a sparse step decomposition, then $\alpha = \alpha_1 * \dots * \alpha_n$ is sparse.*

3 Regular and non-regular languages

Universality. The *universal* HDA \mathcal{U} consists of the precubical set $\mathcal{U} = \square$ with $\text{ev}(\mathcal{U}) = \mathcal{U}$, face maps $\delta_{A,U}^\nu(\mathcal{U}) = \mathcal{U} - A$ and start and accept cells $\perp_{\mathcal{U}} = \top_{\mathcal{U}} = \mathcal{U}$. That is, \mathcal{U} has one cell for every conclist ($\mathcal{U}[U] = U$), which uniquely determines the face maps.⁵ Now $\mathbf{L}(\mathcal{U}) = \text{iiPoms}$ (as shown in Lem. 10) but \mathcal{U} is an *infinite* HDA. Indeed, given that the width of iiPoms is unbounded (and Cor. 6), no finite HDA exists which recognises all of iiPoms .

If we only regard conclists of bounded width, this changes. Let $\mathcal{U}_{\leq k} = \{U \in \square \mid |U| \leq k\}$ with face maps as above, and again all cells start and accept cells. Then $\mathcal{U}_{\leq k}$ is a finite HDA, and $\mathbf{L}(\mathcal{U}_{\leq k}) = \text{iiPoms}_{\leq k}$ (see Lem. 10). Hence width-bounded universal HDAs exist, and by Thm. 22 below, width-bounded universality is decidable.

► **Lemma 10.** $\mathbf{L}(\mathcal{U}) = \text{iiPoms}$ and $\mathbf{L}(\mathcal{U}_{\leq k}) = \text{iiPoms}_{\leq k}$.

Proof. As proofs for both statements are very similar, we concentrate here on $\mathcal{U}_{\leq k}$. We use structural induction on a step decomposition of an ipomset $P \in \text{iiPoms}_{\leq k}$. Note that all paths in \mathcal{U}_k are accepting.

- If $P = \text{id}_U$ is an identity of width $\leq k$, then $U \in \mathcal{U}_{\leq k}$ and P is recognised by a path (U).
- If P is a starter $A \uparrow U$ or a terminator $U \downarrow_B$, then $|A|, |B|, |U| \leq k$, and P is recognised by a path $(U - A, \nearrow^A, U)$ or $(U, \searrow_B, U - B)$.
- If $P = P_1 P_2$ is a decomposition, then $T_{P_1} = S_{P_2} =: U$. By hypothesis there are paths α_1 and α_2 in $\mathcal{U}_{\leq k}$ for which $\text{ev}(\alpha_1) = P_1$ and $\text{ev}(\alpha_2) = P_2$. We must have $\text{tgt}(\alpha_1) = \text{src}(\alpha_2) = U$, and then the concatenation $\alpha_1 \alpha_2$ recognises P . ◀

► **Corollary 11.** $\text{iiPoms}_{\leq k}$ is regular for every $k \geq 0$.

⁴ Every ipomset P may be converted into a *track object* \square^P , see [4], which is an HDA with the property that for any HDA X , $P \in \mathbf{L}(X)$ iff there is a morphism $\square^P \rightarrow X$. (We will not use these facts.)

⁵ \mathcal{U} is the terminal object in the category of HDAs. Using track objects, Lem. 10 follows immediately.

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

► **Lemma 12.** *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 $Q_1 * \dots * Q_i * (Q_{i+1} * \dots * Q_j)^+ * Q_{j+1} * \dots * Q_n \subseteq L$.*

Proof. Let X be an HDA accepting L and $k > |X|$. Let α be an accepting path in X such that $\text{ev}(\alpha) = P$. By Lem. 9 there exists an accepting path $\beta = \beta_1 * \dots * \beta_n$ such that $\text{ev}(\beta_i) = Q_i$ for all i . Denote $x_i = \text{tgt}(\beta_i) = \text{src}(\beta_{i+1})$. 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, $\text{src}(\beta_{i+1}) = \text{tgt}(\beta_j)$, and for every $r \geq 1$

$$\beta_1 * \dots * \beta_i * (\beta_{i+1} * \dots * \beta_j)^r * \beta_{j+1} * \dots * \beta_n$$

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

► **Corollary 13.** *Let L be a regular language. There exists $k \in \mathbb{N}$ such that any $P \in L$ with $\text{size}(P) > k$ 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. 12 to a dense decomposition $P = Q_1 * \dots * Q_{2 \cdot \text{size}(P)}$, cf. Lem. 3. We may now expose a language which is not regular.

► **Proposition 14.** *The language $L = \{[a]^n * a^n \mid n \geq 1\} \downarrow$ 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 Prop. 14. We give two proofs. The first uses Thm. 5: for every $k \geq 1$, $[a]^k \setminus L = \{[a]^n * a^{n+k} \mid n \geq 0\} \downarrow$, and these are different for different k , so $\text{suff}(L)$ is infinite.

The second proof uses Lem. 12. Assume L to be regular, let k be the constant from the lemma, and take $P = [a]^k * a^k = Q_1 * \dots * Q_k * Q_{k+1}$, where $Q_1 = \dots = Q_k = [a]$ and $Q_{k+1} = a^k$. For $m = 0$ we obtain that $[a]^{k+(j-i)r} a^k \in L$ for all r and some $j - i > 0$: a contradiction. ◀

Using notation for iterated parallel compositions, 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 $a^{\parallel 1} = a$ and $a^{\parallel k} = a \parallel a^{\parallel k-1}$ for $k \geq 2$: the k -fold parallel product of a with itself. Now let $k \geq 1$ and

$$L = \{(a^{\parallel k+1})^n * P^n \mid n \geq 0, P \in \{a^{\parallel k+1}\} \downarrow - \{a^{\parallel k+1}\}\} \downarrow.$$

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 $L_{\leq k} = ((\{a^{\parallel k+1}\} \downarrow - \{a^{\parallel k+1}\})^2)^+$ is. Yet k -restrictions of regular languages remain regular:

► **Proposition 15.** *Let $k \geq 0$. If 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 16.** *The regular languages are closed under \cap .*

Proof. We again give two proofs, one algebraic using Thm. 5 and another, constructive proof using Thm. 7. For the first proof, let L_1 and L_2 be regular, then $\text{suff}(L_1)$ and $\text{suff}(L_2)$ are both finite. Now

$$\begin{aligned} \text{suff}(L_1 \cap L_2) &= \{P \setminus (L_1 \cap L_2) \mid P \in \text{iiPoms}\} \\ &= \{\{Q \in \text{iiPoms} \mid PQ \in L_1 \cap L_2\} \mid P \in \text{iiPoms}\} \\ &= \{\{Q \in \text{iiPoms} \mid PQ \in L_1\} \cap \{Q \in \text{iiPoms} \mid PQ \in L_2\} \mid P \in \text{iiPoms}\} \\ &= \{P \setminus L_1 \cap P \setminus L_2 \mid P \in \text{iiPoms}\} \\ &\subseteq \{M_1 \cap M_2 \mid M_1 \in \text{suff}(L_1), M_2 \in \text{suff}(L_2)\} \end{aligned}$$

which is thus finite.

For the second, constructive proof, let X_1 and X_2 be HDAs; we construct an HDA X with $\mathbf{L}(X) = \mathbf{L}(X_1) \cap \mathbf{L}(X_2)$:⁶

$$\begin{aligned} X &= \{(x_1, x_2) \in X_1 \times X_2 \mid \text{ev}_1(x_1) = \text{ev}_2(x_2)\}, & \text{ev}((x_1, x_2)) &= \text{ev}_1(x_1) = \text{ev}_2(x_2), \\ \delta_A^\nu(x_1, x_2) &= (\delta_A^\nu(x_1), \delta_A^\nu(x_2)), & \perp &= \perp_1 \times \perp_2, & \top &= \top_1 \times \top_2. \end{aligned}$$

For the inclusion $\mathbf{L}(X) \subseteq \mathbf{L}(X_1) \cap \mathbf{L}(X_2)$, any accepting path α in X projects to accepting paths β in X_1 and γ in X_2 , and then $\text{ev}(\beta) = \text{ev}(\gamma) = \text{ev}(\alpha)$. 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 $P \in \mathbf{L}(X_1) \cap \mathbf{L}(X_2)$ and $P = P_1 * \dots * P_n$ the sparse step decomposition. Let $\beta = \beta_1 * \dots * \beta_n$ and $\gamma = \gamma_1 * \dots * \gamma_n$ be sparse accepting paths for P in X_1 and X_2 , respectively, such that $\text{ev}(\alpha_i) = \text{ev}(\beta_i) = P_i$ for all i , cf. Lem. 9.

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

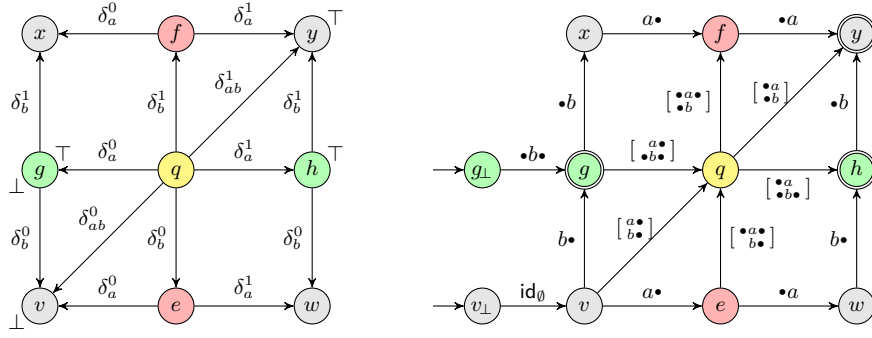
Ambiguity. It is shown in [7] that not all languages are determinizable, that is, there exist regular languages which cannot be recognised by deterministic HDAs. We have not introduced deterministic HDAs here and will not need them in what follows; instead we prove a strengthening of that result. Say that an HDA X is *k-ambiguous*, for $k \geq 1$, if every $P \in \mathbf{L}(X)$ 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 17.** *The regular language $L = ([\begin{smallmatrix} a \\ b \end{smallmatrix}]cd + ab[\begin{smallmatrix} c \\ d \end{smallmatrix}])^+$ is of unbounded ambiguity.*

4 ST-automata

We define a variant of a construction from [4] 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 $k \geq 0$, $\Omega_{\leq k} = \Omega \cap \text{iiPoms}_{\leq k}$; the sets $\Omega_{\leq k}$ are all finite.

⁶ This is the product in the category of HDAs. Using track objects, the property follows immediately.



■ **Figure 6** HDA of Fig. 5 and corresponding ST-automaton (with identity loops not displayed).

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 $Q = X \cup \{x_\perp \mid x \in \perp_X\}$, $I = \{x_\perp \mid x \in \perp_X\}$, $F = \top_X$, and

$$E = \{(\delta_A^0(x), \uparrow_A U, x) \mid x \in X[U], A \subseteq U\} \cup \{(x, U \downarrow_A, \delta_A^1(x)) \mid x \in X[U], A \subseteq U\} \\ \cup \{(x_\perp, \text{id}_U, x) \mid x \in \perp_X \cap X[U]\}.$$

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 \mathcal{W} . Further, $W(\mathcal{A})$ denotes the set of words accepted by a finite automaton \mathcal{A} .

► **Example 18.** Figure 6 displays the ST-automaton $G_2(X)$ pertaining to the HDA X in Fig. 5, with the identity loops $(z, \text{id}_{\text{ev}(z)}, z)$ 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 $L(X)$. We have $L(X) = \{b\bullet, \bullet b\bullet, [\begin{smallmatrix} a \\ b \end{smallmatrix}], [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}], [\begin{smallmatrix} a \\ b \end{smallmatrix}], [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}]\} \downarrow$ and $W(G_2(X)) = \{\text{id}_\emptyset b\bullet, \bullet b\bullet, \text{id}_\emptyset [\begin{smallmatrix} a \\ b \end{smallmatrix}] [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}], \bullet b\bullet [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}] [\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}], \dots\}$.

Define functions $\Phi : \mathcal{L} \rightarrow \mathcal{W}$ and $\Psi : \mathcal{W} \rightarrow \mathcal{L}$ by

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

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 $\text{id}_U \in \Omega$ are not units in \mathcal{W}); this will not be a problem for us later. It is clear that $\Psi(\Phi(L)) = L$ for all $L \in \mathcal{L}$, 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 19.** If $W = \{a\bullet \bullet b\}$ (the word language containing the concatenation of $a\bullet$ and $\bullet b$), then $\Psi(W) = \emptyset$ and thus $\Phi(\Psi(W)) = \emptyset \not\subseteq W$. If $W = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}][\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}]\}$, then $\Psi(W) = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}], ab, ba\}$ and $\Phi(\Psi(W)) \not\subseteq W$.

► **Lemma 20.** Φ respects boolean operations: for all $L_1, L_2 \in \mathcal{L}$, $\Phi(L_1 \cap L_2) = \Phi(L_1) \cap \Phi(L_2)$ and $\Phi(L_1 \cup L_2) = \Phi(L_1) \cup \Phi(L_2)$. Ψ respects regular operations: for all $W_1, W_2 \in \mathcal{W}$, $\Psi(W_1 \cup W_2) = \Psi(W_1) \cup \Psi(W_2)$, $\Psi(W_1 W_2) = \Psi(W_1) * \Psi(W_2)$, 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 $\text{Id} = \{\text{id}_U \mid U \in \square\} \subseteq \Omega$ and, for any $k \geq 0$, $\text{Id}_{\leq k} = \text{Id} \cap \text{iiPoms}_{\leq k} \subseteq \Omega_k$. Then $\text{Id}_{\leq k} \Omega_{\leq k}^* \subseteq \Omega_{\leq k}^*$ (which is a regular word language) denotes the set of all words starting with an identity.

► **Lemma 21.** For any HDA X and $k \geq \dim(X)$, $\mathbf{W}(G_k(X)) = \Phi(\mathbf{L}(X)) \cap \text{Id}_{\leq k} \Omega_{\leq k}^*$.

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

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

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

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

► **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 $k = \max(\dim(X_1), \dim(X_2))$. By Lem. 20, $L_1 \subseteq L_2$ implies $\Phi(L_1) \subseteq \Phi(L_2)$ and $\Phi(L_1) \cap \text{Id}_{\leq k} \Omega_{\leq k}^* \subseteq \Phi(L_2) \cap \text{Id}_{\leq k} \Omega_{\leq k}^*$. Moreover, if $L_1 \neq L_2$, then also $\Phi(L_1) \cap \text{Id}_{\leq k} \Omega_{\leq k}^* \neq \Phi(L_2) \cap \text{Id}_{\leq k} \Omega_{\leq k}^*$, since every ipomset admits a step decomposition starting with an identity. Thus,

$$L_1 \subseteq L_2 \iff \Phi(L_1) \cap \text{Id}_{\leq k} \Omega_{\leq k}^* \subseteq \Phi(L_2) \cap \text{Id}_{\leq k} \Omega_{\leq k}^* \iff \mathbf{W}(G_k(X_1)) \subseteq \mathbf{W}(G_k(X_2))$$

by Lem. 21. Given that these are finite automata, the latter inclusion is decidable. ◀

5 Complement

The *complement* of a language $L \subseteq \text{iiPoms}$, i.e., $\text{iiPoms} - L$, is generally not down-closed and thus not a language. If we define $\overline{L} = (\text{iiPoms} - L)\downarrow$, 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.

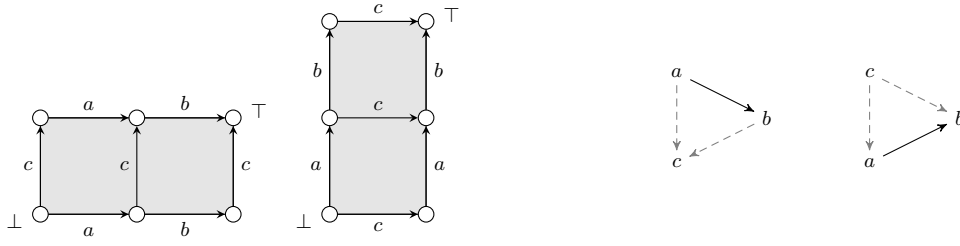
► **Proposition 23.** If L is regular, then \overline{L} has infinite width and hence is not regular.

Proof. By Cor. 6, $\text{wid}(L)$ is finite. For any $k > \text{wid}(L)$, $\text{iiPoms} - L$ contains all ipomsets of width k , hence $\{\text{wid}(P) \mid P \in \overline{L}\}$ 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 $L \in \mathcal{L}$ is $\overline{L}^k = (\text{iiPoms}_{\leq k} - L)\downarrow$.

► **Lemma 25.** Let L and M be languages.



■ **Figure 7** HDA X which accepts language L of Ex. 28 and the two generating ipomsets in L .

1. $\overline{L}^0 = \{\text{id}_\emptyset\} - L$.
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 $k \geq 0$, $\overline{}^k$ is a pseudocomplement on the lattice $(\mathcal{L}_{\leq k}, \supseteq)$, that is, for any $L, M \in \mathcal{L}_{\leq k}$, $L \cup M = \text{iiPoms}_k$ iff $\overline{L}^k \subseteq M$.

Proof. Let $L, M \in \mathcal{L}_{\leq k}$ such that $L \cup M = \text{iiPoms}_k$ and $P \in \overline{L}^k$. There exists $Q \in \text{iiPoms}_{\leq k}$ 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 $L, M \in \mathcal{L}_{\leq k}$ such that $\overline{L}^k \subseteq M$ and $P \in \text{iiPoms}_{\leq k} - M$. 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 $L \cup M = \text{iiPoms}_k$. ◀

The pseudocomplement property immediately gets us the following.

► **Corollary 27.** Let $k \geq 0$ and $L, M \in \mathcal{L}_{\leq k}$. Then $L \cup \overline{L}^k = \text{iiPoms}_{\leq k}$, $\overline{\overline{L}^k}^k = \overline{L}^k$, $\overline{L} \cap \overline{M}^k = \overline{L}^k \cup \overline{M}^k$, $\overline{L \cup M}^k \subseteq \overline{L}^k \cap \overline{M}^k$, and $\overline{\overline{L \cup M}^k}^k = \overline{L}^k \cup \overline{M}^k$. Further, $\overline{L}^k = \emptyset$ iff $L = \text{iiPoms}_{\leq k}$.

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

► **Example 28.** Let $A = \{P \in \text{iiPoms}_{\leq 2} \mid abc \sqsubseteq P\}$, $L = \{[a \rightarrow_c^b], [a \rightarrow_b^c]\} \downarrow$ 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 $[a \rightarrow_c^b]$, while the right part accepts $[a \rightarrow_b^c]$.

Now $abc \sqsubseteq [a \rightarrow_b^c]$ which is not in L , so that $abc \in \overline{L}^2$. Similarly, $abc \sqsubseteq [a \rightarrow_c^b] \notin M$, 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 $\text{wid}(P) \leq 2$ 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 $\overline{L}^3 = \text{iiPoms}_{\leq 3}$, 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 $\overline{L}^k = \emptyset$ as soon as $\text{wid}(L) < k$.

We say that $L \in \mathcal{L}$ is k -skeletal if $L = \overline{\overline{L}^k}^k$. Let \mathcal{S}_k be the set of all k -skeletal languages. We characterise \mathcal{S}_k in the following. By $\overline{\overline{L}^k}^k = \overline{L}^k$ (Cor. 27), $\mathcal{S}_k = \{\overline{L}^k \mid L \in \mathcal{L}\}$, i.e., \mathcal{S}_k is the image of \mathcal{L} under $\overline{}^k$. (This is a general property of pseudocomplements.)

Define $M_k = \{P \in \text{iiPoms}_{\leq k} \mid \forall Q \in \text{iiPoms}_{\leq k} : Q \neq P \implies P \not\sqsubseteq Q\}$, the set of all \sqsubseteq -maximal elements of $\text{iiPoms}_{\leq k}$. In particular, $M_k \downarrow = \text{iiPoms}_{\leq k}$. Note that $P \in M_k$ does not imply $\text{wid}(P) = k$: for example, $[a] \in M_3$.

► **Lemma 29.** For any $L \in \mathcal{L}$, $\overline{L}^k = (M_k - L) \downarrow$.

Proof. We have

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

► **Corollary 30.** *Let $L \in \mathcal{L}$ and $k \geq 0$, then $\overline{L}^k = \text{iiPoms}_{\leq k}$ iff $L \cap \mathbf{M}_k = \emptyset$.*

► **Proposition 31.** $\mathcal{S}_k = \{A \downarrow \mid A \subseteq \mathbf{M}_k\}$.

Proof. Inclusion \subseteq follows from Lem. 29. For the other direction, $A \subseteq \mathbf{M}_k$ implies

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

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

► **Corollary 32.** *For $L, M \in \mathcal{L}$, $\overline{L}^k = \overline{M}^k$ iff $L \cap \mathbf{M}_k = M \cap \mathbf{M}_k$.*

We can now show that bounded complement preserves regularity.

► **Theorem 33.** *If $L \in \mathcal{L}$ is regular, then for all $k \geq 0$ so is \overline{L}^k .*

Proof. By Prop. 15, $L_{\leq k}$ is regular. Let X be an HDA such that $\mathbf{L}(X) = L_{\leq k}$ and $k = \dim(X)$. The $\Omega_{\leq k}$ -language $\text{Id}_{\leq k} \Omega_{\leq k}^* \cap \Phi(\mathbf{L}(X))$ is regular by Lem. 21, hence so is $\text{Id}_{\leq k} \Omega_{\leq k}^* - \Phi(\mathbf{L}(X))$. Ψ preserves regularity, so $\Psi(\text{Id}_{\leq k} \Omega_{\leq k}^* - \Phi(\mathbf{L}(X)))$ is a regular ipomset language. Now for $P \in \text{iiPoms}_{\leq k}$ we have

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

hence $\overline{L_{\leq k}}^k = \Psi(\text{Id}_{\leq k} \Omega_{\leq k}^* - \Phi(L_{\leq k}))$. Lemma 25(4) allows us to conclude. \blacktriangleleft

6 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 [5].

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 [2]; for weighted HDAs, the convolution algebras of [6] provide a useful framework.

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

References

- 1 Marek A. Bednarczyk. *Categories of Asynchronous Systems*. PhD thesis, University of Sussex, UK, 1987.
- 2 Uli Fahrenberg. Higher-dimensional timed and hybrid automata. *Leibniz Transactions on Embedded Systems*, 8(2):03:1–03:16, 2022.
- 3 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>.
- 4 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>.
- 5 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>.
- 6 Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Catoids and modal convolution algebras. *Algebra Universalis*, 84(10), 2023.
- 7 Uli Fahrenberg and Krzysztof Ziemiański. A Myhill-Nerode theorem for higher-dimensional automata. *CoRR*, abs/2210.08298, 2022. <https://arxiv.org/abs/2210.08298>. Accepted at Petri Nets 2023.
- 8 Peter C. Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. Wiley, 1985.
- 9 Eric Goubault. Labelled cubical sets and asynchronous transition systems: an adjunction. In *CMCIM*, 2002. <http://www.lix.polytechnique.fr/~goubault/papers/cmcim02.ps.gz>.
- 10 Jan Grabowski. On partial languages. *Fundamentae Informaticae*, 4(2):427, 1981.
- 11 Ryszard Janicki and Maciej Koutny. Operational semantics, interval orders and sequences of antichains. *Fundamentae Informaticae*, 169(1-2):31–55, 2019.
- 12 Vaughan R. Pratt. Modeling concurrency with geometry. In *POPL*, pages 311–322, New York City, 1991. ACM Press.
- 13 Mike W. Shields. Concurrent machines. *The Computer Journal*, 28(5):449–465, 1985.
- 14 Rob J. van Glabbeek. Bisimulations for higher dimensional automata. Email message, June 1991. <http://theory.stanford.edu/~rvg/hda>.
- 15 Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. See also [16].
- 16 Rob J. van Glabbeek. Erratum to “On the expressiveness of higher dimensional automata”. *Theoretical Computer Science*, 368(1-2):168–194, 2006.
- 17 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. 3, Lem. 4 and Lem. 9.

► **Lemma 3.** *Every dense step decomposition of ipomset P has length $2\text{size}(P)$.*

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 decomposition and every event in $S_P \cup T_P - (S_P \cap T_P)$ to one element. The length of the decomposition is, thus, $2|P| - 2|S_P \cup T_P| + |S_P \cup T_P| - |S_P \cap T_P| = 2|P| - (|S_P| + |T_P| - |S_P \cap T_P|) - |S_P \cap T_P| = 2\text{size}(P)$. ◀

► **Lemma 4.** *For all $A_1, A_2 \subseteq \text{iiPoms}$, $A_1 \downarrow * A_2 \downarrow = \{P_1 * P_2 \mid P_1 \in A_1, P_2 \in A_2\} \downarrow$.*

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 9.** *Let X be an HDA, $P \in \mathbf{L}(X)$ and $P = P_1 * \dots * P_n$ be any decomposition. Then there exists an accepting path $\alpha = \alpha_1 * \dots * \alpha_n$ in X such that $\text{ev}(\alpha_i) = P_i$ for all i . If $P = P_1 * \dots * P_n$ is a sparse step decomposition, then $\alpha = \alpha_1 * \dots * \alpha_n$ is sparse.*

Proof. The first claim follows from Lem. 8 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 $\alpha_1 * \dots * \alpha_n$. ◀

Regular and non-regular languages. We prove here Prop. 15 and Prop. 17.

► **Proposition 15.** *Let $k \geq 0$. If L is regular, then so is $L_{\leq k}$.*

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

► **Proposition 17.** *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 34.** *Let X be an HDA with $\mathbf{L}(X) = L$. Let α and β be essential sparse paths in X with $\text{ev}(\alpha) = [\begin{smallmatrix} a \\ b \end{smallmatrix}]cd$ and $\text{ev}(\beta) = ab[\begin{smallmatrix} c \\ d \end{smallmatrix}]$. 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 $v, x, y, z, v', w', x', z' \in X[\varepsilon]$, $e \in X[c]$, $f \in X[d]$, $g' \in X[a]$, $h' \in X[b]$, $q \in X[[\begin{smallmatrix} a \\ b \end{smallmatrix}]]$, $r' \in X[[\begin{smallmatrix} c \\ d \end{smallmatrix}]]$. 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 $\text{ev}(\bar{\alpha}) = \text{ev}(\bar{\beta}) = abcd$ and $\bar{\alpha} \neq \bar{\beta}$.

Proof. The unique sparse step decomposition of $\begin{bmatrix} a \\ b \end{bmatrix} cd$ is

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

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

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

Proof of Prop. 17. Let X be an HDA with $L(X) = L$. We will show that there exist at least 2^n different sparse accepting paths accepting $(abcd)^n$. Let $P = \begin{bmatrix} a \\ b \end{bmatrix} cd$, $Q = ab \begin{bmatrix} c \\ d \end{bmatrix}$. For every sequence $\mathbf{R} = (R_1, \dots, R_n) \in \{P, Q\}^n$ let $\omega_{\mathbf{R}}$ be an accepting path such that $\text{ev}(\omega_{\mathbf{R}}) = R_1 * \dots * R_n$. By Lem. 9, there exist paths $\omega_{\mathbf{R}}^1, \dots, \omega_{\mathbf{R}}^n$ such that $\text{ev}(\omega_{\mathbf{R}}^k) = R_k$ and $\omega'_{\mathbf{R}} = \omega_{\mathbf{R}}^1 * \dots * \omega_{\mathbf{R}}^n$ is an accepting path. Let $\bar{\omega}_{\mathbf{R}}^k$ be the path defined as in Lem. 34 (*i.e.*, like $\bar{\alpha}$ if $R_k = P$ and $\bar{\beta}$ if $R_k = Q$). Finally, put $\bar{\omega}_{\mathbf{R}} = \bar{\omega}_{\mathbf{R}}^1 * \dots * \bar{\omega}_{\mathbf{R}}^n$.

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 Lem 34 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 $\text{ev}(\bar{\omega}_{\mathbf{R}}) = (abcd)^n$ for all \mathbf{R} . \blacktriangleleft

ST-automata. We prove here Lem. 20.

► **Lemma 20.** Φ respects boolean operations: for all $L_1, L_2 \in \mathcal{L}$, $\Phi(L_1 \cap L_2) = \Phi(L_1) \cap \Phi(L_2)$ and $\Phi(L_1 \cup L_2) = \Phi(L_1) \cup \Phi(L_2)$. Ψ respects regular operations: for all $W_1, W_2 \in \mathcal{W}$, $\Psi(W_1 \cup W_2) = \Psi(W_1) \cup \Psi(W_2)$, $\Psi(W_1 W_2) = \Psi(W_1) * \Psi(W_2)$, 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

$$\begin{aligned} \Psi(W_1) * \Psi(W_2) &= \{P_1 * \dots * P_n \mid P_1 \dots P_n \in W_1, \forall i : T_{P_i} = S_{P_{i+1}}\}\downarrow \\ &\quad * \{Q_1 * \dots * Q_m \mid Q_1 \dots Q_m \in W_2, \forall i : T_{Q_i} = S_{Q_{i+1}}\}\downarrow \\ &= \{P_1 * \dots * P_n * P_{n+1} * \dots * P_{n+m} \\ &\quad \mid P_1 \dots P_n \in W_1, P_{n+1} \dots P_{n+m} \in W_2, \forall i : T_{P_i} = S_{P_{i+1}}\}\downarrow \text{ by Lem 4} \\ &= \Psi(W_1 W_2). \end{aligned}$$

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

Complement. We prove here Lem. 25.

► **Lemma 25.** Let L and M be languages.

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