

Presenting Interval Pomsets with Interfaces

Amazigh Amrane¹, Hugo Bazille¹, Emily Clement², Uli Fahrenberg¹, and
Krzysztof Ziemiański³

¹ EPITA Research Laboratory (LRE), Paris, France

² Université Paris Cité, CNRS, IRIF, Paris, France

³ University of Warsaw, Poland

Abstract. Interval-order partially ordered multisets with interfaces (ipomsets) have shown to be a versatile model for executions of concurrent systems in which both precedence and concurrency need to be taken into account.

In this paper, we develop a presentation of ipomsets as generated by a graph of certain discrete ipomsets (starters and terminators) under the relation which composes subsequent starters and subsequent terminators. Using this presentation, we show that also subsumptions are generated by elementary relations. We develop a similar correspondence on the automata side, relating higher-dimensional automata, which generate ipomsets, and ST-automata, which generate step sequences, and their respective languages.

Keywords: pomset with interfaces, interval order, non-interleaving concurrency, higher-dimensional automaton

1 Introduction

Pomsets with interfaces, or *ipomsets* as they have come to be called, have recently emerged as a versatile model for executions of concurrent systems in which both precedence and concurrency need to be taken into account. Their first appearance was at RAMiCS 2020 in [8] which was based on the realization that the series-parallel pomsets, which are used heavily in concurrency theory [17–19, 23, 27], fail to model some rather simple concurrent executions. To overcome this, [8] introduced a generalization of the serial composition of pomsets, called *gluing*, which may continue events across compositions.

On the other hand, series-parallel pomsets have a nice algebraic characterization, given that they are the free models of concurrent semirings [3, 4, 16]. It is therefore natural to ask whether their generalization in [8] presents similarly nice algebraic properties. The definitive answer to that question is still out, but [8] and its successor paper [11] collect some evidence which weigh to the negative side: for example, the gluing composition of ipomsets is not cancellative, and some ipomsets may be decomposed both as gluing and as parallel compositions.

Another class of pomsets which are important in concurrency theory are interval orders [14, 15]. These are pomsets whose events may be represented as

intervals on the real line and have found their place in relativity [29], concurrency theory [20–22], and distributed computing [5, 24, 25]. Their algebraic theory, however, is less developed. Starting in [9], a notion emerged that for the first, ipomsets in their full generality may not be needed for concurrency but interval orders suffice, and secondly, that ipomsets might provide a suitable algebraic theory for interval orders. Picking up on ideas in [1, 2, 12] and based on the antichain representations of [21], the purpose of this paper is to develop such an algebraic theory of interval orders.

This paper is organized as follows. In Section 2, we recall pomsets with interfaces and a special subclass of starters and terminators. We then show that the category of interval-order ipomsets with interfaces is isomorphic to a category freely generated by starters and terminators under a certain congruence \sim . This is the first major contribution of this paper. It ultimately builds on work of Janicki and Koutny in [21] and only holds because we work with interval orders. For general ipomsets the situation seems to be much more complicated [11].

In Section 3, we extend our algebraic treatment to *subsumptions* of ipomsets. Subsumption is an important notion in concurrency theory [9, 13, 16] which frequently reasons about models and languages which are closed under subsumptions. As our second major contribution, we show that subsumptions of interval-order ipomsets are freely generated by elementary transpositions of starters and terminators up to \sim .

In Section 4 we extend our results to the operational side. We recall higher-dimensional automata (HDAs), whose languages are subsumption-closed sets of interval-order ipomsets, and introduce ST-automata, whose languages are sequences of starters and terminators under \sim . Precursors of ST-automata have been used in [1, 2, 6, 7, 10]; our third major contribution is to make the definition precise (and simpler) and expose the exact relation between ST-automata and HDAs. We provide translations in both directions, but only the translation from HDAs to ST-automata preserves languages (using the isomorphisms of the previous sections). The translation from ST-automata to HDAs introduces identifications and closures which imply that in the general case, the language of an ST-automaton is only included in that of its corresponding HDA. We leave open the problem whether there exists a syntactic restriction of ST-automata on which the translation preserves languages.

2 Ipomsets and Step Sequences

Let us first define pomsets with interfaces and step sequences. We fix an alphabet Σ , finite or infinite, throughout this paper.

2.1 Pomsets with interfaces

An *ipomset* (over Σ) is a structure $(P, <, \dashrightarrow, S, T, \lambda)$ consisting of the following:

- a finite set P of *events*;

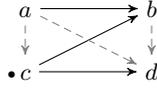


Fig. 1. An interval ipomset, *cf.* Ex. 1.

- a strict partial order (*i.e.*, an asymmetric, transitive and thus irreflexive relation) $< \subseteq P \times P$ called the *precedence order*;
- a strict partial order $--\rightarrow \subseteq P \times P$ called the *event order*;
- a subset $S \subseteq P$ called the *source set*;
- a subset $T \subseteq P$ called the *target set*; and
- a *labeling* $\lambda : P \rightarrow \Sigma$.

We require that

- the relation $< \cup --\rightarrow$ is total, *i.e.*, for all $x, y \in P$, at least one of $x = y$, $x < y$, $y < x$, $x --\rightarrow y$, or $y --\rightarrow x$ holds;
- events in S are $<$ -minimal in P , *i.e.*, for all $x \in S$ and $y \in P$, $y \not< x$; and
- events in T are $<$ -maximal in P , *i.e.*, for all $x \in T$ and $y \in P$, $x \not< y$.

We may add subscripts “ P ” to the elements above if necessary and omit any empty substructures from the signature. We will also often use the notation ${}_S P_T$ instead of $(P, <, --\rightarrow, S, T, \lambda)$ if no confusion may arise.

Example 1. Figure 1 depicts an ipomset $P = \{x_1, x_2, x_3, x_4\}$ with four events labelled $\lambda(x_1) = a$, $\lambda(x_2) = b$, $\lambda(x_3) = c$, and $\lambda(x_4) = d$. (We do not show the identity of events, only their labels.) Its precedence order is given by $x_1 < x_2$, $x_3 < x_2$ and $x_3 < x_4$ and its event order by $x_1 --\rightarrow x_3$, $x_1 --\rightarrow x_4$ and $x_2 --\rightarrow x_4$. The sources are $S = \{x_3\}$ and the targets $T = \emptyset$. (We denote these by “•”.) We think of events in S as being already active at the beginning of P , and the ones in T (here there are none) continue beyond the ipomset P .

An ipomset $(P, <, --\rightarrow, S, T, \lambda)$ is

- *discrete* if $<$ is empty (hence $--\rightarrow$ is total);
- a *pomset* if $S = T = \emptyset$;
- a *conclist* (short for “concurrency list”) if it is a discrete pomset;
- a *starter* if it is discrete and $T = P$;
- a *terminator* if it is discrete and $S = P$; and
- an *identity* if it is both a starter and a terminator.

The *source interface* of P is the conclist $S_P = (S, --\rightarrow_{1S \times S}, \lambda_{1S})$ where “ 1 ” denotes restriction; the *target interface* of P is the conclist $T_P = (T, --\rightarrow_{1T \times T}, \lambda_{1T})$. We call a starter or terminator *elementary* if $|S| = 1$, resp. $|T| = 1$, that is if it starts or terminates exactly one event.

An ipomset P is *interval* if $<_P$ is an interval order [15], *i.e.*, if it admits an interval representation given by functions $b, e : (P, <_P) \rightarrow (\mathbb{R}, <_{\mathbb{R}})$ such that

- $b(x) \leq_{\mathbb{R}} e(x)$ for all $x \in P$ and
- $x <_P y$ iff $e(x) <_{\mathbb{R}} b(y)$ for all $x, y \in P$.

That is, every element x of P is associated with a real interval $[b(x), e(x)]$ such that $x < y$ in P iff the interval of x ends before the one of y begins. The ipomset of Fig. 1 is interval. We will only treat interval ipomsets in this paper and thus omit the qualification “interval”.

The set of (interval) ipomsets is written iiPoms . We also denote

- by \square the set of conclists $U = (U, \dashrightarrow, \lambda)$;
- by St the set of starters ${}_S U_U = (U, \dashrightarrow, S, U, \lambda)$;
- by Te the set of terminators ${}_U U_T = (U, \dashrightarrow, U, T, \lambda)$;
- by $\text{Id} = \text{St} \cap \text{Te}$ the set of identities ${}_U U_U = (U, \dashrightarrow, U, U, \lambda)$;
- and let $\Omega = \text{St} \cup \text{Te}$, $\text{St}_+ = \text{St} \setminus \text{Id}$, and $\text{Te}_+ = \text{Te} \setminus \text{Id}$.

An *isomorphism* of ipomsets P and Q is a bijection $f : P \rightarrow Q$ for which

1. $f(S_P) = S_Q$; $f(T_P) = T_Q$; $\lambda_Q \circ f = \lambda_P$;
2. $f(x) <_Q f(y)$ iff $x <_P y$; and
3. $x \not<_P y$ and $y \not<_P x$ imply that $x \dashrightarrow_P y$ iff $f(x) \dashrightarrow_Q f(y)$.

We write $P \simeq Q$ if P and Q are isomorphic. The third axiom demands that if x and y are concurrent and hence ordered by \dashrightarrow_P , then f respects that order. Because of transitivity, event order may also appear between non-concurrent events; isomorphisms ignore such inessential event order.

Due to the requirement that all elements are ordered by $<$ or \dashrightarrow , there is at most one isomorphism between any two ipomsets. The following lemma is trivial but rather important; it states that we may always choose representatives in isomorphism classes such that isomorphisms become equalities.

Lemma 2. *For any ipomsets P and Q with $T_P \simeq S_Q$ there exists $Q' \simeq Q$ such that $T_P = S_{Q'} = P \cap Q'$. \square*

Let us recall the definition of the gluing operation of ipomsets.

Definition 3. *Let P and Q be two ipomsets with $T_P \simeq S_Q$. The gluing of P and Q is defined as $P * Q = (R, <, \dashrightarrow, S, T, \lambda)$ where:*

1. $R = (P \sqcup Q)_{x=f(x)}$, the quotient of the disjoint union under the unique isomorphism $f : T_P \rightarrow S_Q$;
2. $< = (\{(i(x), i(y)) \mid x <_P y\} \cup \{(j(x), j(y)) \mid x <_Q y\} \cup \{(i(x), j(y)) \mid x \in P \setminus T_P, y \in Q \setminus S_Q\})^+$, where $i : P \rightarrow R$ and $j : Q \rightarrow R$ are the injections and $^+$ denotes transitive closure;
3. $\dashrightarrow = \{(i(x), i(y)) \mid x \dashrightarrow_P y\} \cup \{(j(x), j(y)) \mid x \dashrightarrow_Q y\}$;
4. $S = i(S_P)$; $T = j(T_Q)$;
5. $\lambda(i(x)) = \lambda_P(x)$, $\lambda(j(x)) = \lambda_Q(x)$.

Remark 4. The relation \dashrightarrow is automatically transitive. On another note, composition is not cancellative: for example, $a * [\bullet_a^a] = a * [\bullet_a^a]$.

Gluing of isomorphic ipomsets are isomorphic. The next lemma extends Lem. 2 and follows directly from it.

Lemma 5. *For any ipomsets P and Q with $T_P \simeq S_Q$ there exist $Q' \simeq Q$ and $R \simeq P * Q$ such that $T_P = S_{Q'} = P \cap Q'$, $R = P \cup Q'$, $\langle_R = (\langle_P \cup \langle_{Q'} \cup (P \setminus Q') \times (Q' \setminus P))^+$, $\dashrightarrow_R = \dashrightarrow_P \cup \dashrightarrow_{Q'}$, $S_R = S_P$, and $T_R = T_{Q'}$. \square*

The following is clear and shown in [11].

Proposition 6. *Isomorphism classes of ipomsets form a category iiPoms_{\simeq} :*

- objects are isomorphism classes of conclists;
- morphisms in $\text{iiPoms}_{\simeq}(U, V)$ are isomorphism classes of ipomsets P with $S_P = U$ and $T_P = V$;
- composition of morphisms is gluing;
- identities are $\text{id}_U = {}_U U_U \in \text{iiPoms}_{\simeq}(U, U)$. \square

In analogy to iiPoms_{\simeq} we will also write Ω_{\simeq} , St_{\simeq} etc. for subsets of ipomsets-up-to-isomorphism.

2.2 Starters and terminators

We develop a representation of the category iiPoms_{\simeq} by generators and relations, using the step decompositions introduced in [12].

Let $\bar{\Omega}_{\simeq}$ be the directed multigraph given as follows:

- Vertices are isomorphism classes of conclists.
- Edges in $\bar{\Omega}_{\simeq}(U, V)$ are isomorphism classes of starters and terminators P with $S_P = U$ and $T_P = V$.

Note that for all $U, V \in \square_{\simeq}$, $\bar{\Omega}_{\simeq}(U, V) \subseteq \text{St}_{\simeq}$ or $\bar{\Omega}_{\simeq}(U, V) \subseteq \text{Te}_{\simeq}$.

Let Coh_{\simeq} be the free category generated by $\bar{\Omega}_{\simeq}$. Then non-identity morphisms in $\text{Coh}_{\simeq}(U, V)$ are words

$$P_1 \dots P_n \in \Omega_{\simeq}^+ \quad (1)$$

such that $T_{P_i} = S_{P_{i+1}}$ for all $i = 1, \dots, n-1$. Such words are called *coherent* in [2]. Note that $P_1 \dots P_n$ is coherent iff the gluing $P_1 * \dots * P_n$ is defined.

The following property is immediate from Lemma 5. It permits to choose the representant of a coherent word so that events overlap and will often be used implicitly in the following.

Lemma 7. *For every $U, V \in \square_{\simeq}$ and every non-identity morphism $P_1 \dots P_n \in \text{Coh}_{\simeq}(U, V)$ there is $Q_1 \dots Q_n \in \text{Coh}_{\simeq}(U, V)$ such that $P_1 = Q_1$ and for all $i = 2, \dots, n$, $P_i \simeq Q_i$ and $T_{Q_{i-1}} = S_{Q_i} = Q_{i-1} \cap Q_i$. \square*

Let \sim be the congruence on Coh_{\simeq} generated by the relations Z

$$\begin{aligned} PQ &\sim P * Q && (P, Q \in \text{St} \text{ or } P, Q \in \text{Te}), \\ \text{id}_U &\sim {}_U U_U && (U \in \square). \end{aligned} \tag{2}$$

The first of these allows to compose subsequent starters and subsequent terminators, and the second identifies the (freely generated) identities at U with the corresponding ipomset identities in Id . (Note that the gluing of two starters is again a starter, and similarly for terminators; but “mixed” gluings do not have this property.) It is clear that \sim is compatible with ipomset isomorphism. We let Coh_{\sim} denote the quotient of Coh_{\simeq} under \sim .

Let $\bar{\Psi} : \text{Coh}_{\simeq} \rightarrow \text{iiPoms}_{\simeq}$ be the functor induced by the inclusion:

$$\bar{\Psi}(U) = U, \quad \bar{\Psi}(P) = P.$$

Then $\bar{\Psi}(P_1 \dots P_n) = P_1 * \dots * P_n$. The following is straightforward.

Lemma 8. *If $P_1 \dots P_n \sim Q_1 \dots Q_m$, then $\bar{\Psi}(P_1 \dots P_n) = \bar{\Psi}(Q_1 \dots Q_m)$. \square*

Thus $\bar{\Psi}$ induces a functor $\Psi : \text{Coh}_{\sim} \rightarrow \text{iiPoms}_{\simeq}$; we show below that Ψ is an isomorphism of categories.

2.3 Step sequences

A *step sequence* [1] is a morphism in Coh_{\sim} , that is, an equivalence class of coherent words under \sim . It is shown in [12] that every ipomset may be decomposed into a step sequence:

Lemma 9 ([12]). *For every $P \in \text{iiPoms}_{\simeq}$ there exists $w \in \text{Coh}_{\sim}$ such that $\Psi(w) = P$.*

A word $P_1 \dots P_n \in \text{Coh}_{\simeq}$ is *dense* if all its elements are elementary, *i.e.*, start or terminate precisely one event. It is *sparse* if starters and terminators are alternating, that is, for all $i = 1, \dots, n-1$, $(P_i, P_{i+1}) \in (\text{St}_+ \times \text{Te}_+) \cup (\text{Te}_+ \times \text{St}_+)$. By convention, identities $\text{id}_U \in \text{Coh}_{\simeq}$ are both dense and sparse.

Lemma 10 ([12]). *Every step sequence contains exactly one sparse representative.*

Showing existence of sparse decompositions is easy and consists of gluing starters and terminators until no more such gluing is possible. Showing uniqueness is more tedious, see [12].

Example 11. The unique sparse step decomposition of the ipomset in Fig. 1 is

$$P = \begin{bmatrix} a \bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ d \bullet \end{bmatrix} \begin{bmatrix} \bullet a \\ \bullet d \bullet \end{bmatrix} \begin{bmatrix} b \bullet \\ \bullet d \bullet \end{bmatrix} \begin{bmatrix} \bullet b \\ \bullet d \end{bmatrix} :$$

it first starts a , then terminates c , then starts d and terminates a , then starts b and finally terminates both b and d . The dense step decompositions of P are

$$\begin{aligned} P &= \begin{bmatrix} a\bullet \\ \bullet c\bullet \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ d\bullet \end{bmatrix} \begin{bmatrix} \bullet a \\ \bullet d\bullet \end{bmatrix} \begin{bmatrix} b\bullet \\ \bullet d\bullet \end{bmatrix} \begin{bmatrix} \bullet b \\ \bullet d\bullet \end{bmatrix} [\bullet d] \\ &= \begin{bmatrix} a\bullet \\ \bullet c\bullet \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ d\bullet \end{bmatrix} \begin{bmatrix} \bullet a \\ \bullet d\bullet \end{bmatrix} \begin{bmatrix} b\bullet \\ \bullet d\bullet \end{bmatrix} \begin{bmatrix} \bullet b \\ \bullet d \end{bmatrix} [\bullet b], \end{aligned}$$

which differ only in the order in which b and d are terminated at the end.

Using Lemmas 9 and 10 we may now define a functor $\Phi : \text{iiPoms}_{\simeq} \rightarrow \text{Coh}_{\sim}$ which will serve as inverse to Ψ : for $P \in \text{iiPoms}$ let $w \in \text{Coh}_{\simeq}$ be its unique sparse step decomposition and put $\Phi(P) = [w]_{\sim}$.

Theorem 12. Φ is a functor, $\Psi \circ \Phi = \text{Id}_{\text{iiPoms}_{\simeq}}$, and $\Phi \circ \Psi = \text{Id}_{\text{Coh}_{\sim}}$. Hence $\Phi : \text{iiPoms}_{\simeq} \xrightarrow{\sim} \text{Coh}_{\sim} : \Psi$ is an isomorphism of categories.

Proof. We have $\Phi(P * Q) = \Phi(P)\Phi(Q)$ by definition of \sim , and the other claims follow. \square

Corollary 13. The category iiPoms_{\simeq} is freely generated by the directed multi-graph $\bar{\Omega}_{\simeq}$ using gluing composition.

3 Subsumptions in Step Sequences

A *subsumption* of ipomsets P and Q is a bijection $f : P \rightarrow Q$ for which

1. $f(S_P) = S_Q$; $f(T_P) = T_Q$; $\lambda_Q \circ f = \lambda_P$;
2. $f(x) <_Q f(y)$ implies $x <_P y$; and
3. $x \not<_P y$, $y \not<_P x$, and $x \dashrightarrow_P y$ imply $f(x) \dashrightarrow_Q f(y)$.

We write $P \sqsubseteq Q$ if there is a subsumption $f : P \rightarrow Q$ and $P \sqsubset Q$ if $P \sqsubseteq Q$ and $P \not\cong Q$. Thus, subsumptions preserve interfaces and labels but may remove precedence order and add essential event order. Isomorphisms of ipomsets are precisely invertible subsumptions.

In this section we extend the equivalence between ipomsets and step sequences from Thm. 12 to also cover subsumptions.

Definition 14. A subsumption $f : P \rightarrow Q$ is elementary if there is a unique pair $x, y \in P$ such that $x <_P y$ and $f(x) \not<_Q f(y)$.

We show that subsumptions are generated by elementary ones. To this end, let $P \sqsubset Q$ with witness $f : P \rightarrow Q$. Define $O = <_P \setminus <_Q$ (we tacitly identify $<_P$ with its image in Q by f), and let $\preceq \subseteq O \times O$ be the relation given by $(x, y) \preceq (x', y')$ if $x \leq_P x'$ and $y \geq_P y'$.

Lemma 15. \preceq is a partial order on O and has a minimal element (x_0, y_0) . Let R be defined as P , but with $<_R = <_P \setminus \{(x_0, y_0)\}$ and $\dashrightarrow_R = \dashrightarrow_Q$, then $\text{id} : P \rightarrow R$ is an elementary subsumption and $f : R \rightarrow Q$ a subsumption.

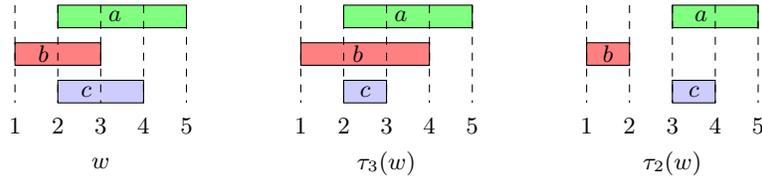


Fig. 2. Interval representations of several ipomsets, cf. Ex. 17.

Proof. Reflexivity, anti-symmetry and transitivity of \preceq come from \leq_P . As O is finite, there exists a minimal element.

For the second part, we start by proving that R is an ipomset. S , T and λ all come from P , and $<_{R \cup \dashrightarrow R}$ is total because $(x_0, y_0) \in O$. All that is left is to prove is that $<_R$ is a strict partial order.

Irreflexivity and anti-symmetry are given by $<_P$. For transitivity, let $x <_R y$ and $y <_R z$. Then $x <_P z$ and we have to show that either $x \neq x_0$ or $z \neq y_0$. Assuming $(x, z) = (x_0, y_0)$, there would be y such that $x_0 <_R y <_R y_0$ which contradicts the minimality of (x_0, y_0) in the order \preceq .

Finally, $\text{id} : P \rightarrow R$ and $f : R \rightarrow Q$ clearly are subsumptions, and the first is elementary because $|<_P \setminus <_R| = 1$. \square

Let us now introduce some useful notation for starters and terminators to more clearly specify the conclists of events which are started or terminated. For $U \in \square$ and $A, B \subseteq U$ we write $A \uparrow U = U \setminus A \downarrow U$ for the starter which starts the events in A and $U \downarrow_B = U \setminus B \uparrow$ for the terminator which terminates the events in B . In the following definition we express the result of transpositions of elements of a coherent word $w = P_1 \dots P_n \in \text{Coh}_{\preceq}$.

Definition 16. Let $U \in \square$ with $A, B \subseteq U$ and $A \cap B = \emptyset$. Let $w = P_1 \dots P_n \in \text{Coh}_{\preceq}$ and $i \in \{1, \dots, n-1\}$. The i -th transposition on w , that we denote $\tau_i(w)$, is equal to $P_1 \dots P_{i-1} P'_i P'_{i+1} P_{i+2} \dots P_n$, with $P'_i P'_{i+1} =$

$$\begin{array}{ll}
 B \uparrow (U \setminus A) A \uparrow U & \text{if } P_i = A \uparrow (U \setminus B) \text{ and } P_{i+1} = B \uparrow U, \\
 U \downarrow_B (U \setminus B) \downarrow_A & \text{if } P_i = U \downarrow_A \text{ and } P_{i+1} = (U \setminus A) \downarrow_B, \\
 (U \setminus A) \downarrow_B A \uparrow (U \setminus B) & \text{if } P_i = A \uparrow U \text{ and } P_{i+1} = U \downarrow_B, \\
 B \uparrow U U \downarrow_A & \text{if } P_i = (U \setminus B) \downarrow_A \text{ and } P_{i+1} = B \uparrow (U \setminus A).
 \end{array}$$

That is, $\tau_i(w)$ swaps the i th and $(i+1)$ th element of w , but takes care of adjusting them to preserve coherency between the start and ending of events.

Example 17. Figure 2 presents several interval representations of ipomsets. Intuitively, the left one is a depiction of

$$w = [b \bullet] \begin{bmatrix} a \bullet \\ \bullet b \bullet \\ c \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \\ \bullet c \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet c \bullet \end{bmatrix} [\bullet a].$$

By swapping the third and fourth elements (corresponding here to the second case of the definition), we obtain

$$\tau_3(w) = [b\bullet] \begin{bmatrix} a\bullet \\ \bullet b\bullet \\ c\bullet \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ \bullet b\bullet \\ \bullet c \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ \bullet b \end{bmatrix} [\bullet a],$$

represented in the middle of the figure. If we instead apply the transformation τ_2 (third case of the definition), we obtain

$$\tau_2(w) = [b\bullet] [\bullet b] \begin{bmatrix} a\bullet \\ c\bullet \end{bmatrix} \begin{bmatrix} \bullet a\bullet \\ \bullet c \end{bmatrix} [\bullet a]$$

shown in the right part of the figure. Note that $\Psi(w) = \Psi(\tau_3(w))$ but $\Psi(w) \sqsubset \Psi(\tau_2(w))$.

Remark 18. In the context of HDAs, [28] defines a notion of *adjacency* for paths which consists of precisely the analogues of the transformations that we define above. Adjacency is then used to define *homotopy* of paths, whereas we use it to define subsumptions. On paths, homotopy is the symmetric and transitive closure of subsumption.

Our goal now is to prove that the transpositions defined above are exactly what is needed to express the notion of subsumption on step sequences. This will be done in Thm. 25, but first we need some technical lemmas.

First we treat the first two cases in Def. 16, swapping starters or swapping terminators:

Lemma 19. *Let $w = P_1 \dots P_n \in \text{Coh}_{\simeq}$. If $P_i, P_{i+1} \in \text{St}$ or $P_i, P_{i+1} \in \text{Te}$, then $\tau_i(w) \sim w$.*

Proof. Let $P_i = {}_A\uparrow U'$ and $P_{i+1} = {}_B\uparrow U$. As w is coherent, $U = U' \cup B$ and $A \cap B = \emptyset$. Then the lemma follows from $P_i * P_{i+1} = {}_{A \cup B}\uparrow U = {}_B\uparrow (U' \setminus A) * {}_A\uparrow U$. The arguments are similar in the case of successive terminators. \square

Lemma 20. *Let $w_1, w_2 \in \text{Coh}_{\simeq} \cap \text{St}^*$, resp. $w_1, w_2 \in \text{Coh}_{\simeq} \cap \text{Te}^*$, be dense. Then $w_1 \sim w_2$ iff there exist dense words $u_1, \dots, u_m \in \text{Coh}_{\simeq} \cap \text{St}^*$, resp. $u_1, \dots, u_m \in \text{Coh}_{\simeq} \cap \text{Te}^*$, for some $m \geq 0$, and indices $i_1, \dots, i_{m-1} \in \{1, \dots, n-1\}$ such that $u_1 = w_1$, $u_m = w_2$ and, for all k , $u_{k+1} = \tau_{i_k}(u_k)$.*

Proof. The direction from right to left follows from Lem. 19. For the converse, write $w_1 = P_1 \dots P_n$ and $w_2 = Q_1 \dots Q_n$ and assume that $w_1, w_2 \in \text{St}^*$. Let, for all $i \leq n$, x_i be the event started in Q_i and in the i_j th starter of u_k . Set $u_{k+1} = \tau_i \circ \dots \circ \tau_{i_{j-1}}(u_k)$. Then $u_m = w_2$. The arguments are similar for $w_1, w_2 \in \text{Te}^*$. \square

Lemma 21. *Let $w_1 = P_1 \dots P_n$, $w_2 = Q_1 \dots Q_n \in \text{Coh}_{\simeq}$ be dense. Then $w_1 \sim w_2$ iff there exist dense words $u_1, \dots, u_m \in \text{Coh}_{\simeq}$, for some $m \geq 0$, and indices $i_1, \dots, i_{m-1} \in \{1, \dots, n-1\}$ such that $u_1 = w_1$, $u_m = w_2$ and, for all k , $u_{k+1} = \tau_{i_k}(u_k)$.*

Proof. The direction from right to left follows again from Lem. 19. For the converse, let $w = R_1 \dots R_l$ be a sparse coherent word such that $w \sim w_1 \sim w_2$. Note that $l \leq n$. For all $i \leq l$, there exist factors $w_{1,i} = P_{i,1} \dots P_{i,j_i}$ and $w_{2,i} = Q_{i,1} \dots Q_{i,j'_i}$ of respectively w_1 and w_2 such that $w_{1,i} \sim w_{2,i} \sim R_i$. In addition, if R_i is a starter (resp. terminator) then $w_{1,i}$ and $w_{2,i}$ are sequences of starters (resp. terminators). The lemma now follows using Lem. 20. \square

Now we treat the last two cases in Def. 16 which swap starters with terminators or terminators with starters:

Lemma 22. *Let $w = P_1 \dots P_n \in \text{Coh}_{\simeq}$. If $P_i P_{i+1} = A \uparrow U U' \downarrow_B$ with $A \cap B = \emptyset$, then $\Psi(\tau_i(w)) \sqsubset \Psi(w)$. If $P_i P_{i+1} = U \downarrow_B A \uparrow U'$, then $\Psi(w) \sqsubset \Psi(\tau_i(w))$.*

Proof. Since w is coherent, we have $U = U'$. Let $P'_i P'_{i+1} = (U \setminus A) \downarrow_B A \uparrow (U \setminus B)$. Then the lemma follows from $P'_i * P'_{i+1} \sqsubset P_i * P_{i+1}$. Indeed, in $P'_i * P'_{i+1}$ the elements of B must precede those of A , while they are concurrent in $P_i * P_{i+1}$. As to the second claim, if $P_i P_{i+1} = U \downarrow_B A \uparrow U'$, then $U' \setminus A = U \setminus B$, and the events of B precede those of A in $P_i * P_{i+1}$, while they are concurrent in ${}_B \uparrow U * U \downarrow_A$. \square

Lemma 23. *Let $f : P \rightarrow Q$ be a subsumption and $P = P_1 * \dots * P_n$, $Q = Q_1 * \dots * Q_n$ dense step decompositions. There exist dense words $u_1, \dots, u_m \in \text{Coh}_{\simeq}$ for some $m \geq 0$, and indices $i_1, \dots, i_{m-1} \in \{1, \dots, n-1\}$ such that $u_1 = w_1 = P_1 \dots P_n$, $u_m = w_2 = Q_1 \dots Q_n$ and, for all k , $u_{k+1} = \tau_{i_k}(u_k)$.*

Proof. We proceed by induction on $ar = |\langle_P \setminus \langle_Q|$ (we again tacitly identify \langle_P with its image in Q by f). For $ar = 0$ the claim is clear.

If $ar = 1$, then exists exactly one pair $(x, y) \in P \times P$ such that $x \langle_P y$ and, without loss of generality, $x \dashrightarrow_Q y$. This means that x must be terminated and y started in both P and Q . Moreover, there exists $P'_1 \dots P'_n \sim P_1 \dots P_n$ and i, U, U' such that $P'_i = U \downarrow_{\{x\}}$ and $P'_{i+1} = \{y\} \uparrow U'$. Else, there would exist z, w (not necessarily distinct) such that $x \langle z$, $w \langle y$, $z \not\prec w$ and $w \not\prec z$. Then, removing the order between x and y would force to remove either between x and z or w and y , which contradicts the assumption. We also have that $\tau_i(P'_1 \dots P'_n) \sim Q_1 \dots Q_n$. Using Lem. 21, there are g, h such that $Q_1 \dots Q_n = h \circ \tau_i \circ g(P_1 \dots P_n)$ and every transposition respects the condition in the lemma.

If $ar > 1$, then let (x_0, y_0) and P' be as defined in Lem. 15, with $P' = P'_1 \dots P'_n$ any dense decomposition. Using the base case, there exists a transformation from $P_1 \dots P_n$ to $P'_1 \dots P'_n$. Then, $|\langle_{P'} \setminus \langle_Q| = ar - 1$, and using the induction hypothesis allows us to conclude there exists a transformation from $P'_1 \dots P'_n$ to $Q_1 \dots Q_n$. By composing those two, we obtain a transformation from $P_1 \dots P_n$ to $Q_1 \dots Q_n$, hence the result. \square

Example 24. Let $P = ab$, $Q = \begin{bmatrix} a \\ b \end{bmatrix}$ and $a \bullet ab \bullet b$ and $b \bullet \begin{bmatrix} a \bullet \\ \bullet b \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \bullet a$ be dense step decompositions of P resp. Q . An example of a sequence as in Lem. 23 is

$$\begin{aligned} w_0 &= a \bullet ab \bullet b, \\ w_1 &= a \bullet \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \bullet b, \\ w_2 &= a \bullet \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \bullet a, \\ w_3 &= b \bullet \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \begin{bmatrix} \bullet a \bullet \\ \bullet b \bullet \end{bmatrix} \bullet a, \end{aligned}$$

with $i_1 = 2$, $i_2 = 3$, and $i_3 = 1$.

Let $<_e$ be the relation on Coh_{\simeq} defined by $w_1 <_e w_2$ if there is an index i such that $w_2 = \tau_i(w_1)$ and $P_i P_{i+1} = A \uparrow U U' \downarrow_B$ with $A \cap B = \emptyset$ (third case of Def. 16). Denote by the same symbol $<_e$ the relation induced in the quotient Coh_{\sim} and by $\leq = <_e^*$ the reflexive, transitive closures.

Theorem 25. *For $P_1, P_2 \in \text{iiPoms}_{\simeq}$, $P_1 \sqsubseteq P_2$ iff $\Phi(P_1) \leq \Phi(P_2)$. For $w_1, w_2 \in \text{Coh}_{\sim}$, $w_1 \leq w_2$ iff $\Psi(w_1) \sqsubseteq \Psi(w_2)$.*

Proof. By Lemmas 22 and 23. □

Corollary 26. *Subsumptions of ipomsets are freely generated by the relation $<_e$.*

4 Higher-Dimensional Automata and ST-Automata

We now transfer the isomorphisms of the previous sections 2 and 3 to the operational side. We recall higher-dimensional automata which generate ipomsets and introduce ST-automata which generate step sequences, and we clarify their relation.

4.1 Higher-dimensional automata

We give a quick introduction to higher-dimensional automata and their languages and refer the interested reader to [2, 12] for details and examples.

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 $\text{ev} : X \rightarrow \square$ which to every cell assigns a conlist of concurrent events which are active in it. We write $X[U] = \{q \in X \mid \text{ev}(q) = U\}$ for the cells of type U . For every $U \in \square$ and $A \subseteq U$ there are *face maps* $\delta_A^0, \delta_A^1 : X[U] \rightarrow X[U \setminus A]$ (we often omit the extra subscript U) which satisfy

$$\delta_A^\nu \delta_B^\mu = \delta_B^\mu \delta_A^\nu \text{ for } A \cap B = \emptyset \text{ and } \nu, \mu \in \{0, 1\}. \quad (3)$$

The *upper* face maps δ_A^1 terminate events in A and the *lower* face maps δ_A^0 transform a cell q into one in which the events in A have not yet started.

A *higher-dimensional automaton (HDA)* $X = (X, \perp, \top)$ is a precubical set together with subsets $\perp, \top \subseteq X$ of *start* and *accept* cells. Note that we do not assume HDAs to be finite here; finiteness is needed when reasoning about regular languages, see [10, 12], but we will not do that here. See Fig. 3 for an example.

A *path* in an HDA X is a sequence $\alpha = (q_0, \varphi_1, q_1, \dots, \varphi_n, q_n)$ consisting of cells $q_i \in X$ and symbols φ_i which indicate face map types: for every $i = 1, \dots, n$, $(q_{i-1}, \varphi_i, q_i)$ is either

- $(\delta_A^0(q_i), \nearrow^A, q_i)$ for $A \subseteq \text{ev}(q_i)$ (an *upstep*) or

- $(q_{i-1}, \downarrow_A, \delta_A^1(q_{i-1}))$ for $A \subseteq \text{ev}(q_{i-1})$ (a *downstep*).

The *source* and *target* of α as above are $\text{src}(\alpha) = q_0$ and $\text{tgt}(\alpha) = q_n$, and α is *accepting* if $\text{src}(\alpha) \in \perp$ and $\text{tgt}(\alpha) \in \top$. Paths α and β may be *concatenated* to $\alpha * \beta$ if $\text{tgt}(\alpha) = \text{src}(\beta)$.

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

- if $\alpha = (q)$, then $\text{ev}(\alpha) = \text{id}_{\text{ev}(q)}$;
- if $\alpha = (q \uparrow^A p)$, then $\text{ev}(\alpha) = A \uparrow \text{ev}(p)$;
- if $\alpha = (p \downarrow_B q)$, then $\text{ev}(\alpha) = \text{ev}(p) \downarrow_B$;
- if $\alpha = \alpha_1 * \dots * \alpha_n$ is a concatenation, then $\text{ev}(\alpha) = \text{ev}(\alpha_1) * \dots * \text{ev}(\alpha_n)$.

The *language* of an HDA X is

$$L(X) = \{[\text{ev}(\alpha)]_{\simeq} \mid \alpha \text{ accepting path in } X\} \subseteq 2^{\text{iiPoms}_{\simeq}}.$$

Languages of HDAs are closed under subsumption [10]: whenever $P \sqsubseteq Q \in L(X)$, also $P \in L(X)$.

4.2 ST-automata

Definition 27. An ST-automaton is a structure $A = (Q, E, I, F, \lambda)$ consisting of sets Q , $E \subseteq Q \times \Omega_{\simeq} \times Q$, $I, F \subseteq Q$, and a function $\lambda : Q \rightarrow \square$ such that for all $(q, sU_T, r) \in E$, $\lambda(q) = S$ and $\lambda(r) = T$.

This is thus a plain automaton over Ω_{\simeq} (finite or infinite) with an additional labeling of states with conclists that is consistent with the labeling of edges. (But note that the alphabet Ω_{\simeq} is infinite.)

Remark 28. Equivalently, an ST-automaton may be defined as a directed multi-graph G together with a graph morphism $\text{ev} : G \rightarrow \bar{\Omega}_{\simeq}$ and initial and final states I and F . This definition would be slightly more general than the one above, given that it allows for multiple edges with the same label between the same pair of states.

A *path* in an ST-automaton A is defined as usual: an alternating sequence $\pi = (q_0, e_1, q_1, \dots, e_n, q_n)$ of states q_i and transitions e_i such that $e_i = (q_{i-1}, P_i, q_i)$ for every $i = 1, \dots, n$ and a sequence $P_1, \dots, P_n \in \Omega_{\simeq}$. The path is *accepting* if $q_0 \in I$ and $q_n \in F$. The *label* of π as above is $\ell(\pi) = [\text{id}_{\lambda(q_0)} P_1 \text{id}_{\lambda(q_1)} \dots P_n \text{id}_{\lambda(q_n)}]_{\sim}$, the equivalence class under \sim .

The *language* of an ST-automaton A is

$$L(A) = \{\ell(\pi) \mid \pi \text{ accepting path in } A\} \subseteq 2^{\text{Coh}_{\sim}}.$$

Contrary to languages of HDAs, languages of ST-automata may not be closed under subsumption, see below.

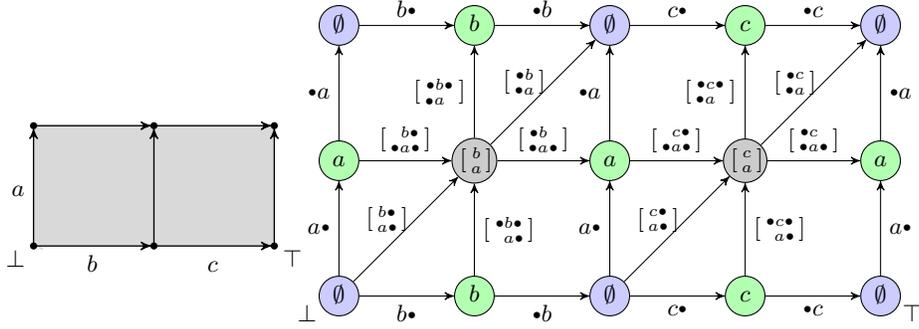


Fig. 3. Two-dimensional HDA X (left) and corresponding ST-automaton $F(X)$ (right).

4.3 From HDAs to ST-automata

We now define translations between HDAs and ST-automata. In order to relate them to their languages, we extend the pair of functors $\Phi : \text{iiPoms}_{\sim} \rightleftarrows \text{Coh}_{\sim} : \Psi$ to the power sets the usual way:

$$\Phi(A) = \{\Phi(P) \mid P \in A\}, \quad \Psi(B) = \{\Psi(w) \mid w \in B\}.$$

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

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

That is, the transitions of $F(X)$ precisely mimic the starting and terminating of events in X . (Note that lower faces in X are inverted to get the starting transitions.)

Example 29. Figure 3 shows an HDA X with $L(X) = \{bc\}$ together with its translation to an ST-automaton $F(X)$.

Theorem 30. *For any HDA X , $L(F(X)) = \Phi(L(X))$.*

Proof. For identities note that a path with a single cell q is accepting in X if and only if it is accepting in $F(X)$, and $\Phi(\text{id}_{\text{ev}(q)}) = [\text{id}_{\lambda(q)}]_{\sim}$. Now let $w = P_1 \dots P_m \in L(F(X))$ be a non-identity. By definition, there exists $\pi = (q_0, e_1, q_1, \dots, e_n, q_n)$ where $e_i = (q_{i-1}, P'_i, q_i)$, $P'_i \in \Omega_{\sim}$ such that $\text{id}_{\lambda(q_0)} P'_1 \text{id}_{\lambda(q_1)} \dots P'_n \text{id}_{\lambda(q_n)} \sim w$. This means that $P'_1 * \dots * P'_n$ is a decomposition of some $P \in L(X)$, hence $P \in \Phi(L(X))$.

For the converse, let $w = P_1 \dots P_m \in \Phi(L(X))$. Let $P'_1 * \dots * P'_n$ be the sparse step decomposition of $P = P_1 * \dots * P_m$. We have $P'_1 \dots P'_n \sim w$. In addition, there exists an accepting path $\alpha = \beta_1 * \dots * \beta_n$ in X such that $\text{ev}(\beta_i) = P'_i$. By construction there exists an accepting path $\pi = (\text{src}(\beta_1), e_1, \text{tgt}(\beta_1), \dots, e_n, \text{tgt}(\beta_n))$ in $F(X)$ where $e_i = (\text{src}(\beta_i), P'_i, \text{tgt}(\beta_i))$. We have $\ell(\pi) \sim w$. \square

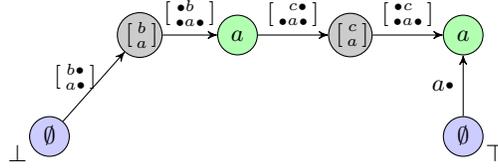


Fig. 4. An ST-automaton with missing face relations, *cf.* Ex. 31.

4.4 From ST-automata to HDAs

Let $A = (Q, E, I, F, \lambda)$ be an ST-automaton, then we define an HDA $G(A) = (X, \perp, \top)$. Ideally, the cells of the precubical set X would be the states of A and $\text{ev} = \lambda$; but this does not quite work as the result is not necessarily a precubical set. The difficulty is in the face maps which we first define as *relations* on Q : For every $U \in \square$ and $A \subseteq U$, let

$$\delta_{A,U}^0 = \{(q, p) \mid (p, A \uparrow U, q) \in E\}, \quad \delta_{A,U}^1 = \{(q, r) \mid (q, U \downarrow_A, r) \in E\}.$$

(Hence, starting transitions in A are inverted to get lower face relations.) Now there are three problems which may appear: the so-defined face relations may not be *total* (*i.e.*, undefined for some faces), they may not be *functional* (*i.e.*, multi-valued for some faces), and they may not satisfy the precubical identities (3).

Example 31. Let A be the ST-automaton of Fig. 4. Then $L(A) = \emptyset$ as there are no paths from I to F . Translating A to an HDA will require adding all faces which are shown on the right of Fig. 3.

The first problem is easily solved, as we may just freely add missing faces. Given that we address functionality afterwards, we can as well freely add *all* faces, so define

$$\bar{Q} = \{(q, B, C) \mid q \in Q, B, C \subseteq \text{ev}(q), B \cap C = \emptyset\},$$

with $\bar{\text{ev}} : \bar{Q} \rightarrow \square$ given by $\bar{\text{ev}}((q, B, C)) = \text{ev}(q) \setminus (B \cup C)$ and

$$\begin{aligned} \bar{\delta}_{A,U}^0 &= \{((q, B, C), (r, B, C)) \mid (q, r) \in \delta_{A,U}^0, A \subseteq U \setminus (B \cup C)\} \\ &\quad \cup \{((q, B, C), (q, B \cup A, C)) \mid A \cap C = \emptyset\}, \\ \bar{\delta}_{A,U}^1 &= \{((q, B, C), (r, B, C)) \mid (q, r) \in \delta_{A,U}^1, A \subseteq U \setminus (B \cup C)\} \\ &\quad \cup \{((q, B, C), (q, B, C \cup A)) \mid A \cap B = \emptyset\}. \end{aligned}$$

That is, each existing cell $q \in Q$ is associated with $(q, \emptyset, \emptyset)$ and (q, B, C) is its (formal) face where B is unstarted and C terminated. Existing face maps $(q, r) \in \delta_{A,U}^\nu$ are copied to each pair $((q, B, C), (r, B, C))$ for which $A \subseteq U \setminus (B \cup C)$.

In order to solve the second problem, we need to identify some elements of \bar{Q} with others. Let \sim be the equivalence relation on \bar{Q} generated by

$$q \sim q' \wedge \exists A \subseteq U \in \square, \nu \in \{0, 1\} : (q, r), (q', r') \in \bar{\delta}_{A,U}^\nu \implies r \sim r',$$

let $Q' = \bar{Q}_\sim$ be the quotient, and let $\delta_{A,U}^0, \delta_{A,U}^1$ be the face relations induced on Q' . These are now single-valued and total, *i.e.*, functions. Given that $q \sim q'$ implies $\bar{e}v(q) = \bar{e}v(q')$, also $\bar{e}v$ passes to the quotient.

Lastly, we need to make sure that the precubical identities (3) are satisfied. This may again be done by defining an equivalence relation, again denoted \sim , on Q' , which identifies faces which should be equal according to (3). Let $X = Q'_\sim$ be the quotient, and let $e\bar{v}, \delta_{A,U}^0, \delta_{A,U}^1$ be the mappings induced on X . Then X is a precubical set, and we may define $\perp, \top \subseteq X$ to be the equivalence classes of $I, F \subseteq Q$.

Example 32. Continuing Ex. 31, let again A be the ST-automaton of Fig. 4. Then $G(A)$ is the HDA of Fig. 3 and $L(G(A)) = \{bc\} \neq \emptyset = L(A)$.

Remark 33. The above translation from ST-automata to HDAs may be understood as a colimit in a presheaf category. HDAs are presheaves over a *precube* category, see [10]. Using Rem. 28, we may view an ST-automaton A as a morphism into a variant of $\bar{\Omega}_\sim$ which may be embedded into that precube category (inverting starters in the process), and then $G(A)$ is the colimit closure of the composition of these morphisms with the Yoneda embedding.

Theorem 34. *For any ST-automaton A , $\Psi(L(A)) \subseteq L(G(A))$.*

Proof. Let $\pi = (q_0, e_1, q_1, \dots, e_n, q_n)$, $e_i = (q_{i-1}, P_i, q_i)$ be an accepting path in A . Then $\alpha = (q_0, \varphi_1, q_1, \dots, \varphi_n, q_n)$, where

- $\varphi_i = \nearrow^A$ if $P_i = A \uparrow U$ is a starter,
- $\varphi_i = \searrow_B$ if $P_i = U \downarrow_B$ is a terminator,

is an accepting path in $G(A)$. Furthermore,

$$\Psi(\lambda(\pi)) = \Psi(P_1 P_2 \cdots P_n) = P_1 * \cdots * P_n = \lambda(\alpha) \in L(G(A)),$$

which concludes the proof. □

Example 32 shows that the inclusion in the theorem may be strict. It is clear that for any HDA X , $G(F(X)) = X$; but as we have seen, $F(G(A))$ may be very different from a given ST-automaton A . The following lemma collects a few properties of ST-automata which are in the image of the HDA translation.

Lemma 35. *Let $A = F(X)$ for some HDA X . Then $A = (Q, E, I, F, \lambda)$ has the following properties:*

1. for all $q \in Q$ with $\lambda(q) = \text{id}_U$ and all $S, T \subseteq U$, there exist $p, r \in Q$ such that $(p, S U_U, q), (q, U U_T, r) \in E$;
2. for all $(p, P, q), (q, Q, r) \in E$, if $P, Q \in \text{St}$ or $P, Q \in \text{Te}$, then also $(p, P * Q, r) \in E$;
3. for all $(p, P * Q, r) \in E$ there is $q \in Q$ such that $(p, P, q), (q, Q, r) \in E$.

Proof. The first item is clear because X has all face maps, so for all $q \in Q = X$ with $\text{ev}(q) = U$ and all $S, T \subseteq U$, also $\delta_S^0(q), \delta_T^1(q) \in Q$.

The second property is induced by compositionality of lower resp. upper face maps. To show it, assume first that $P, Q \in \text{St}$, then $P = {}_B\uparrow U'$ for some $U' \in \square$ and $Q = {}_C\uparrow U$ where $U = U' \cup B$. These transitions are derived from the lower face maps $\delta_B^0 : X[U] \rightarrow X[U \setminus B]$ and $\delta_C^0 : X[U \setminus B] \rightarrow X[U \setminus (B \cup C)]$. Since X is a precubical set, we also have the face map $\delta_{B \cup C}^0 : X[U] \rightarrow X[U \setminus (B \cup C)]$ which gives the transition $(p, P * Q, r)$. We argue similarly when $P, Q \in \text{Te}$.

The argument for the third item is inverse to the above: $(p, P * Q, r) \in E$ mimicks $\delta_{B \cup C}^0$ (assuming $P, Q \in \text{St}$), which may be split into $\delta_C^0 \delta_B^0$. \square

We leave open the problem to give a precise characterization of ST-automata which are translations of HDAs.

5 Conclusion

Several previous works have studied interval pomsets with interfaces, their representations and their associated operational model. This paper unifies two different presentations (as a combinatorial object and as a word on a non-free monoid) and states how standard operations and transformations are expressed in both of these presentations.

We have shown that to every interval ipomset (up to isomorphism) corresponds an equivalence class of words, called step sequences, and that the transformation from one to another induces an isomorphism of categories. This implies that interval ipomsets are freely generated by certain discrete ipomsets (starters and terminators) under the relation which composes subsequent starters and subsequent terminators. We have also (constructively) exhibited a partial order on step sequences to represent subsumptions. Finally, we have explored the operational model on step sequences, that is ST-automata, and exposed translations between higher-dimensional automata (HDAs) and ST-automata. However, from ST-automata to HDAs this translation does not preserve languages, we only have inclusion. Stating the properties needed for an ST-automaton to have a precise HDA translation stays an open problem for now.

One thing which is missing from this paper is a treatment of interval representations of (interval) ipomsets. We believe that using the work of Myers in [26] it may be shown that any interval ipomset has a canonical interval representation which is closely related to its sparse step decomposition, see Fig. 2.

References

1. Amazigh Amrane, Hugo Bazille, Emily Clement, and Uli Fahrenberg. Languages of higher-dimensional timed automata. In *PETRI NETS*, 2024. Accepted.
2. Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. In Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa, editors, *ICTAC*, volume 14446 of *Lecture Notes in Computer Science*, pages 295–312. Springer-Verlag, 2023.

3. Stephen L. Bloom and Zoltán Ésik. Free shuffle algebras in language varieties. *Theoretical Computer Science*, 163(1&2):55–98, 1996.
4. Stephen L. Bloom and Zoltán Ésik. Varieties generated by languages with poset operations. *Mathematical Structures in Computer Science*, 7(6):701–713, 1997.
5. Armando Castañeda, Sergio Rajsbaum, and Michel Raynal. Unifying concurrent objects and distributed tasks: Interval-linearizability. *Journal of the ACM*, 65(6):45:1–45:42, 2018.
6. Uli Fahrenberg. Higher-dimensional timed automata. In Alessandro Abate, Antoine Girard, and Maurice Heemels, editors, *ADHS*, volume 51 of *IFAC-PapersOnLine*, pages 109–114. Elsevier, 2018.
7. Uli Fahrenberg. Higher-dimensional timed and hybrid automata. *Leibniz Transactions on Embedded Systems*, 8(2):03:1–03:16, 2022.
8. Uli Fahrenberg, Christian Johansen, Georg Struth, and Ratan Bahadur Thapa. Generating posets beyond N. In Uli Fahrenberg, Peter Jipsen, and Michael Winter, editors, *RAMiCS*, volume 12062 of *Lecture Notes in Computer Science*, pages 82–99. Springer-Verlag, 2020.
9. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021.
10. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. A Kleene theorem for higher-dimensional automata. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *CONCUR*, volume 243 of *LIPICs*, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
11. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285(2):104914, 2022.
12. 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-Verlag, 2023.
13. Jean Fanchon and Rémi Morin. Regular sets of pomsets with autoconcurrency. In Luboš Brim, Petr Jančar, Mojmir Křetínský, and Antonín Kučera, editors, *CONCUR*, volume 2421 of *Lecture Notes in Computer Science*, pages 402–417. Springer-Verlag, 2002.
14. Peter C. Fishburn. Intransitive indifference with unequal indifference intervals. *Journal of Mathematical Psychology*, 7(1):144–149, 1970.
15. Peter C. Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. Wiley, 1985.
16. Jay L. Gischer. The equational theory of pomsets. *Theoretical Computer Science*, 61:199–224, 1988.
17. J. Grabowski. On partial languages. *Fundamenta Informaticae*, 4(2):427, 1981.
18. Tony Hoare, Bernhard Möller, Georg Struth, and Ian Wehrman. Concurrent Kleene algebra and its foundations. *Journal of Logic and Algebraic Methods in Programming*, 80(6):266–296, 2011.
19. Tony Hoare, Stephan van Staden, Bernhard Möller, Georg Struth, and Huibiao Zhu. Developments in concurrent Kleene algebra. *Journal of Logic and Algebraic Methods in Programming*, 85(4):617–636, 2016.
20. Ryszard Janicki, Jetty Kleijn, Maciej Koutny, and Lukasz Mikulski. *Paradigms of Concurrency - Observations, Behaviours, and Systems - a Petri Net View*, volume 1020 of *Studies in Computational Intelligence*. Springer-Verlag, 2022.

21. Ryszard Janicki and Maciej Koutny. Operational semantics, interval orders and sequences of antichains. *Fundamenta Informaticae*, 169(1-2):31–55, 2019.
22. Ryszard Janicki and Xiang Yin. Modeling concurrency with interval traces. *Information and Computation*, 253:78–108, 2017.
23. Tobias Kappé, Paul Brunet, Alexandra Silva, and Fabio Zanasi. Concurrent Kleene algebra: Free model and completeness. In Amal Ahmed, editor, *ESOP 2018*, volume 10801 of *Lecture Notes in Computer Science*, pages 856–882. Springer-Verlag, 2018.
24. Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.
25. Leslie Lamport. On interprocess communication. Part I: basic formalism. *Distributed Computing*, 1(2):77–85, 1986.
26. Amy Myers. Basic interval orders. *Order*, 16(3):261–275, 1999.
27. Vaughan R. Pratt. Modeling concurrency with partial orders. *J. Parallel Programming*, 15(1):33–71, Feb 1986.
28. Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. See also [?].
29. Norbert Wiener. A contribution to the theory of relative position. *Proceedings of the Cambridge Philosophical Society*, 17:441–449, 1914.