



# Higher-Dimensional Automata: Extension to Infinite Tracks

Luc Passemard   

IRIF & Université Paris Cité, Paris, France

Amazigh Amrane 

EPITA Research Laboratory (LRE), Le Kremlin-Bicêtre, France

Uli Fahrenberg 

EPITA Research Laboratory (LRE), Le Kremlin-Bicêtre, France

---

## Abstract

We introduce higher-dimensional automata for infinite interval ipomsets ( $\omega$ -HDAs). We define key concepts from different points of view, inspired from their finite counterparts. Then we explore languages recognized by  $\omega$ -HDAs under Büchi and Muller semantics. We show that Muller acceptance is more expressive than Büchi acceptance and, in contrast to the finite case, both semantics do not yield languages closed under subsumption. Then, we adapt the original rational operations to deal with  $\omega$ -HDAs and show that while languages of  $\omega$ -HDAs are  $\omega$ -rational, not all  $\omega$ -rational languages can be expressed by  $\omega$ -HDAs.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Automata over infinite objects

**Keywords and phrases** Higher-dimensional automata, concurrency theory, omega pomsets, Büchi acceptance, Muller acceptance, interval pomsets, pomsets with interfaces

**Digital Object Identifier** 10.4230/LIPIcs.FSCD.2025.31

**Related Version** *Full Version:* <https://arxiv.org/abs/2503.07881> [38]

**Funding** *Luc Passemard:* This work was done while the first author was a Master’s intern at EPITA.

**Acknowledgements** The authors are grateful to anonymous referees for their comments and corrections.

## 1 Introduction

Automata theory is fundamental for modeling and analyzing computational systems. It is used to verify system correctness, infer models for unknown systems, synthesize components from specifications, and develop decision procedures. Finite automata over words (Kleene automata) model terminating sequential systems with finite memory, where accepted words represent execution sequences. Their theory, backed by the Kleene, Büchi, and Myhill-Nerode theorems, connects regular expressions, monadic second-order logic, and semigroups. For concurrent systems, executions may be represented as *pomsets* (partially ordered multisets) [40] instead of words. In a pomset, concurrent events are represented as labeled elements that are not ordered relative to each other. Different classes of pomsets and their associated automata models exist, reflecting diverse interpretations of concurrency. We can cite for example branching automata and series-parallel pomsets [32–35], step transition systems and local trace languages [23], communicating finite-state machines and message sequence charts [27], asynchronous automata and Mazurkiewicz traces [46] or higher-dimensional automata (HDAs) and interval pomsets [18].

In this paper, we focus on HDAs [41, 42]. They are general models of concurrency that extend, for example, event structures and safe Petri nets [6, 43], asynchronous transition systems [9, 45] and obviously Kleene automata. Initially studied from a geometrical or



© Luc Passemard, Amazigh Amrane, and Uli Fahrenberg;  
licensed under Creative Commons License CC-BY 4.0

10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025).

Editor: Maribel Fernández; Article No. 31; pp. 31:1–31:23

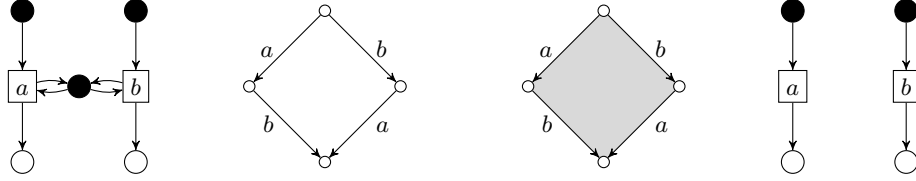
Leibniz International Proceedings in Informatics



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

categorical point of view, the language theory of HDAs has become another focus for research in the past few years since [18]. Fahrenberg et al have now shown a Kleene theorem [19], a Myhill-Nerode theorem [22] and a Büchi theorem [5]. Higher-dimensional timed automata are introduced in [16] and their languages in [2]. HDAs consist of a collection of cells in which events are running concurrently, connected by face maps which model the start and termination of events. The language of an HDA is defined as a set of *interval pomsets* [25] with interfaces (interval ipomsets or *ipomsets*) [20]. Interval pomsets are suitable for situations where events in concurrent systems extend over time, such as producer-consumer systems, which series-parallel pomsets cannot capture. Modelling executions with interval pomsets supports partial-order reduction, with a representation that is exponentially smaller than other alternatives.

The idea in an HDA is that each event in an execution  $P$  is a time interval of process activity. The execution is built by joining elementary steps, each representing segments of  $P$ . This gluing composition allows events to extend across segments, connecting one part to the next. In addition, any order extension of  $P$  is also a valid behaviour for the HDA. We say that the language is closed under *subsumption*. As an example, Fig. 1 shows Petri net and HDA models for a system with two events, labeled  $a$  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  $b$  where the process  $a \parallel b$  is a continuous path (called *track*) through the surface of the filled-in square, starting at the top and terminating at the bottom node. The shape of such a track defines the interval scheduling of  $a$  and  $b$  where the intervals overlap.



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

Muller [37] and Büchi [10] introduced automata recognizing  $\omega$ -words to study non-terminating sequential machines and decision problems. McNaughton [36] later proved the equivalence of these definitions and extended Kleene's theorem to  $\omega$ -words through a non-nested  $\omega$ -iteration. These automata also have logical [10] and algebraic [11] characterizations. Beyond their theoretical significance, they and their variations (such as Rabin, Streett, and parity automata) play a key role in specifying and verifying reactive systems [14]. This led to the extension of automata models for concurrency to the infinite case, with fundamental results carrying over. For instance,  $\omega$ -branching automata admit Kleene-like and Büchi-like theorems [31], while similar developments apply to traces [12, 15, 26], leading to decision procedures as corollaries.

In this paper we introduce higher-dimensional automata for  $\omega$ -interval pomsets:  $\omega$ -HDAs. To do so, we first define a class of  $\omega$ -interval pomsets with interfaces suitable for  $\omega$ -HDAs and extend the fundamental concepts to the infinite case. This may be effective for modelling (and checking) reactive concurrent systems that may not terminate, especially when reasoning about liveness properties. As in the finite case, we show that isomorphisms of  $\omega$ -ipomsets are unique and that interval  $\omega$ -ipomsets admit several equivalent definitions and canonical decompositions. Then we define languages of  $\omega$ -HDAs in terms of interval  $\omega$ -ipomsets by

considering Büchi and Muller acceptances. We show that unlike HDAs, languages of  $\omega$ -HDAs may not be closed under subsumption. Thus, we revise the rational operations defined in [19] by ensuring that subsumption is not implicitly assumed and introduce an omega iteration to define  $\omega$ -rational languages of ipomsets. We show by translating  $\omega$ -HDAs to  $\omega$ -automata over discrete ipomsets, called ST-automata, that languages of  $\omega$ -HDAs are  $\omega$ -rational. On the other hand, we show that, unlike  $\omega$ -automata, Büchi acceptance is less expressive than the Muller one in  $\omega$ -HDAs. In addition, there are  $\omega$ -rational languages that cannot be recognized by Muller  $\omega$ -automata. To address this, we initiate first steps towards characterizing the subclass of omega-rational languages that are as expressive as Büchi or Muller  $\omega$ -HDAs. We refer to the long version [38] for detailed proofs of our results.

## 2 Ipomsets

Ipomsets generalize pomsets [28, 29, 40]: a well-established model for non-interleaving concurrency. In this section, we review the fundamental definitions and extend them to the infinite case.

### 2.1 Finite ipomsets

We fix a finite alphabet  $\Sigma$ . An *ipomset* (over  $\Sigma$ ) is a structure  $(P, <, \dashrightarrow, S, T, \lambda)$  consisting of a finite set of *events*  $P$ , two strict partial orders:  $<$  the *precedence order*, and  $\dashrightarrow$  the *event order*, a set  $S \subseteq P$  of *source* interfaces, a set  $T \subseteq P$  of *target* interfaces such that the elements of  $S$  are  $<$ -minimal and those of  $T$  are  $<$ -maximal, and a labeling function  $\lambda : P \rightarrow \Sigma$ . In addition, we require that the relation  $< \cup \dashrightarrow$  is total, but not necessarily an order. In fact, if we have  $x < y$  and  $y \dashrightarrow x$ , then  $< \cup \dashrightarrow$  is not an order.

For the purpose of notation, we use  ${}_S P_T$  for an ipomset  $(P, <, \dashrightarrow, S, T, \lambda)$ , or refer to it by its set of events  $P$  and to its components by adding the subscript  ${}_P$ .

We highlight two special cases of ipomsets: *conclists*  $(U, \dashrightarrow, \lambda)$  where  $S_U = T_U = \emptyset$  and  $<_U = \emptyset$  hence  $\dashrightarrow$  total, with  $\square$  the set of conclists, and *identities*  $\text{id}_U = {}_U U_U$  where  $S_U = T_U = U$  and  $<_U = \emptyset$ , with  $\text{Id}$  the set of identities. We call  $\text{id}_\emptyset$  the empty conclist/identity. Note that for any ipomset  $P$ ,  $S_P = (S, \dashrightarrow|_{S \times S}, \lambda|_S)$  and  $T_P = (T, \dashrightarrow|_{T \times T}, \lambda|_T)$ , where “ $|$ ” denotes restriction, are conclists. An ipomset  $P$  is said to be *interval* (called *iipomset*) if  $<_P$  is an interval order, i.e. if for all  $w, x, y, z \in P$ , if  $w < y$  and  $x < z$ , then  $w < z$  or  $x < y$ .

Let  $P$  and  $Q$  be two ipomsets. We say that  $Q$  *subsumes*  $P$  (written  $P \sqsubseteq Q$ ) if there is a bijection  $f : P \rightarrow Q$  (a *subsumption*) such that  $\lambda_Q \circ f = \lambda_P$ ,  $f(S_P) = S_Q$ , and  $f(T_P) = T_Q$ ,  $\forall x, y \in P$ ,  $f(x) <_Q f(y) \Rightarrow x <_P y$  and  $\forall x, y \in P$  such that  $x \not<_P y$  and  $y \not<_P x$ , we have  $x \dashrightarrow_P y \Rightarrow f(x) \dashrightarrow_Q f(y)$ . Informally speaking,  $P$  is more precedence ordered than  $Q$ .

► **Example 1.** An example of subsumption is depicted in Fig. 2. Note that  $e_1 < e_2$  when the activity interval of  $e_1$  finishes before the beginning of the one of  $e_2$ . When the activity intervals of two events overlap, they are  $\dashrightarrow$ -ordered from top to bottom.

An *isomorphism* of ipomsets is an invertible subsumption (whose inverse is again a subsumption); isomorphic ipomsets are denoted  $P \cong Q$ . Due to the totality of  $< \cup \dashrightarrow$ , isomorphisms between ipomsets are unique [4], so we may switch freely between ipomsets and their isomorphism classes. The set of ipomsets is denoted  $\text{iPoms}$ , and the set of interval ipomsets is written  $\text{iiPoms}$ .

In an ipomset  $P$ , a *chain* is a subset of  $P$  totally ordered by  $<_P$ . An *antichain*  $A$  of  $P$  is such that  $<_P \cap (A \times A) = \emptyset$ . Hence  $A$  is totally ordered by  $\dashrightarrow_P$ . The width of  $P$  is  $\text{wd}(P) = \sup\{|A| \mid A \text{ is an } <\text{-antichain of } P\}$ .



■ **Figure 2** Ipomsets (left) and their corresponding interval representations (right). Full arrows indicate precedence order; dashed arrows indicate event order; bullets indicate interfaces.

## 2.2 The set of $\omega$ -ipomsets

Similarly to ipomsets, an  $\omega$ -ipomset is a structure of the form  $(P, <, \dashrightarrow, S, \lambda)$ . The only changes are that  $P$  is countably infinite and there is no target interface. However, let  $\Omega_P$  denote the set of *all*  $<_P$ -maximal elements. This is not similar to a target interface: even if  $\Omega_P$  represents unfinished events, terminating them (by trying to glue an ipomset after  $P$ ) may cause chains greater than  $\omega$ .

For the remainder of the paper, we focus on a subclass of  $\omega$ -ipomsets.

► **Definition 2** (Valid  $\omega$ -ipomset). *A valid  $\omega$ -ipomset is an  $\omega$ -ipomset  $P$  such that*

- $\forall x \in P, \{y \mid y <_P x\}$  is finite;
- every  $<_P$ -antichain of  $P$  is finite.

The first point is here to forbid  $\omega$ -ipomsets with chains greater than  $\omega$ , and therefore extend properly classical  $\omega$ -words and  $\omega$ -pomsets [31]. It is a classical property required in event structures (see [44] for example). The second point is to avoid an infinite number of concurrent events. This condition implies in particular that  $S_P$  and  $\Omega_P$  are finite. Both conditions also imply that  $<_P$  is a well quasi-order, but the converse is not true. In the rest of this paper, unless specified, we will talk about valid  $\omega$ -ipomsets, and will omit the word “valid”.

Let  $P$  and  $Q$  be two (valid)  $\omega$ -ipomsets. We define the subsumption  $P \sqsubseteq Q$  as in the finite case by preserving also the  $<_P$ -maximal elements  $\Omega_P$ . Again, isomorphisms of  $\omega$ -ipomsets are invertible subsumptions (whose inverse is again a subsumption). The first result of this paper is the unicity of these isomorphisms. The proof is similar to the finite case, using pomset filtration (see [18, Lem. 34]):

► **Proposition 3.** *Isomorphisms between  $\omega$ -ipomsets are unique.*

Note that the proposition does not hold for general (non-valid)  $\omega$ -ipomsets: for example, all shifts by  $n \in \mathbb{Z}$  are non-trivial automorphisms of  $\mathbb{Z}$ . As in the finite case, we may now switch freely between  $\omega$ -ipomsets and their isomorphism classes. The set of  $\omega$ -ipomsets is denoted  $\text{iPoms}^\omega$ , and we use the notation  $\text{iPoms}^\infty := \text{iPoms} \cup \text{iPoms}^\omega$ .

## 2.3 Operations on $\omega$ -ipomsets

We extend here the operations  $*$  (gluing) and  $\parallel$  (parallel composition) to  $\omega$ -ipomsets.

For  $(P, Q) \in (\text{iPoms}^\infty \times \text{iPoms}^\infty) \setminus (\text{iPoms} \times \text{iPoms})$ , the *parallel composition*  $P \parallel Q$  (or  $\left[ \begin{smallmatrix} P \\ Q \end{smallmatrix} \right]$ ) is the  $\omega$ -ipomset  $(P \sqcup Q, <_P \sqcup <_Q, \dashrightarrow, S_P \sqcup S_Q, \lambda_P \sqcup \lambda_Q)$  where  $\sqcup$  is the disjoint union and  $x \dashrightarrow y$  iff  $(x, y) \in P \times Q$  or  $x \dashrightarrow_P y$  or  $x \dashrightarrow_Q y$ .

The *gluing composition*  $P * Q$  (or  $PQ$ ) of  $(P, Q) \in \text{iPoms} \times \text{iPoms}^\omega$  is defined if there exists a unique<sup>1</sup> isomorphism  $f: T_P \rightarrow S_Q$ , by  $((P \sqcup Q)_{/x \sim f(x)}, <, \dashrightarrow, S_P, \lambda_P \cup \lambda_Q)^2$ , where:

- $x < y$  iff  $x <_P y$ ,  $x <_Q y$ , or  $x \in P \setminus T_P$  and  $y \in Q \setminus S_Q$ ;
- $\dashrightarrow$  is the transitive closure of  $\dashrightarrow_P \cup \dashrightarrow_Q$  on  $(P \sqcup Q)_{/x \sim f(x)}$ .

<sup>1</sup> Isomorphisms of conclists is a special case of ipomset isomorphisms. Unicity is ensured by event order.

<sup>2</sup>  $(P \sqcup Q)_{/x \sim f(x)}$  is the quotient of the disjoint union under the unique isomorphism  $f: T_P \rightarrow S_Q$ .

Informally, the sources of  $Q$  are attached to the targets of  $P$ .

Note that both  $*$  and  $\parallel$  are associative and non-commutative. The non-commutativity of  $\parallel$  is due to event order. The latter is crucial for isomorphism unicity and to define gluing composition. Still, even if  $\begin{bmatrix} a \\ b \end{bmatrix} \neq \begin{bmatrix} b \\ a \end{bmatrix}$ , both express that  $a$  and  $b$  are running concurrently. Alur et al in [1] introduced a similar “ordered parallel composition” for application purposes.

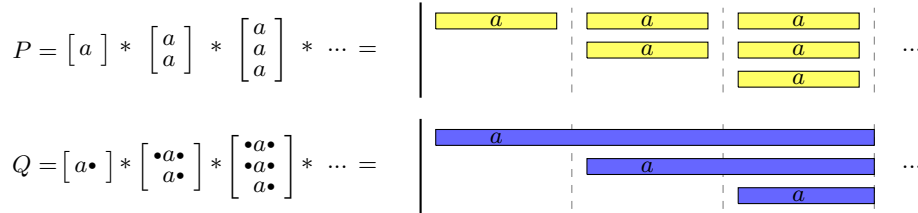
We also introduce an *infinite gluing* of an  $\omega$ -sequence of ipomsets. Let  $(P_i)_{i \in \mathbb{N}} \in \text{iPoms}^{\mathbb{N}}$  such that for all  $i$ ,  $T_{P_i} \cong S_{P_{i+1}}$ . We define  $P = P_0 * P_1 * \dots$  by  $(P, <, \dashrightarrow, S_P, \lambda_P)$ , where

- $P = (\bigcup_{i \in \mathbb{N}} (P_i, i)) / x \sim_f(x)$  where  $f : \bigcup_{i \in \mathbb{N}} (T_i, i) \rightarrow \bigcup_{i \in \mathbb{N}} (S_i, i)$  such that  $f(x, i) = (f_i(x), i + 1)$  with  $f_i$  the unique isomorphism between  $T_{P_i}$  and  $S_{P_{i+1}}$ ;
- $S_P = S_{P_0} \times \{0\}$ ;
- $(x, i) <_P (y, j)$  if  $(i = j \text{ and } x <_{P_i} y)$  or  $(i + 1 \leq j, x \in P_i \setminus T_{P_i} \text{ and } y \in P_{i+1} \setminus S_{P_{i+1}})$ ;
- $\dashrightarrow_P$  is the transitive closure of  $\bigcup_{i \in \mathbb{N}} \dashrightarrow_{P_i}$ ;
- $\lambda_P(x, i) = \lambda_{P_i}(x)$  for  $i \in \mathbb{N}$  and  $x \in P_i$ .

Note in particular that when  $P_i \in \text{Id}$  we have  $P_0 * \dots * P_{i-1} * P_i * P_{i+1} * \dots = P_0 * \dots * P_{i-1} * P_{i+1} * \dots$ . We let denote by  $P^\omega$  the  $\omega$ -ipomset defined by the infinite gluing of the constant sequence equal to  $P$  with  $S_P \cong T_P$ .

► **Lemma 4.** For  $(P_i)_{i \in \mathbb{N}} \in \text{iPoms}^{\mathbb{N}}$  with  $T_{P_i} \cong S_{P_{i+1}}$ ,  $P = P_0 * P_1 * \dots$  is valid if the number of  $P_i \in \text{iPoms} \setminus \text{Id}$  is infinite and there exists  $m \in \mathbb{N}$  such that for all  $i \in \mathbb{N}$ ,  $\text{wd}(P_i) \leq m$ .

Intuitively,  $P$  is infinite since it is obtained by composing infinitely many non-identities, finite past is a consequence of the infinite gluing definition, and finiteness of antichains is due to width-boundedness of the operands. Note that this is not an equivalence condition. The infinite gluing may define a valid  $\omega$ -ipomset even if  $\text{wd}(P_i)$  are not bounded (see Fig. 3). Finding an equivalence condition on  $(P_i)$  is hard, because it has to consider event order. For example, Fig. 6 exhibits two infinite products behaving differently with only a change of event order.



■ **Figure 3** Two unbounded infinite products giving a valid (up) and an invalid (down)  $\omega$ -ipomset.

## 2.4 Interval $\omega$ -ipomsets

We will here only deal with interval  $\omega$ -ipomsets. An  $\omega$ -ipomset is *interval* (denoted  $\omega$ -ipomset) if for all  $w, x, y, z \in P$ , if  $w < y$  and  $x < z$ , then  $w < z$  or  $x < y$ . The set of all interval  $\omega$ -ipomsets is denoted  $\text{iiPoms}^\omega$  (and we use the notation  $\text{iiPoms}^\infty := \text{iiPoms} \cup \text{iiPoms}^\omega$ ).

There are multiple equivalent ways to define interval ipomsets [18, Lem. 39], and the same goes for  $\omega$ -ipomsets.

We say that  $P \in \text{iPoms}^\omega$  admits an *interval representation* if there are two functions  $b, e : P \rightarrow \mathbb{N} \cup \{+\infty\}$  such that for all  $x, y \in P$ ,  $b(x) \leq e(x)$  and  $x <_P y \Leftrightarrow e(x) < b(y)$ .

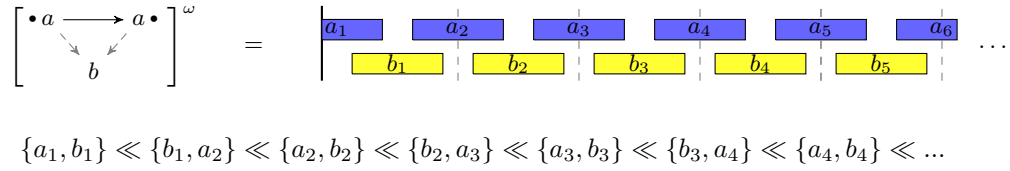
As for finite pomsets and with similar arguments, the following holds:

► **Proposition 5.** *Let  $P$  be an  $\omega$ -ipomset, the following are equivalent:*

1.  $P$  is an interval  $\omega$ -ipomset;
2.  $P$  has an interval representation;
3. the order  $\ll$  on maximal antichains of  $P$  defined by  $A \ll B$  iff  $A \neq B$  and  $\forall (a, b) \in A \times B$ ,  $b \not\prec_P a$  is linear.

Establishing the equivalence between the first two items is a routine. The third item is less intuitive, but it implies the second one by showing that the set of maximal antichains equipped with  $\ll$  is isomorphic to  $(\mathbb{N}, <)$ . From that, to an event  $a \in P$  we can associate the interval going from the index of the first appearance of  $a$  in the antichains to the index of the last appearance.

► **Example 6.** The minimal non-interval ipomset  $aa \parallel bb$  can be generalized to the non-interval  $\omega$ -ipomset  $a^\omega \parallel b^\omega$ , while  $(a \parallel b)^\omega \in \text{iiPoms}^\omega$ , as well as  $[\bullet a \bullet]^\omega$  represented in Fig. 4.

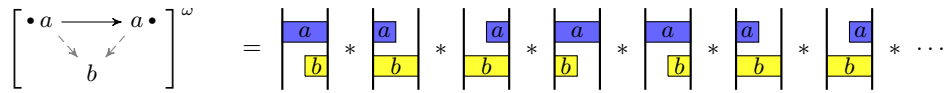


■ **Figure 4** Interval representation and linear order on antichains of the  $\omega$ -ipomset of Ex. 6.

## 2.5 Decomposition of $\omega$ -ipomsets

A *starter*  $U$  is a discrete ipomset (i.e.  $<_U$  is empty) with  $T_U = U$  (hence written  ${}_S U_U$ ), and a *terminator*  $U$  is a discrete ipomset with  $S_U = U$  (written  ${}_U U_T$ ). We denote by  $\Xi$  the set of starters and terminators. Note that  $\text{Id} \subseteq \Xi$ . We say that  $U \in \Xi$  is *proper* if  $U \in \Xi \setminus \text{Id}$ . A starter  ${}_U {}_{-A} U_U$  will be written  ${}_A \uparrow U$  (meaning it contains the events in  $U$  and start the events in  $A \subseteq U$ ), and a terminator  ${}_U U_{-A}$  will be written  $U \downarrow_A$ . We call a *step decomposition* of  $P \in \text{iPoms}^\omega$  a sequence of starters and terminators  $(U_i)_{i \in \mathbb{N}}$  such that  $P = U_0 * U_1 * \dots$ , and a decomposition is said to be *sparse* if proper starters and terminators are alternating. As for finite ipomsets [22], we can prove uniqueness of sparse decompositions of  $\omega$ -ipomsets:

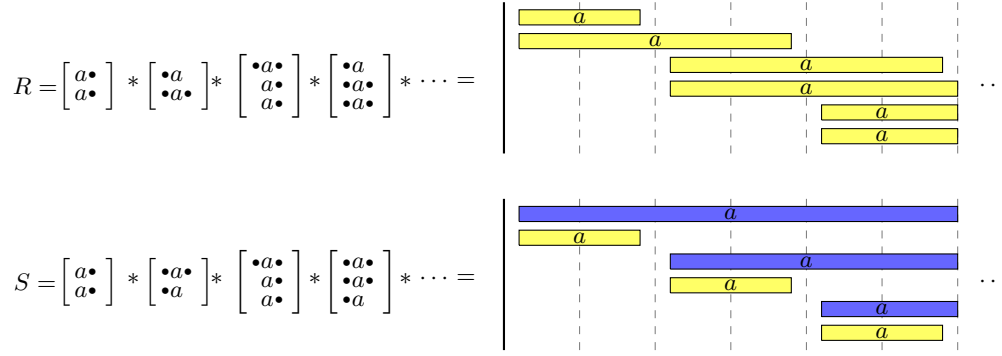
► **Theorem 7.** *Every interval  $\omega$ -ipomset has a unique sparse step decomposition.*



■ **Figure 5** Sparse decomposition of Ex. 6.

► **Remark 8.** In Th. 7,  $\omega$ -ipomsets are in  $\text{iiPoms}^\omega$  so supposed valid, thus we do not have to verify that the infinite product effectively gives a valid result. Indeed, an infinite product of alternating proper starters and terminators may yield a non-valid  $\omega$ -ipomset (see Fig. 6).

Defining such a decomposition gives us a characterization of  $\omega$ -ipomsets in terms of their prefixes. We say that  $A \in \text{iPoms}$  is a (finite) *prefix* of  $P \in \text{iPoms}^\omega$  if there is  $Q \in \text{iPoms}^\omega$  such that  $P = AQ$ , and then write  $A \prec P$ . We call  $\text{Pref}(P)$  the set of all prefixes of  $P$ .



■ **Figure 6** Two sparse decompositions with different event orders, such that only  $R$  is valid.

► **Remark 9.** Unlike classical  $\omega$ -words theory, two prefixes of  $P$  may be incomparable, for example  $a\bullet \prec [\frac{a}{b}]$  and  $b\bullet \prec [\frac{a}{b}]$  but  $a\bullet \not\prec b\bullet$  and  $b\bullet \not\prec a\bullet$ . But we still have the property that for  $A \in \text{Pref}(P)$  and  $x \in A$ , if  $y <_P x$  then  $y \in A$ . Note that neither  $a\bullet$  nor  $b\bullet$  could be the first step of the sparse decomposition of  $(a \parallel b)^\omega$ . Its only possible first step is  $[\frac{a\bullet}{b\bullet}]$ .

The following lemma is trivial in classical  $\omega$ -word theory, but here relies crucially on the fact that we deal with valid  $\omega$ -ipomsets.

► **Lemma 10.** *If  $(A_i)_{i \in I}$  is an infinite set of prefixes of an  $\omega$ -ipomset  $P$ , and  $R$  is a finite subset of  $P$  (seen as a set), then there is  $i \in I$  such that  $R \subseteq A_i$ .*

**Proof.** Let  $(A_i)_{i \in I} \in \text{Pref}(P)^I$  (with  $I$  infinite), and  $x \in R$ . We want to prove that the set  $\{A_i \mid x \in A_i\}$  is infinite. Let  $x \in P$ , and  $P^m$  be (the antichain of) the  $<$ -minimal elements of  $\{x\} \cup \{p \in P \mid x \not\prec p \text{ and } p \not\prec x\}$ , and  $Q := P^m \cup \{p \in P \mid p < x\}$ . As  $P$  is valid,  $Q$  is finite, and so there is a finite number of ipomsets included in  $Q$ . Thus there is an infinite number of  $A_i$  s.t.  $A_i \cap (P - Q) \neq \emptyset$ . In such  $A_i$ , let  $y_i \in A_i \cap (P - Q)$ . There is an element  $z_i \in P^m$  s.t.  $z_i < y_i$ : either  $x < y_i$  (and so  $z_i = x$ ), or  $y_i \in \{p \in P \mid x \not\prec p \text{ and } p \not\prec x\}$  (and so there is  $z_i \in P^m \subseteq Q$  such that  $z_i \leq y_i$ , and  $z_i \neq y_i$  as  $z_i \in Q$  and  $y_i \notin Q$ ). As  $A_i$  is a prefix of  $P$ ,  $y_i \in A_i$  implies that  $z_i$  is necessarily in  $A_i - T_{A_i}$ , and so  $x \in A_i$  (otherwise  $x \in B_i - S_{B_i}$  with  $P = A_i * B_i$  and so  $z_i < x$ , which is impossible because  $P^m$  is an antichain). So there are infinitely many  $A_i$  such that  $x \in A_i$ .

Then, we take  $(A_i)_{i \in I'}$ , an infinite set of prefixes of  $P$  that all contain  $x$ , and we iteratively use the previous argument to get all elements of the (finite) set  $R$ . At the end we have  $(A_i)_{i \in J}$ , an infinite set of prefixes of  $P$  that all contain  $R$ , and we conclude the proof by taking any one of them. ◀

► **Remark 11.** Note that  $P$  is not necessarily interval in Lem. 10. It only suffices to have an infinite number of prefixes. However being an interval  $\omega$ -ipomset ensures the existence of an infinite set of prefixes, see below. For example,  $P = a^\omega \parallel b^\omega \notin \text{iiPoms}^\omega$  has a finite number of prefixes:  $\text{Pref}(P) = \{a\bullet, b\bullet, [\frac{a\bullet}{b\bullet}]\}$ , while  $Q = (aa \parallel bb).c^\omega \notin \text{iiPoms}^\omega$  has an infinite number of prefixes:  $\text{Pref}(Q) \supseteq \{(aa \parallel bb).c^n \mid n \in \mathbb{N}\}$ .

From Lem. 10 follows a sequential characterization of interval  $\omega$ -ipomsets:

► **Proposition 12.** *Let  $P$  be an  $\omega$ -ipomset, the following are equivalent:*

1.  $P$  is an interval  $\omega$ -ipomset;
2.  $\text{Pref}(P)$  is infinite and for all  $A \in \text{Pref}(P)$ ,  $A$  is an interval ipomset;
3. an infinite number of prefixes of  $P$  are interval ipomsets.



In addition, from the last point of Prop. 12 along with the preservation of the interval property by gluing composition (see [18, Lem. 41]) we have the following:

► **Corollary 13.** *If  $P = P_0 * P_1 * \dots$  is a valid  $\omega$ -ipomset and  $\forall i \in \mathbb{N}, P_i \in \text{iiPoms}$ , then  $P \in \text{iiPoms}^\omega$ .*

### 3 Higher-dimensional automata

In this section, we recall *higher-dimensional automata* (HDAs) over ipomsets [7, 18, 22] and introduce  $\omega$ -HDAs. Recall that  $\square$  denotes the set of conclists.

#### 3.1 HDAs over finite ipomsets

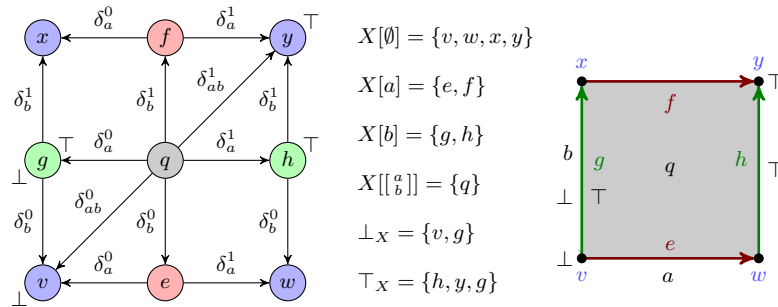
A *precubical set* is a structure  $(X, \text{ev}, \Delta)$  with:

- a set of cells  $X$ ;
- a function  $\text{ev} : X \rightarrow \square$  which assigns to a cell its list of active events (for  $U \in \square$  we write  $X[U] = \{q \in X \mid \text{ev}(q) = U\}$ );
- face maps  $\Delta = \{\delta_{A,U}^0, \delta_{A,U}^1 \mid U \in \square, A \subseteq U\}$  such that  $\delta_{A,U}^0, \delta_{A,U}^1 : X[U] \rightarrow X[U - A]$ ;
- for  $A, B \subseteq U$  s.t.  $A \cap B = \emptyset$  and  $\mu, \nu \in \{0, 1\}$ , we have the following *precubical identities*:  
 $\delta_{A,U-B}^\mu \circ \delta_{B,U}^\nu = \delta_{B,U-A}^\nu \circ \delta_{A,U}^\mu$ .

We usually refer to a precubical set by its set of cells  $X$ , and for face maps we often omit the second subscript  $U$ . The *dimension* of a cell  $q \in X$  is the size  $|\text{ev}(q)|$ . An *upper* face map  $\delta_A^1$  terminates the events in  $A$ , whereas a *lower* face map  $\delta_A^0$  “unstarts” these events, that is, it maps to a cell where the events of  $A$  are not yet started.

A *higher-dimensional automaton* is a tuple  $(X, \perp_X, \top_X)$  where  $X$  is a *finite* precubical set, and  $\perp_X, \top_X \subseteq X$  are the sets of *starting* and *accepting* cells. We may omit the subscripts  $X$  if the context is clear, and refer to an HDA only by its precubical set.

► **Example 14.** Fig. 7 shows a two-dimensional HDA as a combinatorial object (left) and in a geometric realisation (right). The arrows between the cells on the left representation correspond to the face maps connecting them. It consists of nine cells: the corner cells  $X_0 = \{x, y, v, w\}$  in which no event is active (for all  $z \in X_0$ ,  $\text{ev}(z) = \emptyset$ ), the transition cells  $X_1 = \{g, h, f, e\}$  in which one event is active ( $\text{ev}(f) = \text{ev}(e) = a$  and  $\text{ev}(g) = \text{ev}(h) = b$ ), and the square cell  $q$  where  $\text{ev}(q) = [a, b]$ . By convention, for  $a \dashrightarrow b$ , we represent the event  $a$  horizontally and the event  $b$  vertically.



■ **Figure 7** A two-dimensional HDA  $X$  on  $\Sigma = \{a, b\}$ , see Ex. 14.



A *track*  $\alpha$  in  $X$  is a sequence  $(q_0, \varphi_1, q_1, \dots, \varphi_n, q_n)$  where  $q_i \in X$  are cells and  $\varphi_i$  denote face map types. That is, for all  $i \leq n$ ,  $(q_i, \varphi_i, q_{i+1})$  is:

- either an *upstep*:  $(\delta_A^0(q_{i+1}), \nearrow^A, q_{i+1})$  with  $A \subseteq \text{ev}(q_{i+1})$ ,
- or a *downstep*:  $(q_i, \searrow_B, \delta_B^1(q_i))$  with  $B \subseteq \text{ev}(q_i)$ .

The *source* of  $\alpha$  is  $\text{src}(\alpha) = q_0$  and its *target* is  $\text{tgt}(\alpha) = q_n$ . A track is *accepting* if  $\text{src}(\alpha) \in \perp_X$  and  $\text{tgt}(\alpha) \in \top_X$ . Note that tracks are concatenations (denoted using  $*$ ) of upsteps and downsteps. The language of an HDA is defined in terms of the event ipomsets of its accepting tracks. Formally, the *event ipomset* of a track  $\alpha$  (written  $\text{ev}(\alpha)$ ) is defined by:

- $\text{ev}((q)) = \text{id}_{\text{ev}(q)}$  (with  $q \in X$  and  $(q)$  a single-cell track)
- $\text{ev}((q, \nearrow^A, p)) = A \uparrow \text{ev}(p)$  (the starter which starts events  $A$ )
- $\text{ev}((q, \searrow_B, p)) = \text{ev}(q) \downarrow_B$  (the terminator which terminates events  $B$ )
- $\text{ev}(\alpha_1 * \dots * \alpha_n) = \text{ev}(\alpha_1) * \dots * \text{ev}(\alpha_n)$

We say that two tracks  $\alpha$  and  $\beta$  are equivalent, denoted  $\alpha \simeq \beta$ , with  $\simeq$  the relation generated by the three cases  $(x \nearrow^A y \nearrow^B z) \simeq (x \nearrow^{A \cup B} z)$ ,  $(x \searrow_A y \searrow_B z) \simeq (x \searrow_{A \cup B} z)$ , and  $\gamma \alpha \delta \simeq \gamma \beta \delta$  whenever  $\alpha \simeq \beta$ . Basically, two tracks are equivalent if their “contractions” into sparse tracks (i.e. alternating upsteps and downsteps) and event ipomsets are equal.

► **Example 15.** The HDA  $X$  of Fig. 7 admits several accepting tracks with target  $h$ , for example  $v \nearrow^{ab} q \searrow_a h$ . This is a sparse track and equivalent to the non-sparse tracks  $v \nearrow^a e \nearrow^b q \searrow_a h$  and  $v \nearrow^b g \nearrow^a q \searrow_a h$ . Their event ipomset is  $[ \begin{smallmatrix} a \\ b \bullet \end{smallmatrix} ]$ . The track  $v \nearrow^a e \searrow_a w \nearrow^b h$  is also accepting with event ipomset  $ab \bullet \sqsubseteq [ \begin{smallmatrix} a \\ b \bullet \end{smallmatrix} ]$ . In addition, since  $g$  is both a start and accept cell, we have also  $g$  and  $v \nearrow^b g$  as accepting tracks, with event ipomsets  $\bullet b \bullet$  and  $b \bullet$ , respectively. Note that tracks move forward along upper face maps but backward along lower ones.  $X$  accept tracks whose ipomsets are in  $\{b \bullet, \bullet b \bullet, ab \bullet, [ \begin{smallmatrix} a \\ b \bullet \end{smallmatrix} ], [ \begin{smallmatrix} a \\ \bullet b \bullet \end{smallmatrix} ], \bullet ba, [ \begin{smallmatrix} a \\ \bullet \end{smallmatrix} ], ab, ba, [ \begin{smallmatrix} a \\ b \end{smallmatrix} ]\}$ .

The *language* of an HDA  $X$  is  $L(X) = \{\text{ev}(\alpha) \mid \text{src}(\alpha) \in \perp_X \text{ and } \text{tgt}(\alpha) \in \top_X\}$ . The following result about languages of HDAs is a fundamental property:

► **Proposition 16** ([19, Prop. 10]). *Languages of HDAs over finite ipomsets are closed under subsumption (or down-closed).*

For  $L \subseteq \text{iPoms}$ , we write  $L \downarrow = \{P \in \text{iiPoms} \mid \exists Q \in L, P \sqsubseteq Q\}$  for the subsumption closure of  $L$ . For example, the language of Ex. 14 is  $L(X) = \{b \bullet, \bullet b \bullet, [ \begin{smallmatrix} a \\ b \bullet \end{smallmatrix} ], [ \begin{smallmatrix} a \\ \bullet b \bullet \end{smallmatrix} ], [ \begin{smallmatrix} a \\ b \end{smallmatrix} ], [ \begin{smallmatrix} a \\ \bullet \end{smallmatrix} ]\} \downarrow$ . Let us extend the notation to  $R \subseteq 2^{\text{iPoms}}$  with  $R \downarrow = \{L \downarrow \mid L \in R\}$ .

Previous work [19] defines *down-closed rational operations*  $\cup, * \downarrow, \parallel \downarrow$  and (Kleene plus)  $^{+ \downarrow}$  for languages of finite ipomsets, as follows:

- $L * \downarrow M = \{P * Q \mid P \in L, Q \in M, T_P = S_Q\} \downarrow$ ,
- $L \parallel \downarrow M = \{P \parallel Q \mid P \in L, Q \in M\} \downarrow$ ,
- $L^{+ \downarrow} = \bigcup_{n \geq 1} L^n$ , with  $L^1 = L$  and  $L^{n+1} = L * \downarrow L^n$ .

The class  $D\text{-}\mathcal{Rat}$  of *down-closed rational languages* is then defined to be the smallest class that contains  $\emptyset, \{\text{id}_\emptyset\}, \{a\}, \{\bullet a\}, \{a \bullet\}, \{\bullet a \bullet\}$  (for  $a \in \Sigma$ ), and is closed under the operations above. Note that the explicit downclosure  $\downarrow$  of the operations above ensures that the built languages contain only interval ipomsets.<sup>3</sup> We will below introduce rational operations which *do not* apply down-closure; to distinguish them from the ones above we have added a subscript  $\downarrow$ .

► **Theorem 17** ([19]). *A language in  $2^{\text{iiPoms}}$  can be recognized by an HDA iff it is in  $D\text{-}\mathcal{Rat}$ .*

<sup>3</sup> For  $L = \{ac\}$  and  $M = \{\bullet bd \bullet\}$ ,  $(ac \parallel \bullet bd \bullet) \notin L \parallel \downarrow M$  but, for example, the ipomsets of Fig 2 are in  $L \parallel \downarrow M$ .

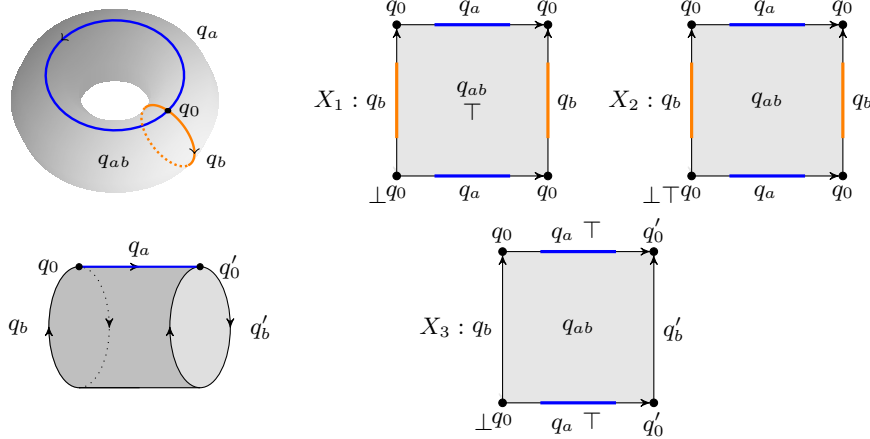
### 3.2 $\omega$ -higher-dimensional automata

An  $\omega$ -HDA is an HDA whose tracks are infinite. Formally, an  $\omega$ -track is an infinite sequence  $\alpha = (q_0, \varphi_1, q_1, \dots)$  where  $q_i \in X$  and each  $(q_i, \varphi_i, q_{i+1})$  is an upstep or a downstep. The event ipomset of an  $\omega$ -track is defined similarly using the infinite gluing. From this we may directly introduce the classical acceptance conditions of  $\omega$ -automata theory:

Let  $\text{Inf}(\alpha) = \{q \in X \mid |\{i \mid q_i = q\}| = +\infty\}$  be the set of cells seen infinitely often by  $\alpha$ . A *Büchi*  $\omega$ -HDA is an HDA  $(X, \perp_X, \top_X)$  where  $\perp_X, \top_X \subseteq X$  are the sets of *starting* and *accepting* cells, and where an  $\omega$ -track  $\alpha$  is accepting if  $\text{src}(\alpha) \in \perp_X$  and  $\text{Inf}(\alpha) \cap \top_X \neq \emptyset$ . Similarly, a *Muller*  $\omega$ -HDA is an HDA  $(X, \perp_X, F_X)$  where  $\perp_X \subseteq X$  is the set of *starting* cells and  $F_X \subseteq 2^X$  is the set of *accepting sets* of cells, and where an  $\omega$ -track  $\alpha$  is accepting if  $\text{src}(\alpha) \in \perp_X$  and  $\text{Inf}(\alpha) \in F_X$ .

► **Example 18.** Büchi  $\omega$ -HDAs  $X_1, X_2$  and  $X_3$  are defined in Fig. 8. Cells of the same color are identified in each  $\omega$ -HDA, and so are their respective faces (as required by the definition of precubical sets). Observe that  $X_1$  and  $X_2$  form a torus and  $X_3$  a cylinder. Their languages will be specified later in the paper (see Ex. 35 and Ex. 40).

► **Example 19.** Let  $X_4$  be the torus HDA of Fig. 8 and assume  $\perp_{X_4} = \{q_0\}$  and  $F_{X_4} = \{\{q_0, q_{ab}\}\}$ . Then  $(X_4, \perp_{X_4}, F_{X_4})$  defines a Muller  $\omega$ -HDA. The  $\omega$ -track  $\alpha = (q_0, \nearrow^a, q_a, \searrow_a, q_0, \nearrow^{ab}, q_{ab}, \searrow_{ab}, q_0, \nearrow^{ab}, q_{ab}, \searrow_{ab}, q_0, \dots)$  of event  $\text{ev}(\alpha) = a \cdot \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]^\omega$  is accepting as  $\text{Inf}(\alpha) = \{q_0, q_{ab}\} \in F_{X_4}$ .



■ **Figure 8** Büchi  $\omega$ -HDAs (left) and unfolded views (right). See Ex. 18.

The *language* of a Büchi  $\omega$ -HDA  $X$  is  $L(X) = \{\text{ev}(\alpha) \mid \text{src}(\alpha) \in \perp_X \text{ and } \text{Inf}(\alpha) \cap \top_X \neq \emptyset\}$  and we call *Büchi  $\omega$ -regular* such a language. Similarly, for a Muller  $\omega$ -HDA  $Y$  we have  $L(Y) = \{\text{ev}(\alpha) \mid \text{src}(\alpha) \in \perp_Y \text{ and } \text{Inf}(\alpha) \in F_Y\}$  and we call *Muller  $\omega$ -regular* such a language. As in classical  $\omega$ -theory, a Büchi  $\omega$ -HDA  $X_B$  can be seen as a Muller  $\omega$ -HDA  $X_M$ , where the acceptance condition is modified to  $F_{X_M} = \{A \subseteq X \mid A \cap \top_X \neq \emptyset\}$ . Hence Büchi  $\omega$ -regular languages are Muller  $\omega$ -regular. However, unlike HDAs over finite ipomsets we have:

► **Proposition 20.**  *$\omega$ -regular languages may not be down-closed.*

**Proof.** In the Büchi  $\omega$ -HDA  $X_1$  of Ex. 18, we have  $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]^\omega = \text{ev}(q_0, \nearrow^{ab}, q_{ab}, \searrow_{ab}, q_0, \dots) \in L(X_1)$  (as it goes infinitely many times through  $q_{ab} \in \top_{X_1}$ ). But  $(ab)^\omega \sqsubseteq \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]^\omega$ , and the only  $\omega$ -track whose event is this  $\omega$ -ipomset is  $(q_0, \nearrow^a, q_a, \searrow_a, q_0, \nearrow^b, q_b, \searrow_b, q_0, \dots)$  which is not accepting. Thus  $(ab)^\omega \notin L(X_1)$ . The same can be done for the Muller  $\omega$ -HDA  $X_4$  of Ex. 19. ◀

## 4 $\omega$ -rational languages

We want to extend the notion of rationality to languages of interval  $\omega$ -ipomsets and capture the expressiveness of  $\omega$ -HDAs. Thus, we introduce a non-nested  $\omega$ -iteration over languages of finite interval ipomsets. On the other hand, the rational operations defined in [19] take subsumption into account. As languages of  $\omega$ -HDA are not down-closed, we cannot just extend these to  $\omega$ -ipomsets, but need to redefine them without subsumption closure.

We define the  $\omega$ -rational operations over  $2^{\text{iiPoms}^\infty}$  by:

- For  $(L, M) \subseteq \text{iiPoms}^\infty \times \text{iiPoms}^\infty$ :  $L + M = L \cup M$
- For  $(L, M) \subseteq \text{iiPoms} \times \text{iiPoms}^\infty$ :  $L * M = LM = \{P * Q \mid P \in L, Q \in M, T_P \cong S_Q\}$
- For  $L \subseteq \text{iiPoms}$ :  $L^+ = \bigcup_{n \geq 1} L^n$  (with  $L^1 = L$  and  $L^{n+1} = L * L^n$ )
- For  $L \subseteq \text{iiPoms}$ :  $L^\omega = \{P_1 * P_2 * \dots \mid \forall i, P_i \in L, T_{P_i} \cong S_{P_{i+1}} \text{ and } |\{i : P_i \notin \text{Id}\}| = +\infty\}$

As in the finite case, the Kleene iteration is the non-empty  $+$  instead of  $*$  because  $L^0 = \text{Id}$  is not regular since it has infinite width. In addition, the operation  $\parallel$  defined over languages of finite interval ipomsets (and over finite [28] and  $\omega$ -series-parallel pomsets [31]) is not used in our case. The reason is that it would typically not give interval  $\omega$ -ipomsets ( $a^\omega \parallel b^\omega$  for example). In [19, Prop. 16], it is shown that  $D\text{-Rat}$  may also be obtained from  $\emptyset$  and  $\Xi$  (the set of starters and terminators over  $\Sigma$ ) using down-closed rational operations *without*  $\parallel$ . We define  $\omega$ -rational languages  $\text{Rat}^\infty$  as the smallest set such that:

- $\emptyset \in \text{Rat}^\infty$  and for  $U \in \Xi$ ,  $\{U\} \in \text{Rat}^\infty$ ;
- if  $L, M \in \text{Rat}^\infty$ , then  $L + M \in \text{Rat}^\infty$ ;
- if  $L, M \in \text{Rat}^\infty$  and  $L \subseteq \text{iPoms}$ , then  $L * M \in \text{Rat}^\infty$ ;
- if  $L \in \text{Rat}^\infty$  and  $L \subseteq \text{iPoms}$ , then  $L^+ \in \text{Rat}^\infty$  and  $L^\omega \in \text{Rat}^\infty$ .

We will often use  $U$  instead of  $\{U\}$  for ease of reading. We define the *width of a language*  $L$  by  $\text{wd}(L) = \sup\{\text{wd}(P) \mid P \in L\}$ . As they are built inductively from  $\Xi$ , using finitely many  $\omega$ -rational operations, all  $\omega$ -rational languages have finite width.

► **Proposition 21.** For  $L \in \text{Rat}^\infty$ ,  $\text{wd}(L) < +\infty$ .

Thus, Prop. 21 and Lem. 4 imply that  $\text{Rat}^\infty \subseteq 2^{\text{iiPoms}^\infty}$ , that is,  $\omega$ -rational operations preserve validity. In addition, since  $\Xi \subseteq \text{iiPoms}$  and by Cor. 13, we have:

► **Proposition 22.** For all  $L \in \text{Rat}^\infty$ , we have  $L \subseteq \text{iiPoms}^\infty$ .

We write  $\text{Rat}^\omega = \text{Rat}^\infty \cap 2^{\text{iiPoms}^\omega}$  for  $\omega$ -rational languages, and  $\text{Rat} = \text{Rat}^\infty \cap 2^{\text{iiPoms}}$  for finite ones. Note that  $\text{Rat}$  is not the class of rational languages defined in [19] (that we denote  $D\text{-Rat}$  for down-closed rational). For example,  $\{[\begin{smallmatrix} a \\ \bullet \end{smallmatrix}]\} * \{[\begin{smallmatrix} \bullet \\ a \end{smallmatrix}]\} = \{[\begin{smallmatrix} a \\ \bullet \\ \bullet \\ a \end{smallmatrix}]\} \in \text{Rat}$  as the gluing of a starter and a terminator, but is not in  $D\text{-Rat}$  (because not down-closed). We show in Prop. 36 that up to down-closure, both notions are equivalent in the finite case.

## 5 $\omega$ -rationality vs. $\omega$ -regularity

In this section, we explore the connections between  $\omega$ -rationality and  $\omega$ -regularity. We show that  $\omega$ -regular languages are  $\omega$ -rational. The proof relies on a type of classical  $\omega$ -automata derived from  $\omega$ -HDAs called *ST-automata*. On the other hand, the opposite does not hold. Indeed, we show that Muller acceptance is more expressive than Büchi acceptance and that there are  $\omega$ -rational languages that are not Muller  $\omega$ -regular.

### 5.1 ST-automata for $\omega$ -HDA

An ST-automaton is a plain  $(\omega)$ -automaton over  $\Xi$ , with an additional labeling of states, that can produce  $(\omega)$ -iipomsets if the letters are glued. It especially can mimic the behavior of an  $(\omega)$ -HDA with a canonical translation. We use here the most recent definition of these objects, introduced in [4], which subsumes other variants that have been used in [2, 3, 7, 8, 16, 17, 19].

We let  $\bar{\Xi} = (\bar{\Xi}, \cdot, \varepsilon)$  denote the free monoid on  $\bar{\Xi} = \Xi$  seen as an alphabet, using concatenation instead of gluing and  $\varepsilon$  for the empty word (which is different from the letter  $\text{id}_\emptyset \in \bar{\Xi}$ ).

An *ST-automaton*  $A$  is a finite labeled automaton  $(Q, E, I, F, \lambda)$  over the (infinite) alphabet  $\bar{\Xi}$ , where  $Q$  is the set of states,  $E \subseteq Q \times (\bar{\Xi} \setminus \text{Id}) \times Q$  is the set of transitions,  $I \subseteq Q$  are the initial states,  $F$  is an acceptance condition and  $\lambda : Q \rightarrow \square$  is a labeling of states that is coherent with  $E$ , meaning that for all  $(p, sUT, q) \in E$ ,  $\lambda(p) = S$  and  $\lambda(q) = T$ .

A *path*  $\pi = (q_0, e_0, q_1, e_1, \dots, e_{n-1}, q_n)$  in an ST-automaton  $A$  is an alternation of states and transitions such that  $e_i = (q_i, P_i, q_{i+1}) \in E$ . Its label is  $l(\pi) = \text{id}_{\lambda(q_0)} \cdot P_0 \cdot \text{id}_{\lambda(q_1)} \cdot \dots \cdot P_{n-1} \cdot \text{id}_{\lambda(q_n)}$  seen as a word. We say that  $\pi$  is accepting if  $q_0 \in I$  and  $q_n \in F$ . The language of an ST-automaton  $A$ , denoted  $L(A)$ , is the set of labels of its accepting paths. The same can be done for  $\omega$ -paths, which are accepted according to a Büchi or Muller condition.

► **Remark 23.** Path labels of ST-automata are elements of  $\text{Id} \cdot (\bar{\Xi} \cdot \text{Id})^*$  (or  $\text{Id} \cdot (\bar{\Xi} \cdot \text{Id})^\omega$  in the infinite case). In particular, the labeling of states and their consideration in the path labels forbid to have the empty word  $\varepsilon$  as the label of a path. Note also that labels of states are not used twice: given  $\pi_1 = (q_0, e_0, \dots, e_{i-1}, q_i)$  and  $\pi_2 = (q_i, e_i, \dots, e_{n-1}, q_n)$ , we have  $l(\pi_1) \cdot l(\pi_2) = \text{id}_{\lambda(q_0)} \cdot P_0 \cdot \text{id}_{\lambda(q_1)} \cdot \dots \cdot P_{i-1} \cdot \text{id}_{\lambda(q_i)} \cdot P_{i+1} \cdot \dots \cdot P_{n-1} \cdot \text{id}_{\lambda(q_n)}$ . We are thus only able to concatenate  $l(\pi_1)$  and  $l(\pi_2)$  if the end of  $\pi_1$  has the same label as the start of  $\pi_2$ . We will consider  $(\omega)$ -rational expressions of ST-automata as classical  $(\omega)$ -rational expressions, the only difference being that the concatenation operator behaves as stated above.

In the following, we build an ST-automaton  $ST(X)$  from an  $(\omega)$ -HDA  $X$ , following [4]. The intuition is that each cell (of any dimension) becomes a state, and for each upstep  $(p = \delta_A^0(q), \nearrow^A, q)$  resp. downstep  $(p, \searrow_B, \delta_B^1(p) = q)$  in  $X$ , a transition is introduced from the state corresponding to  $p$  to the one corresponding to  $q$ , labeled with  $A \uparrow \text{ev}(q)$  resp.  $\text{ev}(p) \downarrow_B$ , to mimic the behavior of the HDA. More formally, for an HDA  $(X, \perp_X, F_X)$ , we associate the ST-automaton  $ST(X) = (Q, E, I, F, \lambda)$  with:

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

By construction, there is a one-to-one correspondence between tracks in  $X$  and paths in  $ST(X)$ : with a track  $\alpha = (q_0, \varphi_0, q_1, \dots)$  of  $X$  we associate the path  $ST(\alpha) = (q_0, e_0, q_1, \dots)$  of  $ST(X)$  such that

- $(q_i, \varphi_i, q_{i+1}) = (\delta_{A_i}^0(q_{i+1}), \nearrow^{A_i}, q_{i+1}) \Rightarrow A_i \subseteq \text{ev}(q_{i+1})$ ,  $e_i = (q_i, A_i \uparrow \text{ev}(q_{i+1}), q_{i+1}) \in E$
- $(q_i, \varphi_i, q_{i+1}) = (q_i, \searrow_{A_i}, \delta_{A_i}^1(q_i)) \Rightarrow A_i \subseteq \text{ev}(q_i)$ ,  $e_i = (q_i, \text{ev}(q_i) \downarrow_{A_i}, q_{i+1}) \in E$

This proves the following lemma:

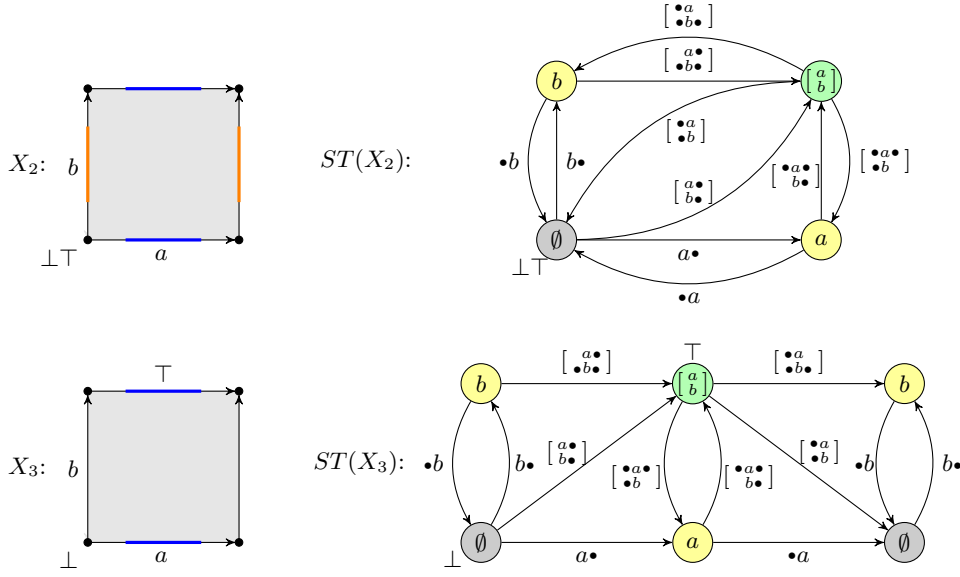
► **Lemma 24.** *A track  $\alpha$  is accepting in  $X$  if and only if  $ST(\alpha)$  is accepting in  $ST(X)$ .*

Since the labeling  $\lambda$  is coherent with  $E$ , a word produced by an ST-automaton  $A$  is *coherent* in the sense that if  $P_1 \cdot P_2 \cdot \dots$  is the label of a path in  $A$ , then  $T_{P_i} \cong S_{P_{i+1}}$  for all  $i \geq 0$  (and thus it can be seen as the gluing of starters and terminators). To be more precise, the set of coherent finite words over  $\bar{\Xi}$  is denoted  $\text{Coh}^+(\bar{\Xi}) = \{U_1 \cdot \dots \cdot U_n \mid U_i \in \bar{\Xi}, T_{U_i} \cong S_{U_{i+1}}\}$ .

For  $\omega$ -words we denote by  $\text{Coh}^\omega(\bar{\Xi})$  the set of coherent infinite words over  $\bar{\Xi}$ , defined by

$$\text{Coh}^\omega(\bar{\Xi}) = \{U_1 \cdot U_2 \cdot \dots \mid U_i \in \bar{\Xi}, \#\{i : U_i \notin \text{Id}\} = +\infty \text{ and } T_{U_i} = S_{U_{i+1}}\}$$

and as usual we write  $\text{Coh}^\infty(\bar{\Xi}) = \text{Coh}^+(\bar{\Xi}) \cup \text{Coh}^\omega(\bar{\Xi})$ .



■ **Figure 9** ST-automata of  $X_2$  and  $X_3$  of Ex. 18. Cells of dimension 0 are in gray, dimension 1 in yellow, and dimension 2 in green.

► **Lemma 25.** *Let  $A$  be an ST-automaton.*

- *If  $A$  accepts finite words then  $L(A) \subseteq \text{Coh}^+(\overline{\Xi})$ .*
- *If  $A$  accepts infinite words then  $L(A) \subseteq \text{Coh}^\omega(\overline{\Xi})$ .*

Thus, one can “glue” words accepted by an ST-automaton to obtain an  $(\omega)$ -iipomset. Let  $\Psi: \text{Coh}^\infty(\overline{\Xi}) \rightarrow \text{iPoms}^\infty$  be the *gluing function* defined by  $\Psi(P_1 \cdot P_2 \cdot \dots) = P_1 * P_2 * \dots$ . For a set  $L \subseteq \text{Coh}^\infty(\overline{\Xi})$ , we also define  $\Psi(L) = \{\Psi(P) \mid P \in L\}$ . In [4], the authors show the following for ST-automata derived from HDAs over finite iipomsets:

► **Proposition 26** ([4]). *For any HDA  $X$ ,  $L(X) = \Psi(L(ST(X)))$ .*

We show that the proposition holds also for  $\omega$ -HDAs. First, since the labeling of states is coherent with the transitions, we have the following by applying definitions.

► **Lemma 27.** *For all  $\omega$ -tracks  $\alpha$  of  $X$ , we have  $\text{ev}(\alpha) = \Psi(l(ST(\alpha)))$ .*

We can now show the  $\omega$ -equivalent of Prop. 26, by combining Lem. 24 and Lem. 27:

► **Proposition 28.** *For any (Büchi or Muller)  $\omega$ -HDA  $X$ ,  $L(X) = \Psi(L(ST(X)))$ .*

## 5.2 $\omega$ -HDAs are $\omega$ -rational

The formalism of ST-automata provides a strong tool for HDAs. It allows to use classical theorems of  $\omega$ -automata. This helps in particular to show that  $\omega$ -regular languages are  $\omega$ -rational as we will see in this section.

Indeed, the language of any ST-automaton  $A$  is also the one of a rational expression  $e_A$ , using the operations  $+$ ,  $\cdot$ ,  $*$  and letters  $U_i \in \overline{\Xi}$ . Similarly, given an ST-automaton  $A'$  over infinite words, one can effectively build an equivalent  $\omega$ -rational expression  $e_{A'}$  [36].

A  $(\omega)$ -rational expression is said *positive* if it has no occurrences of  $*$  and  $\varepsilon$ . Since ST-automata do not accept  $\varepsilon$ , ST-automata languages can be expressed by positive expressions:

► **Lemma 29.** *Any rational expression  $e$  of an ST-automaton can be transformed into a positive rational expression  $\bar{e}$  such that  $\mathcal{L}(e) = \mathcal{L}(\bar{e})$ .*

We define inductively a function  $\Psi'$  transforming a positive rational expression over  $\Xi$  into a set of ipomsets in the usual way by replacing  $\cdot$  by  $*$ . We directly have:

- **Lemma 30.** *Let  $e(U_1, \dots, U_n)$  be a positive rational expression over  $\Xi$ .*
- *If  $e$  is rational, then  $\Psi'(e) \in \mathcal{Rat}$ .*
  - *If  $e$  is  $\omega$ -rational, then  $\Psi'(e) \in \mathcal{Rat}^\omega$ .*

In addition, we can show by induction that  $\Psi'$  preserves languages.

► **Lemma 31.** *If  $e$  is an  $(\omega)$ -rational expression over  $\Xi$ , then  $\Psi'(e) = \Psi(\mathcal{L}(e))$*

Thus, languages of ST-automata are rational:

- **Lemma 32.** *Let  $A$  be an ST-automaton.*
- *If  $A$  accepts finite words, then  $\Psi(L(A)) \in \mathcal{Rat}$ .*
  - *If  $A$  accepts infinite words, then  $\Psi(L(A)) \in \mathcal{Rat}^\omega$ .*

► **Theorem 33.** *Büchi and Muller  $\omega$ -regular languages are  $\omega$ -rational.*

**Proof.** Let  $(X, \perp_X, F_X)$  be a Büchi or Muller  $\omega$ -HDA. By definition of  $\omega$ -HDAs,  $X$  is finite, hence so is its ST-automaton  $ST(X)$ . By Prop. 28,  $L(X) = \Psi(L(ST(X)))$ , and by Lem. 32,  $\Psi(L(ST(X))) \in \mathcal{Rat}^\omega$ , thus all  $\omega$ -regular languages are  $\omega$ -rational. ◀

As a corollary of the above and Prop. 22 we have:

► **Corollary 34.** *If  $L$  is an  $\omega$ -regular language, then  $\text{wd}(L) < +\infty$  and  $L \subseteq \text{iiPoms}^\omega$ .*

► **Example 35.** In Fig. 9 are represented two  $\omega$ -HDAs with their corresponding ST-automata. From this we can directly compute their language in  $\Xi^\omega$  (we omit the letters in  $\text{Id}$ ), and then use  $\Psi$  to compute the languages of the original Büchi  $\omega$ -HDAs:

$$\begin{aligned} L(ST(X_2)) &= \left( a \cdot \cdot a + b \cdot \cdot b + (a \cdot \cdot [\cdot^a \cdot] + [\cdot^a \cdot] + b \cdot \cdot [\cdot^a \cdot]) \cdot ([\cdot^a \cdot] \cdot [\cdot^a \cdot] + [\cdot^a \cdot] \cdot [\cdot^a \cdot])^* \right. \\ &\quad \left. \cdot ([\cdot^a \cdot] \cdot a + [\cdot^a \cdot] + [\cdot^a \cdot] \cdot b) \right)^\omega \\ L(ST(X_3)) &= (b \cdot \cdot b)^* \cdot (a \cdot \cdot [\cdot^a \cdot] + b \cdot \cdot [\cdot^a \cdot] + [\cdot^a \cdot]) \cdot ([\cdot^a \cdot] \cdot [\cdot^a \cdot])^\omega \end{aligned}$$

Languages in  $\text{Coh}^\omega(\Xi)$  can be turned into the language of the original HDA using the function  $\Psi$  and Prop. 26. We just replace the occurrences of  $A^*$  by  $A^+ + \varepsilon$  and develop the expression to avoid the word  $\varepsilon$ , then remove multiple occurrences of the same ipomset in unions:

- $L(X_2) = \Psi(L(ST(X_2))) = (a + b + [\cdot^a \cdot] + [\cdot^a \cdot] * ([\cdot^a \cdot] + [\cdot^a \cdot])^+ * [\cdot^a \cdot])^\omega$
- $L(X_3) = \Psi(L(ST(X_3))) = b^+ * [\cdot^a \cdot] * [\cdot^a \cdot]^\omega + [\cdot^a \cdot] * [\cdot^a \cdot]^\omega$

As we saw in the previous section, our definition of rational operations is different from the original one [19] since we do not take subsumption closure into account. Nevertheless:

► **Proposition 36.**  $\mathcal{Rat}_\downarrow = D\text{-}\mathcal{Rat}$ .

**Proof.** The inclusion  $\mathcal{Rat}_\downarrow \subseteq D\text{-}\mathcal{Rat}$  is done by induction. The other follows from Lem. 32 and the fact that languages of  $D\text{-}\mathcal{Rat}$  are closed under subsumption (see App. B for details). ◀

### 5.3 Muller vs Büchi vs $\omega$ -rational

In the classical  $\omega$ -theory, rational languages are as expressive as Muller and Büchi automata. For  $\omega$ -HDA, these are no longer true. We describe and prove here some differences:

► **Theorem 37.** *The Muller  $\omega$ -regular class is strictly bigger than the Büchi  $\omega$ -regular class.*

**Proof.** As for the classical  $\omega$ -theory, Büchi  $\omega$ -regular languages are Muller  $\omega$ -regular.

However, the converse is false. Take the Muller  $\omega$ -HDA  $X_4$  of Ex. 19. The only cycle (up to shift) using only states of  $F_{X_4}$  is  $(q_0, \nearrow^{ab}, q_{ab}, \searrow_{ab}, q_0, \dots)$ , so the language of  $X_4$  is such that every accepted  $\omega$ -ipomset ends by  $[\frac{a}{b}]^\omega$ . Assume that there is a Büchi  $\omega$ -HDA  $Y$  that recognizes  $L(X_4)$ . Let  $\alpha$  be an accepting  $\omega$ -track of  $Y$ , which exists because  $L(X_4) \neq \emptyset$ . We have  $\text{ev}(\alpha) = A * [\frac{a}{b}]^\omega$  for some  $A \in \text{iiPoms}$ . So  $\alpha$  goes through  $Y[[\frac{a}{b}]]$  an infinite number of times. However  $Y$  is finite, so there is  $p \in Y[[\frac{a}{b}]]$  such that  $\alpha$  goes through  $p$  an infinite number of times. Thus there is a subtrack  $\alpha_p := (\delta_{ab}^0(p), \nearrow^{ab}, p, \searrow_{ab}, \delta_{ab}^1(p))$  which appears an infinite number of times in  $\alpha$ . (It may be a track equivalent to it, but as there is a finite number of them, it suffices to choose one of them that is used an infinite number of times.) As  $Y$  is a precubical set, we can define the track  $\alpha'_p = (\delta_{ab}^0(p), \nearrow^a, \delta_b^0(p), \searrow_a, \delta_a^1(\delta_b^0(p)), \nearrow^b, \delta_a^1(p), \searrow_b, \delta_{ab}^1(p))$  with label  $ab$ , and then replace one in two occurrences of  $\alpha_p$  by  $\alpha'_p$ . The cells visited an infinite number of times by  $\alpha$  are still visited an infinite number of times, hence the new  $\omega$ -track is also accepting. However, its label does not end by  $[\frac{a}{b}]^\omega$  because  $ab$  appears an infinite number of times. So  $L(X_4)$  is not Büchi  $\omega$ -regular. ◀

► **Theorem 38.** *Some languages of  $\text{Rat}^\omega$  cannot be expressed by Muller  $\omega$ -HDA.*

**Proof.** Let  $L_r = ([\frac{a}{b}] * [\frac{a}{b}])^\omega = [\frac{a}{b}]^\omega \in \text{Rat}^\omega$ . Suppose that  $(X, \perp_X, F_X)$  is a Muller  $\omega$ -HDA such that  $L(X) = L_r$ . An accepting  $\omega$ -track  $\alpha$  (with label  $\text{ev}(\alpha) = [\frac{a}{b}]^\omega$ ) in  $X$  must start by a track (equivalent to)  $\alpha_p := (\delta_{ab}^0(p), \nearrow^{ab}, p, \searrow_{ab}, \delta_{ab}^1(p))$  with  $p \in X[[\frac{a}{b}]]$ . As  $X$  is a precubical set,  $\alpha'_p = (\delta_{ab}^0(p), \nearrow^a, \delta_b^0(p), \searrow_a, \delta_a^1(\delta_b^0(p)), \nearrow^b, \delta_a^1(p), \searrow_b, \delta_{ab}^1(p))$  is well-defined. If  $\beta$  is such that  $\alpha = \alpha_p * \beta$ , then  $\alpha' = \alpha'_p * \beta$  is a track in  $X$  which is accepting (as Muller acceptance conditions only care about states seen an infinite number of times). Then  $\text{ev}(\alpha') = ab[\frac{a}{b}]^\omega \in L(X)$  but  $ab[\frac{a}{b}]^\omega \notin L_r$ , and we conclude by contradiction. ◀

Note that defining  $\omega$ -rational operations with subsumption closure (following [19]) would have made  $\omega$ -rational languages *less* expressive than  $\omega$ -HDAs. For example, as  $\{[\frac{a}{b}]\} \downarrow = \{ab, ba, [\frac{a}{b}]\}$ , the language  $L(X_4)$  of Ex. 19 would then not be  $\omega$ -rational.

As  $\text{Rat}^\omega$  is more expressive than Büchi and Muller  $\omega$ -HDAs, it is natural to inquire about the form of languages of  $\text{Rat}^\omega$  with the same expressiveness. As a first step, define a *locally down-closed lasso* (*ldl*)  $\omega$ -language to be a language of the form  $\bigcup_{i \leq n} M_i \downarrow * ((R_i^+) \downarrow)^\omega$  for  $n \in \mathbb{N}$  and  $M_i, R_i \in \text{Rat}$  for all  $i \leq n$ .

► **Theorem 39.** *Any Büchi regular  $\omega$ -language is a locally down-closed lasso  $\omega$ -language.*

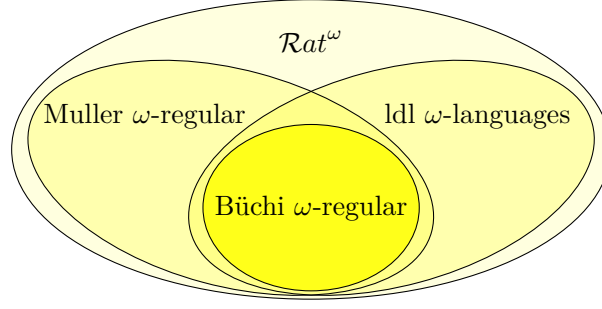
Thus,  $\omega$ -HDAs allow local subsumptions in some finite factors coming from  $M_i$  and  $R_i^+$ . Using  $R_i^+$  instead of  $R_i$  more accurately reflects the behaviour of Büchi  $\omega$ -HDAs. Indeed, for  $R = \{[\frac{a}{b}]\}$  and  $M = [\frac{a}{b}]$ ,  $M$  and  $R$  are down-closed, but  $M * R^\omega = M \downarrow * (R \downarrow)^\omega = \{[\frac{a}{b}]\}^\omega$  which cannot be recognized by a Büchi  $\omega$ -HDA as seen in the proof of Th. 37.

► **Example 40.** We can use again Ex. 35, and take  $L(X_3)$  which is Büchi  $\omega$ -regular:

$$L(X_3) = b^+ * [\frac{a}{b}] * [\frac{a}{b}]^\omega + [\frac{a}{b}] * [\frac{a}{b}]^\omega = (b^+ + \text{id}_\emptyset) * [\frac{a}{b}] * ([\frac{a}{b}]^+)^\omega$$

For  $M = (b^+ + \text{id}_\emptyset) * [\frac{a}{b}]$  and  $R = [\frac{a}{b}]$ , we have  $L(X_3) = M \downarrow * ((R^+) \downarrow)^\omega$ .





■ **Figure 10** Inclusion of classes of  $\omega$ -languages, with  $\text{Büchi} \subsetneq \text{Muller} \subsetneq \text{Rat}^\omega$ .

## 6 Conclusion and future work

We have defined interval  $\omega$ -pomsets with interfaces ( $\omega$ -iipomsets), shown that isomorphisms between them are unique, and that they admit a unique sparse decomposition. Then, we have introduced  $\omega$ -higher-dimensional automata ( $\omega$ -HDAs) over  $\omega$ -iipomsets as a generalization of HDAs over finite iipomsets. We have studied their languages under Büchi and Muller acceptance conditions and demonstrated that Muller is more expressive than Büchi.

Unlike in the finite case, languages of  $\omega$ -HDAs are not closed under subsumption. In pursuit of a Kleene-like theorem, we have adapted the rational operations on finite iipomsets to disregard subsumption, showing that up to subsumption, the notion of rational language remains unchanged under this modification. We have also introduced a non-nested  $\omega$ -iteration to these operations and defined  $\omega$ -rational languages over  $\omega$ -iipomsets. However, we show that this natural class is bigger than the one of  $\omega$ -iipomset languages recognized by  $\omega$ -HDAs. The diagram in Fig. 10 summarizes our results about  $\omega$ -HDAs and  $\omega$ -rational languages. In the diagram, the inclusion of Büchi  $\omega$ -regular languages into locally down-closed lasso  $\omega$ -languages is shown as strict (cf. Th. 39); but we conjecture it to be an equivalence.

The challenge in establishing a Kleene-like theorem arises from the precubical identities that force some factors to be subsumption-closed. A solution may be found by passing to the *partial* HDAs of [13, 21]. Intuitively, partial HDAs relax the face maps to be partial functions, satisfying precubical inclusions rather than precubical identities. However, while the language theory of HDAs has seen much recent progress, partial HDAs remain largely unexplored. The use of such objects is particularly relevant, as with our adapted rational operations, we have shown that the language of any ST-automaton is ( $\omega$ )-rational, which is *not* the case for the subsumption-closed rational operations of [19]. This notably implies that “our” rational languages are closed under bounded-width complement, whereas the original definition is only closed under bounded pseudocomplement [7].

This work represents a first step toward developing a theory of  $\omega$ -HDAs. Such a theory should be well-suited for modeling non-terminating concurrent systems with dependence and independence relations. In particular it would avoid some problems with state-space explosion; for example, an infinite execution where event  $a$  must precede  $b$  is modeled as the subsumption closure of the pomset with  $a < b$  and all other events occurring in parallel, instead of considering all interleavings separately. Developing a logical characterization, similar to the finite case [5], would also be of interest.

## References

- 1 Rajeev Alur, Caleb Stanford, and Christopher Watson. A robust theory of series parallel graphs. *Proc. ACM Program. Lang.*, 7(POPL):1058–1088, 2023. doi:10.1145/3571230.
- 2 Amazigh Amrane, Hugo Bazille, Emily Clement, and Uli Fahrenberg. Languages of higher-dimensional timed automata. In Lars Michael Kristensen and Jan Martijn E. M. van der Werf, editors, *Application and Theory of Petri Nets and Concurrency - 45th International Conference, PETRI NETS 2024, Geneva, Switzerland, June 26-28, 2024, Proceedings*, volume 14628 of *Lecture Notes in Computer Science*, pages 197–219. Springer, 2024. doi:10.1007/978-3-031-61433-0\_10.
- 3 Amazigh Amrane, Hugo Bazille, Emily Clement, Uli Fahrenberg, and Philipp Schlehuber-Caissier. Higher-dimensional timed automata for real-time concurrency. In Andreas Rauh, Bernd Finkbeiner, and Paul Kröger, editors, *Design and Verification of Cyber-Physical Systems*. Springer, 2025. To be published. <https://arxiv.org/abs/2401.17444>.
- 4 Amazigh Amrane, Hugo Bazille, Emily Clement, Uli Fahrenberg, and Krzysztof Ziemiański. Presenting interval pomsets with interfaces. In Uli Fahrenberg, Wesley Fussner, and Roland Glück, editors, *Relational and Algebraic Methods in Computer Science - 21st International Conference, RAMiCS 2024, Prague, Czech Republic, August 19-22, 2024, Proceedings*, volume 14787 of *Lecture Notes in Computer Science*, pages 28–45. Springer, 2024. doi:10.1007/978-3-031-68279-7\_3.
- 5 Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Marie Fortin. Logic and languages of higher-dimensional automata. In *International Conference on Developments in Language Theory*, pages 51–67. Springer, 2024. doi:10.1007/978-3-031-66159-4\_5.
- 6 Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, Loïc Hélouët, and Philipp Schlehuber-Caissier. Petri nets and higher-dimensional automata, 2025. Accepted for PETRI NETS. doi:10.48550/arXiv.2502.02354.
- 7 Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. In Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa, editors, *ICTAC*, volume 14446 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2023. doi:10.1007/978-3-031-47963-2\_18.
- 8 Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. *Theoretical Computer Science*, 1036:115156, 2025. doi:10.1016/j.tcs.2025.115156.
- 9 Marek A. Bednarczyk. *Categories of Asynchronous Systems*. PhD thesis, University of Sussex, UK, 1987.
- 10 J Richard Büchi. Symposium on decision problems: On a decision method in restricted second order arithmetic. In *Studies in Logic and the Foundations of Mathematics*, volume 44, pages 1–11. Elsevier, 1966.
- 11 Olivier Carton, Dominique Perrin, and Jean-Eric Pin. Automata and semigroups recognizing infinite words. *Logic and automata*, 2:133–168, 2008.
- 12 Volker Diekert and Anca Muscholl. Deterministic asynchronous automata for infinite traces. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 617–628. Springer, 1993. doi:10.1007/3-540-56503-5\_61.
- 13 Jérémy Dubut. Trees in partial higher dimensional automata. In Mikołaj Bojańczyk and Alex Simpson, editors, *FOSSACS*, volume 11425 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2019. doi:10.1007/978-3-030-17127-8\_13.
- 14 Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. From Spot 2.0 to Spot 2.10: What's new? In *34th International Conference on Computer Aided Verification (CAV'22)*, volume 13372 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 2022. doi:10.1007/978-3-031-13188-2\_9.

- 15 Werner Ebinger and Anca Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154(1):67–84, 1996. doi:10.1016/0304-3975(95)00130-1.
- 16 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. doi:10.1016/j.ifacol.2018.08.019.
- 17 Uli Fahrenberg. Higher-dimensional timed and hybrid automata. *Leibniz Transactions on Embedded Systems*, 8(2):03:1–03:16, 2022. doi:10.4230/LITES.8.2.3.
- 18 Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021. doi:10.1017/S0960129521000293.
- 19 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. doi:10.4230/LIPIcs.CONCUR.2022.29.
- 20 Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285(B):104914, 2022. doi:10.1016/j.ic.2022.104914.
- 21 Uli Fahrenberg and Axel Legay. Partial higher-dimensional automata. In Lawrence S. Moss and Pawel Sobocinski, editors, *CALCO*, volume 35 of *LIPIcs*, pages 101–115. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CALCO.2015.101.
- 22 Uli Fahrenberg and Krzysztof Ziemiański. A Myhill-Nerode theorem for higher-dimensional automata. In Luís Gomes and Robert Lorenz, editors, *PETRI NETS*, volume 13929 of *Lecture Notes in Computer Science*, pages 167–188. Springer, 2023. doi:10.1007/978-3-031-33620-1\_9.
- 23 Jean Fanchon and Rémi Morin. Pomset languages of finite step transition systems. In Giuliana Franceschinis and Karsten Wolf, editors, *PETRI NETS*, volume 5606 of *Lecture Notes in Computer Science*, pages 83–102. Springer, 2009. doi:10.1007/978-3-642-02424-5\_7.
- 24 Berndt Farwer. omega-automata. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2001. doi:10.1007/3-540-36387-4\_1.
- 25 Peter C. Fishburn. Intransitive indifference with unequal indifference intervals. *Journal of Mathematical Psychology*, 7(1):144–149, 1970. doi:10.1016/0022-2496(70)90062-3.
- 26 Paul Gastin, Antoine Petit, and Wiesław Zielonka. An extension of Kleene’s and Ochmański’s theorems to infinite traces. *Theoretical Computer Science*, 125(2):167–204, 1994. doi:10.1016/0304-3975(94)90254-2.
- 27 Blaise Genest, Dietrich Kuske, and Anca Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204(6):920–956, 2006. doi:10.1016/j.ic.2006.01.005.
- 28 Jay L Gischer. The equational theory of pomsets. *Theoretical Computer Science*, 61(2-3):199–224, 1988. doi:10.1016/0304-3975(88)90124-7.
- 29 Jan Grabowski. On partial languages. *Fundamentae Informaticae*, 4(2):427, 1981.
- 30 Kenneth Kunen. *Set Theory: An Introduction to Independence Proofs*. North-Holland, 1980.
- 31 Dietrich Kuske. Infinite series-parallel posets: Logic and languages. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, volume 1853 of *Lecture Notes in Computer Science*, pages 648–662. Springer, 2000. doi:10.1007/3-540-45022-X\_55.
- 32 K. Lodaya and P. Weil. A Kleene iteration for parallelism. In *Foundations of Software Technology and Theoretical Computer Science*, pages 355–366, 1998.

- 33 K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages. In *STACS'98*, volume 1373 of *Lecture Notes in Computer Science*, pages 555–565. Springer, 1998. doi:10.1007/BFB0028590.
- 34 K. Lodaya and P. Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1–2):347–380, 2000. doi:10.1016/S0304-3975(00)00031-1.
- 35 K. Lodaya and P. Weil. Rationality in algebras with a series operation. *Information and Computation*, 171:269–293, 2001. doi:10.1006/INCO.2001.3077.
- 36 Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and control*, 9(5):521–530, 1966. doi:10.1016/S0019-9958(66)80013-X.
- 37 David E Muller. Infinite sequences and finite machines. In *Proceedings of the Fourth Annual Symposium on Switching Circuit Theory and Logical Design (swct 1963)*, pages 3–16. IEEE Computer Society, 1963. doi:10.1109/SWCT.1963.8.
- 38 Luc Passemard, Amazigh Amrane, and Uli Fahrenberg. Higher-dimensional automata : Extension to infinite tracks. *CoRR*, abs/2503.07881, 2025. doi:10.48550/arXiv.2503.07881.
- 39 Dominique Perrin and Jean-Eric Pin. *Infinite words - automata, semigroups, logic and games*, volume 141 of *Pure and applied mathematics series*. Elsevier Morgan Kaufmann, 2004.
- 40 Vaughan R. Pratt. Modeling concurrency with partial orders. *Journal of Parallel Programming*, 15(1):33–71, 1986. doi:10.1007/BF01379149.
- 41 Vaughan R. Pratt. Modeling concurrency with geometry. In *POPL*, pages 311–322, New York City, 1991. ACM Press. doi:10.1145/99583.99625.
- 42 Rob J. van Glabbeek. Bisimulations for higher dimensional automata. Email message, June 1991. URL: <http://theory.stanford.edu/~rvg/hda>.
- 43 Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. doi:10.1016/j.tcs.2006.02.012.
- 44 Glynn Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, pages 325–392, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.
- 45 Glynn Winskel and Mogens Nielsen. Models for concurrency. In Samson Abramsky, Dov M. Gabbay, and Thomas S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4. Clarendon Press, Oxford, 1995.
- 46 Wiesław Zielonka. Notes on finite asynchronous automata. *RAIRO – Informatique Théorique et Applications*, 21(2):99–135, 1987. doi:10.1051/ita/1987210200991.

## A Complementary proofs of Section 2

**Proof of Proposition 3.** Let  $P$  be an  $\omega$ -ipomset and define  $P_0$  as the set of  $<_P$ -minimal elements of  $P$  and  $P_{i+1}$  the set of  $<_P$ -minimal elements of  $P \setminus \{P_0, \dots, P_i\}$  for all  $i < \omega$ . Note that each  $P_i$  is an  $<_P$ -antichain and linearly ordered by  $\dashrightarrow_P$ . Define the binary relation  $\mathcal{R}_P$  by  $x\mathcal{R}_P y$  iff  $(x \in P_i, y \in P_j \text{ and } i < j)$  or  $(x, y \in P_i \text{ and } x \dashrightarrow_P y)$ . Then  $(P, \mathcal{R}_P)$  is a strict well-order. In addition, an isomorphism between  $\omega$ -ipomsets  $P$  and  $Q$  is an isomorphism between  $(P, \mathcal{R}_P)$  and  $(Q, \mathcal{R}_Q)$ . We conclude by noting that well-ordering isomorphisms are unique (see [30, Lem. 6.2]). ◀

**Proof of Lemma 4.** Let  $(P_i)_{i \in \mathbb{N}} \in \text{iPoms}^{\mathbb{N}}$  such that  $S_{P_i} \cong T_{P_{i+1}}$ ,  $I = \{i \in \mathbb{N} \mid P_i \notin \text{Id}\}$  is infinite and there is  $m \in \mathbb{N}$  such that  $\text{wd}(P_i) \leq m$ . We want to show that it is a valid  $\omega$ -ipomset. It is routine to check that  $P = P_0 * P_1 * \dots$  is well-defined, however it can be a finite ipomset (for example for a constant sequence in  $\text{Id}$ ), or can define a not valid  $\omega$ -ipomset.

- Infinity: By contradiction, suppose  $P$  is finite. If so, there is an index  $i_0$  such that, for  $i > i_0$ ,  $P_i$  does not start any event. Let  $Q = P_0 * \dots * P_{i_0} \in \text{iPoms}$ . For  $i \in I$ ,  $P_i \notin \text{Id}$  so it has to start or finish at least one event. Let  $I' = I \cap \{i \in \mathbb{N} \mid i > i_0\}$ , as  $I$  is infinite,  $I'$  is infinite. For  $i \in I'$ ,  $P_i$  has to end at least one event of  $Q$ . But  $|Q|$  is finite and  $I'$  is infinite, contradiction.

- Finite past: Let  $z \in P$ , there is  $j$  s.t.  $z \in P_j$ . By definition of  $<_P$ , if  $y <_P z$  then there is  $i \leq j$  such that  $y \in P_i$ . So the  $<_P$ -predecessors of  $z$  are in  $(\bigcup_{i \leq j} (P_i, i))_{/x \sim f(x)}$  which is finite.
  - No infinite  $<_P$ -antichain: Let  $A$  be an  $<_P$ -antichain, then there is  $i$  such that  $A \subseteq P_i$ . So  $|A| \leq \text{wd}(P_i) \leq m$ , so there is no infinite  $<_P$ -antichain.
- Thus,  $P$  is a valid  $\omega$ -ipomset. ◀

The following lemma is needed for the proof of Prop. 5 (see [38] for a proof):

► **Lemma A.1.** *Let  $P \in \text{iPoms}^\omega$  and  $E$  be its set of maximal  $<_P$ -antichains. If  $\ll$  is a linear order on  $E$ , then  $(E, \ll) \cong (\mathbb{N}, <)$ .*

**Proof of Proposition 5.** We want to prove the equivalence between (1)  $P$  is an interval  $\omega$ -ipomset, (2)  $P$  has an interval representation and (3) the order  $\ll$  on maximal antichains is linear.

- (2)  $\Rightarrow$  (1) : Let  $w, x, y, z \in P$  with  $w <_P y$  and  $x <_P z$ . By (2) we take an interval representation  $b, e$ , then  $e(w) < b(y)$ ,  $e(x) < b(z)$ . Suppose that  $w \not<_P z$ , by (2)  $e(w) \not< b(z)$ , as  $<$  is linear on  $\mathbb{N} \cup \{+\infty\}$ ,  $b(z) \leq e(w)$ , and thus  $e(x) < b(y)$  so by (2),  $x <_P y$ . Thus,  $w <_P z$  or  $x <_P y$ .
  - (1)  $\Rightarrow$  (3) : Let  $A$  and  $B$  be two different maximal antichains and suppose that  $A \not\ll B$  and  $B \not\ll A$ . Then there is  $(x, y) \in A \times B$  such that  $x < y$  and  $(x', y') \in A \times B$  such that  $y' < x'$ . By (1) we have  $x < x'$  or  $y' < y$ , but  $A$  and  $B$  are  $<$ -antichains, contradiction.
  - (3)  $\Rightarrow$  (2) : By Lem. A.1, we can order the set  $E$  of maximal antichains of  $P$  such that  $E = \{A_0, A_1, A_2, \dots\}$  and  $A_0 \ll A_1 \ll A_2 \ll \dots$ . We define  $b(x) = \inf\{i \mid x \in X_i\}$  and  $e(x) = \sup\{i \mid x \in A_i\}$ . By definition we have  $b(x) \leq e(x)$ . Let  $x, y \in P$  s.t.  $x <_P y$ , then for all  $i < j$  if  $x \in A_j$ , we have  $A_i \ll A_j$  so  $y \notin A_i$ , and  $y \notin A_j$  (otherwise  $X_j$  is not an antichain), thus  $e(x) < b(y)$ . Let  $x, y \in P$  s.t.  $x \not<_P y$ . Either  $y <_P x$ , then by the previous point  $e(y) < b(x)$  and so  $b(y) < e(x)$ , or we have  $y \not<_P x$  thus  $\{x, y\}$  is include in a maximal antichain of  $P$ , then there is  $i$  such that  $x, y \in A_i$  and so  $b(y) \leq i \leq e(x)$ . Thus  $x <_P y \Leftrightarrow e(x) < b(y)$ .
- So we have (1)  $\Leftrightarrow$  (2)  $\Leftrightarrow$  (3). ◀

Before proving Th. 7 we need lemma that characterize the nature (starter versus terminator) of the first element of the decomposition (Lem. A.3), and one that characterise it solely from the  $P$  considered (Lem. A.4). We define  $P^m$  the  $<_P$ -minimal elements of  $P$  (which can be seen as a conclist with the induced event order of  $P$ ) and  $P^s := \{p \in P \mid \forall q \in P - P^m, p < q\} \subseteq P^m$ .

► **Lemma A.2.** *Let  $P = P_0 * P_1 * \dots$  be a sparse decomposition, then  $P_0 = P^m$  (seen as conclists).*

**Proof of Lemma A.2.** By definition, we already have that  $P_0 \subseteq P^m$ . Let's show that  $P - P_0 \subseteq P - P^m$  (which is equivalent to  $P^m \subseteq P_0$ ). Let  $x \in P - P_0$ , there is  $i \geq 1$  s.t.  $x \in P_i - S_{P_i}$ .

- If  $P_0$  is a (proper) terminator: it exists  $y \in P_0 - T_{P_0}$  and thus  $y < x$
- If  $P_0$  is a starter: then  $P_1$  is a (proper) terminator so it exists  $y \in P_1 - T_{P_1}$ , and  $i \geq 2$  (because  $P_1 = S_{P_1}$ ), thus  $y < x$

In both cases there is an element before  $x$  so  $x \notin P^m$ , so  $P - P_0 \subseteq P - P^m$ . ◀

► **Lemma A.3.** *Let  $P = P_0 * P_1 * \dots$  be a sparse decomposition, then  $P_0$  is a starter iff  $S_P \subsetneq P^m$  (equivalently,  $P_0$  is a terminator iff  $S_P = P^m$ ).*

**Proof of Lemma A.3.** Using Lem. A.2,  $P_0 = P^m$ , and by definition  $S_{P_0} = S_P$ . Then we have that  $P_0$  is a starter iff  $S_P = S_{P_0} \subsetneq P_0 = P^m$ . ◀

► **Lemma A.4.** Let  $P = P_0 * P_1 * \dots$  be a sparse decomposition:

- If  $P_0$  is a starter,  $P_0 = {}_{P^m-S_P}\uparrow P^m$
- If  $P_0$  is a terminator,  $P_0 = P^m \downarrow_{P^s}$

**Proof of Lemma A.4.** Using Lem. A.2,  $P_0 = P^m$ :

- If  $P_0$  is a starter, by Lem. A.3  $S_P = S_{P_0} \subsetneq P_0 = P^m$  and so  $P_0 = {}_{P_0-S_{P_0}}\uparrow P_0 = {}_{P^m-S_P}\uparrow P^m$ .
- If  $P_0$  is a terminator, we want to show that  $P^s = P^m - T_{P_0}$ , thus  $P_0 = P^m \downarrow_{P^s}$ . First let  $p \in T_{P_0}$ , then  $p \in S_{P_1} \subseteq P_1$ .  $P_1$  is a (proper) starter so there is  $q \in P_1 - S_{P_1}$  so  $q \notin P_0 = P^m$  and  $p \not\prec q$  so  $p \notin P^s$ , thus  $P^s \subseteq P^m - T_{P_0}$ . Then let  $p \in P^m - T_{P_0} = P_0 - T_{P_0}$  and  $q \in P - P^m = P - P_0$ , thus there is  $i \geq 1$  s.t.  $q \in P_i - S_{P_i}$ , so  $p < q$ . Thus  $p \in P^s$  and  $P^m - T_{P_0} \subseteq P^s$ .

Which concludes the proof. ◀

We can now prove the fundamental decomposition of  $\omega$ -ipomsets:

**Proof of Theorem 7.** Let  $P \in \text{iiPoms}^\omega$ , by Lem. A.1, as  $\ll$  is linear by Prop. 5, we have that  $X_0 \ll X_1 \ll \dots$ . For all  $i \in \mathbb{N}$ , we can define  $E_i = X_i - X_{i+1}$  and  $B_i = X_{i+1} - X_i$ . As  $X_i$  are maximal antichains, the sets are not empty and  $\forall (x, y) \in E_i \times B_i$ ,  $x <_P y$ . We then define  $U_{2i} = X_i \downarrow_{E_i}$  and  $U_{2i+1} = {}_{B_i}\uparrow X_{i+1}$  which are proper (the event order on  $U_{2i}$  and  $U_{2i+1}$  is the one induced by  $\rightarrow_P$ ). If  $S_P \subsetneq X_0$ , we shift all indices by 1 and redefine  $U_0 = {}_{X_0-S_P}\uparrow X_0$ . Thus  $U_0 * U_1 * U_2 * \dots$  is a sparse decomposition, and it is routine to check that  $P = U_0 * U_1 * U_2 \dots$ , concluding of the existence of a sparse decomposition.

Let  $P = P_0 * P_1 * \dots = Q_0 * Q_1 * \dots$  be two different sparse decompositions. Without loss of generality, we can delete the common prefix and consider that  $P_0 \neq Q_0$ . If  $S_P \subsetneq P^m$ , by Lem. A.3  $P_0$  and  $Q_0$  are (proper) starters, and by Lem. A.4,  $P_0 = {}_{P^m-S_P}\uparrow P^m = Q_0$ . If  $S_P = P^m$ , by Lem. A.3  $P_0$  and  $Q_0$  are (proper) terminators, and by Lem. A.4,  $P_0 = P^m \downarrow_{P^s} = Q_0$ , so contradiction. ◀

**Proof of Proposition 12.** We show the equivalence between (1)  $P$  is an interval  $\omega$ -ipomset, (4) all prefixes of  $P$  are interval ipomsets and  $\text{Pref}(P)$  is infinite, and (5) an infinite number of prefixes of  $P$  are interval ipomsets.

- (1)  $\Rightarrow$  (4) : Suppose for all  $w, x, y, z \in P$  where  $w < y$  and  $x < z$ , then  $w < z$  or  $x < y$ . Then for all  $A \in \text{Pref}(P)$ , we have the same property for all  $w, x, y, z \in A \subseteq P$ . Moreover,  $\text{Pref}(P)$  is infinite as a direct consequence of the existence part of Th. 7.
- (4)  $\Rightarrow$  (5) : Trivial.
- (5)  $\Rightarrow$  (1) : Let  $(A_i)_{i \in I}$  be the infinite set of interval ipomsets in  $\text{Pref}(P)$ . Let  $w, x, y, z \in P$  where  $w < y$ ,  $x < z$ . By Lem. 10, there is a prefix  $A_i$  with  $\{w, x, y, z\} \subseteq A_i$ , which is an interval ipomset, then  $w < y$  or  $x < z$ .

Which concludes the proof. ◀

## B Complementary proofs of Section 5

**Proof of Lemma 25.** Let  $A = (Q, E, I, F, \lambda)$  be an ST-automaton. As  $\lambda$  is coherent with  $E$ , all interfaces coincide in a word produced by  $A$ . Then it remains to show that the languages does not contain not allowed words.

If  $A$  accepts finite words, then even for a path of one states  $q$ , the label contains at least the letter  $\text{id}_{\lambda(q)}$ . Thus  $\varepsilon \notin L(A)$ , and  $L(A) \subseteq \text{Coh}^+(\bar{\Xi})$ .



If  $A$  accepts infinite words, then for every infinite path, one letter out of two is in  $\bar{\Xi} \setminus \text{Id}$ , so there is an infinite number of non-identity letters, thus  $L(A) \subseteq \text{Coh}^\omega(\bar{\Xi})$ .  $\blacktriangleleft$

**Proof of Lemma 27.** Let  $\alpha = (q_0, \varphi_0, q_1, \dots)$  be an  $\omega$ -track in  $X$  and  $ST(\alpha) = (q_0, e_0, q_1, \dots)$  with  $e_i = (q_i, P_i, q_{i+1})$ . By definition:

- If  $(q_i, \varphi_i, q_{i+1}) = (\delta_{A_i}^0(q_{i+1}), \nearrow^{A_i}, q_{i+1})$ , then  $P_i = \text{ev}(q_i, \varphi_i, q_{i+1})$
- If  $(q_i, \varphi_i, q_{i+1}) = (q_i, \searrow_{A_i}, \delta_{A_i}^1(q_i))$ , then  $P_i = \text{ev}(q_i, \varphi_i, q_{i+1})$

We recall that  $\lambda$  is coherent, so for  $(q_i, e_i, q_{i+1}) \in E$  we have  $\lambda(q_i) = T_{P_i}$ ,  $\lambda(q_{i+1}) = S_{P_{i+1}}$ , hence:

$$\begin{aligned} \text{ev}(\alpha) &= \text{ev}(q_0, \varphi_0, q_1) * \text{ev}(q_1, \varphi_1, q_2) * \dots = P_0 * P_1 * \dots = \text{id}_{\lambda(q_0)} * P_0 * \text{id}_{\lambda(q_1)} * P_1 * \dots \\ &= \Psi(\text{id}_{\lambda(q_0)} \cdot P_0 \cdot \text{id}_{\lambda(q_1)} \cdot P_1 \cdot \dots) = \Psi(l(ST(\alpha))) \end{aligned} \quad \blacktriangleleft$$

**Proof of Lemma 29.** For an expression  $e_A$  of finite words, we replace any occurrence of the Kleene star with the formula  $L^* = \varepsilon + L^+$ , and “developp” the expression. For ST-automaton  $A$ ,  $\varepsilon \notin L(A)$ , the resulting expression after developing is such that there is no  $\varepsilon$  anymore.

For an  $\omega$ -rational expression  $e_{A'}$ , we know that  $e_{A'}$  can be a finite union of the form  $\bigcup X.Y^\omega$  with  $X$  and  $Y$  rational sets of finite words (see Th. 3.2 of [39]). It suffices to show that we can remove the occurrences of  $*$  and  $\varepsilon$  for an expression  $X.Y^\omega$ . We replace any occurrences of the Kleene star by  $L^* = \varepsilon + L^+$ . Then we develop the expression such that:

- If  $\varepsilon \in X$ ,  $X = \varepsilon + X'$  (with  $\varepsilon \notin X'$ ) and  $X.Y^\omega = Y^\omega + X'.Y^\omega = Y.Y^\omega + X'.Y^\omega$
- If  $\varepsilon \in Y$ ,  $Y = \varepsilon + Y'$  (with  $\varepsilon \notin Y'$ ) and because of the definition of  $\omega$ , we have  $X.Y^\omega = X.Y'^\omega$

Thus we can transform  $e_A$  into a positive expression  $\bar{e}_A$  (resp.  $e_{A'}$  into  $\bar{e}_{A'}$ ). For a formula over infinite words, it is moreover still of the form  $\bigcup X.Y^\omega$ .  $\blacktriangleleft$

**Proof of Lemma 32.** Let  $A$  be an ST-automaton of finite words. The ST-automaton is finite, so by the classical Kleene theorem,  $L(A)$  can be represented by a rational expression  $e_A(U_1, \dots, U_n)$ , written  $e_A$  for simplification. We have  $\mathcal{L}(e_A) = L(A)$ . By Lem 29, we can suppose  $e_A$  to be positive, and by Lem 31,  $\Psi(L(A)) = \Psi(\mathcal{L}(e_A)) = \Psi'(e_A)$ . By Lem 30,  $\Psi'(e_A) \in \text{Rat}$ , so  $\Psi(L(A)) \in \text{Rat}$ .

The same can be done for an ST-automaton of infinite words  $A'$ . By the  $\omega$ -Kleene theorem (for example shown in [39]),  $L(A')$  can be represented by a  $\omega$ -rational positive expression  $e_{A'}$ , with  $\mathcal{L}(e_{A'}) = L(A')$ , and we have  $\Psi(L(A')) = \Psi(\mathcal{L}(e_{A'})) = \Psi'(e_{A'})$ , and by Lem 30,  $\Psi(L(A')) \in \text{Rat}^\omega$ .  $\blacktriangleleft$

The following results are needed for the induction part of the proof of Prop. 36:

► **Lemma B.5.** For  $L, M \subseteq \text{iiPoms}$ ,  $(L * M)\downarrow = L\downarrow *_\downarrow M\downarrow$ .

**Proof of Lemma B.5.** Let  $P \in (L * M)\downarrow$ , there is  $A \in L$  and  $B \in M$  such that  $P \sqsubseteq A * B$ . As  $L \subseteq L\downarrow$  and  $M \subseteq M\downarrow$ ,  $A * B \in L\downarrow * M\downarrow$  so  $P \in (L\downarrow * M\downarrow)\downarrow = L\downarrow *_\downarrow M\downarrow$  and  $(L * M)\downarrow \subseteq L\downarrow *_\downarrow M\downarrow$ .

Let  $Q \in L\downarrow *_\downarrow M\downarrow$ , there is  $A \in L\downarrow$  and  $B \in M\downarrow$  such that  $Q \sqsubseteq A * B$ . There is  $A' \in L$  such that  $A \sqsubseteq A'$  and  $B' \in M$  such that  $B \sqsubseteq B'$ . By Lem. 48 of [18],  $A * B \sqsubseteq A' * B'$  and by transitivity,  $Q \sqsubseteq A' * B'$ . But we have  $A' * B' \in L * M$ , so  $Q \in (L * M)\downarrow$  and  $L\downarrow *_\downarrow M\downarrow \subseteq (L * M)\downarrow$ .  $\blacktriangleleft$

► **Corollary B.6.** For  $L, M \subseteq \text{iiPoms}$ ,  $(L^+) \downarrow = (L\downarrow)^{+\downarrow}$ .



**Proof of Proposition 36.** We start by showing that  $\mathcal{Rat}\downarrow \subseteq D\text{-}\mathcal{Rat}$  by induction:

- $\emptyset\downarrow = \emptyset \in D\text{-}\mathcal{Rat}$
- For  $U \in \Xi$ ,  $\{U\}\downarrow = \{U\}$  (as  $<_U$  is empty) and  $\{U\} \in D\text{-}\mathcal{Rat}$  (by a finite use of  $\parallel\downarrow$  and  $\{a\bullet\}$  and  $\{\bullet a\bullet\}$ , or  $\{\bullet a\}$  and  $\{\bullet a\bullet\}$ , for  $a \in \Sigma$ )
- For  $L, M \in \mathcal{Rat}$ , if  $L\downarrow, M\downarrow \in D\text{-}\mathcal{Rat}$ , then  $(L + M)\downarrow = L\downarrow + M\downarrow \in D\text{-}\mathcal{Rat}$
- For  $L, M \in \mathcal{Rat}$ , if  $L\downarrow, M\downarrow \in D\text{-}\mathcal{Rat}$ , then  $(L * M)\downarrow = L\downarrow *_{\downarrow} M\downarrow \in D\text{-}\mathcal{Rat}$  (by Lem. B.5)
- For  $L \in \mathcal{Rat}$ , if  $L\downarrow \in D\text{-}\mathcal{Rat}$ , then  $(L^+)\downarrow = (L\downarrow)^{+\downarrow} \in D\text{-}\mathcal{Rat}$  (by Cor. B.6)

Then by induction  $\mathcal{Rat}\downarrow \subseteq D\text{-}\mathcal{Rat}$ .

Next we show that  $D\text{-}\mathcal{Rat} \subseteq \mathcal{Rat}$ , which is just a more detailed version of the Remark 7.3 of [19]. Let  $L \in D\text{-}\mathcal{Rat}$ , by Th. 17 there is a HDA  $X$  such that  $L = L(X)$ . Let  $ST(X)$  be its ST-automata, by Prop. 26  $L(X) = \Psi(L(ST(X)))$ , and by Lem. 32,  $\Psi(L(ST(X))) \in \mathcal{Rat}$ . Thus  $D\text{-}\mathcal{Rat} \subseteq \mathcal{Rat}$ .

But  $L \in D\text{-}\mathcal{Rat}$ ,  $L$  is down-closed (by Prop. 16 and Th. 17) so as  $L \in \mathcal{Rat}$ ,  $L = L\downarrow \in \mathcal{Rat}\downarrow$ . Then  $D\text{-}\mathcal{Rat} \subseteq \mathcal{Rat}\downarrow$ . ◀

**Proof of Theorem 39.** We use the key property that language of a classical Büchi automaton  $A = (Q, E, I, F)$  is of the form  $\bigcup_{q_i \in I, q_f \in F} W(q_i, q_f).W(q_f, q_f)^\omega$ , where  $W(p, q)$  is the language of the Büchi automaton  $(Q, E, \{p\}, \{q\})$  (see [24] for more details).

Let  $(X, \perp_X, \top_X)$  be a Büchi  $\omega$ -HDA, and  $ST(X) = (Q, E, I, F, \lambda)$  be its ST-automaton. Thus  $L(ST(X)) = \mathcal{L}(\bigcup_{q_i \in I, q_f \in F} W(q_i, q_f).W(q_f, q_f)^\omega)$ . To simplify, we write  $W_1 = W(q_i, q_f)$  and  $W_2 = W(q_f, q_f)$ . As,  $W_2 = (W_2)^+ + \varepsilon$  so we have  $L(ST(X)) = \mathcal{L}(\bigcup W_1.((W_2)^+)^\omega)$ .

But  $\mathcal{L}(W_1)$  and  $\mathcal{L}(W_2)$  are languages of ST-automata from HDAs (over finite tracks), so by Prop. 36 it exists  $X, Y \subseteq \text{iiPoms}$  such that  $\Psi(\mathcal{L}(W_1)) = X\downarrow$  and  $\Psi(\mathcal{L}(W_2)) = Y\downarrow$ . Then:

$$\begin{aligned}
 L(X) &= \Psi(L(ST(X))) = \Psi(\mathcal{L}(\bigcup W_1.((W_2)^+)^\omega)) && \text{(Prop. 28)} \\
 &= \Psi'(\bigcup W_1.((W_2)^+)^\omega) = \bigcup \Psi'(W_1) * (\Psi'(W_2)^+)^\omega && \text{(Lem. 31)} \\
 &= \bigcup \Psi(\mathcal{L}(W_1)) * (\Psi(\mathcal{L}(W_2)))^+{}^\omega && \text{(Lem. 31)} \\
 &= \bigcup X\downarrow * ((Y\downarrow)^+)^\omega
 \end{aligned}$$

Which concludes the proof. ◀