A Kleene Theorem for Higher-Dimensional Automata

Uli Fahrenberg
EPITA Research and Development Laboratory (LRDE), Paris, France

Christian Johansen
NTNU Gjøvik, Norway

Georg Struth
University of Sheffield, UK
Collegium de Lyon, France

Krzysztof Ziemiański
University of Warsaw, Poland

Abstract

We prove a Kleene theorem for higher-dimensional automata (HDAs). It states that the languages they recognise are precisely the rational subsumption-closed sets of interval pomsets. The rational operations include a gluing composition, for which we equip pomsets with interfaces. For our proof, we introduce HDAs with interfaces as presheaves over labelled precube categories and use tools inspired by algebraic topology, such as cylinders and (co)fibrations. HDAs are a general model of non-interleaving concurrency, which subsumes many other models in this field. Interval orders are used as models for concurrent or distributed systems where events extend in time. Our tools and techniques may therefore yield templates for Kleene theorems in various models and applications.

2012 ACM Subject Classification Theory of computation → Automata extensions

Keywords and phrases higher-dimensional automata, interval posets, Kleene theorem, concurrency theory, labelled precube categories

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.29

Related Version *Full Version*: `https://arxiv.org/abs/2202.03791`

1 Introduction

Higher-dimensional automata (HDAs) were introduced by Pratt and van Glabbeek as a general geometric model for non-interleaving concurrency [21,23]. HDAs support autoconcurrency and events with duration or structure, whereas events in interleaving models must be instantaneous. They subsume, for example, event structures and safe Petri nets [24]. Asynchronous transition systems and standard automata are two- and one-dimensional HDAs, respectively [12]. We have recently used van Glabbeek's (execution) paths [24] to relate HDAs with certain languages of interval posets [6]. Yet a precise description of these languages in terms of a Kleene theorem has so far been missing. Our main contribution is the formalisation and proof of such a theorem.

HDAs consist of cells and lists of events that are active in them. Zero-dimensional cells represent states in which no event is active; 1-dimensional cells represent transitions in which exactly one event is active – as in standard automata. Higher n-dimensional cells model concurrent behaviours with n active events. As an example, Fig. 1 shows an HDA with cells of dimension ≤ 2. The cells x and y, for instance, have active events $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} a \\ c \end{smallmatrix}\right]$, respectively. Cells at any dimension may serve as start and accept cells. In Fig. 1, these are marked with incoming and outgoing arrows. Lower dimensional cells or faces are attached to higher dimensional ones using lower and upper face maps. These indicate further when individual

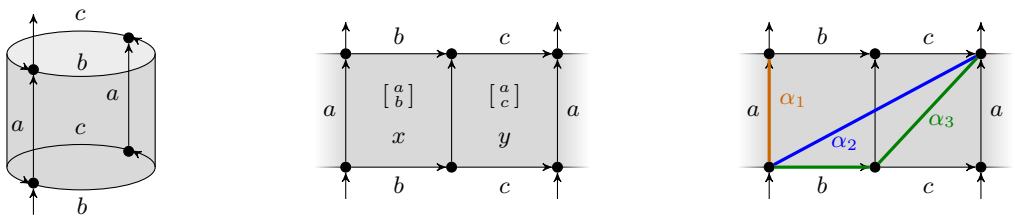


Figure 1 HDA with two 2-dimensional cells x and y connected along transitions and modelling parallel execution of a and $(bc)^*$. Middle: unfolded view; right: three accepting paths.

events start or terminate. In Fig. 1, the lower face $\delta^0_a(x)$ of x is the lower b-transition in which a is not yet active; its upper face $\delta^1_a(x)$ is the upper b-transition in which a is no longer active. Further, $\delta^1_b(x) = \delta^0_c(y)$ and $\delta^1_c(y) = \delta^0_b(x)$.

Executions of HDAs are (higher-dimensional) paths [24]: sequences of cells connected with operations of starting and terminating events. Every path α is characterised by ordering the events $\mathrm{ev}(\alpha)$ that occur in it with respect to precedence. This always yields interval orders. In addition, $\mathrm{ev}(\alpha)$ is equipped with source and target interfaces, which model events active in the initial and final cell of α, respectively, and a secondary event order, which captures the list structure of events in cells. We call (isomorphism classes of) such labelled posets with interfaces and an event order *ipomsets*. The language of an HDA is then related to the set of (interval) ipomsets associated with all its accepting paths – from start to accept cells [6]. Languages of HDAs must in particular be down-closed with respect to less concurrent executions, modelled by a subsumption preorder. This motivates the definition of (interval ipomset) languages as subsumption-closed sets of interval ipomsets.

Kleene theorems usually require a notion of *rational* language. Here it is based on the union $\cup$, gluing (serial) composition $*$, parallel composition $\|$, and (serial) Kleene plus $^+$ of languages. These definitions are not entirely straightforward, as down-closure and the interval property must be preserved. In particular, $*$ is more complicated than, for instance, the standard series composition of pomsets due to interfaces. Without interfaces, it would reduce to the latter. We consider finite HDAs only and thus can neither include the parallel Kleene star nor the full serial Kleene star as a rational operation. The latter contains the identity language, which requires an HDA of infinite dimension.

Our Kleene theorem shows that the rational languages are precisely the regular languages (recognised by finite HDAs). To show that regular languages are rational, we translate the cells of an HDA into a standard automaton and reuse one direction of the standard Kleene theorem. Proving that rational languages are regular is harder. Regularity of $\cup$ is straightforward, and for $\|$, the corresponding operation on HDAs is simply a tensor product. But $*$ and $^+$ require an intricate gluing operation on HDAs along higher-dimensional cells.

Beyond the Kleene theorem, three further contributions seem of independent interest. We model HDAs as presheaves on novel precube categories that feature events and labels in the base category. These are equivalent to standard HDAs [24], but constructions become simpler and the relation between iposets and precubical sets clearer.

We also introduce iHDAs – HDAs with interfaces – which may assign events to source or target interfaces. Target events cannot be terminated: they either remain active at the end of an execution or do not appear at all. By opposition, source events cannot be "unstarted": they are either active at the beginning of an execution or do not appear at all. Additionally, all events of start cells are source events and all events of accept cells, target events. Every

HDA can be converted into an equivalent iHDA (recognising the same language) and vice versa, using operations of resolution and closure. Both models play a technical role in our proofs, and we frequently switch between them.

Another tool used in the Kleene theorem is motivated by algebraic topology. We introduce cylinder objects and show that every map between (i)HDAs can be decomposed into an (initial or final) inclusion followed by a (future or past) path-lifting map. This allows us to "pull apart" start and accept cells of iHDAs when dealing with serial compositions and loops.

In this paper we introduce the concepts needed for formulating and proving the Kleene theorem, and we outline its proof. Most technical details can be found in the long version [7].

2 Higher-Dimensional Automata

HDAs and iHDAs are particular (pre)cubical sets. Like simplicial sets, they are typically modelled as presheaves on certain base categories. Here we introduce new labelled precube categories and variants with interfaces, which tame the combinatorics of concurrent events in well-structured ways. Their objects are lo-sets, which model concurrent events that are active in some cell of an HDA, or ilo-sets, which equip lo-sets with interfaces. Their morphisms are coface maps (opposites of face maps), which insert events into lo-sets and preserve interfaces if present. Some constructions require isomorphism classes of labelled precube categories (with interfaces), and we define these as well. Finally, we briefly introduce resolution and closure functors that translate between HDAs and iHDAs, in preparation for the summary of our Kleene theorem in Section 5. We contextualise our approach in App. A. Throughout the paper, we fix an alphabet Σ, finite or infinite.

Lo-sets and ilo-sets. A *lo-set* $(U, \dashrightarrow, \lambda)$ is a finite set U totally ordered by the strict order $\dashrightarrow$ and labelled by $\lambda : U \to \Sigma$. A *lo-set with interfaces* (*ilo-set*) is a triple (S, U, T) of a lo-set U, a *source interface* $S \subseteq U$ and a *target interface* $T \subseteq U$. Lo-sets are regarded as ilo-sets with empty interfaces. We write ${}_S U_T$ or just U for ilo-sets.

The labels of ilo-sets indicate the actions associated with events. The *event order* $\dashrightarrow$ captures their list structure, which is convenient for regarding ilo-sets as ipomsets in Section 3.

A *lo-map* is an order and label preserving function between lo-sets. Lo-maps are strict order embeddings and thus injective, as $\dashrightarrow$ is total. A *lo-isomorphism* is therefore a surjective lo-map. An *ilo-map* $f : U \to V$ must also satisfy $S_U = f^{-1}(S_V)$ and $T_U = f^{-1}(T_V)$ (and $f(S_U) = S_V$ and $f(T_U) = f(T_V)$ if it is an ilo-isomorphism). We write $U \simeq V$ if (i)lo-sets U and V are isomorphic – there is at most one isomorphism between (i)lo-sets. Isomorphism classes of lo-sets are words over Σ. Those of ilo-sets are words over an extended alphabet $\Sigma_\bullet = \{a, \bullet a, a \bullet, \bullet a \bullet \mid a \in \Sigma\}$, where $\bullet a$, for example, indicates membership in a source interface. As in Fig. 1, we represent such words as column vectors.

Each (i)lo-map $f : U \hookrightarrow V$ defines and is defined by a unique $A = V \setminus f(U) \subseteq V$. We write ∂_A for this map; it is an isomorphism iff $A = \emptyset$. The composite of (i)lo-maps $\partial_A : U \hookrightarrow V$, $\partial_B : V \hookrightarrow W$ is $\partial_{\partial_B(A) \cup B} : U \hookrightarrow W$. This data defines categories of (i)lo-sets. Each $\partial_A : U \to V$ constructs V by inserting the events of A into U, while the face maps in the introduction delete events and thus map in the opposite direction, see Fig. 2. Linking (i)lo-maps with face maps and initial and final cells of HDAs requires refinements.

Labelled precube categories. Labelled precube categories account for the active, unstarted and terminated events in HDAs and equip them with interfaces. The arrows of labelled precube categories are coface maps. These map in the opposite direction of face maps.

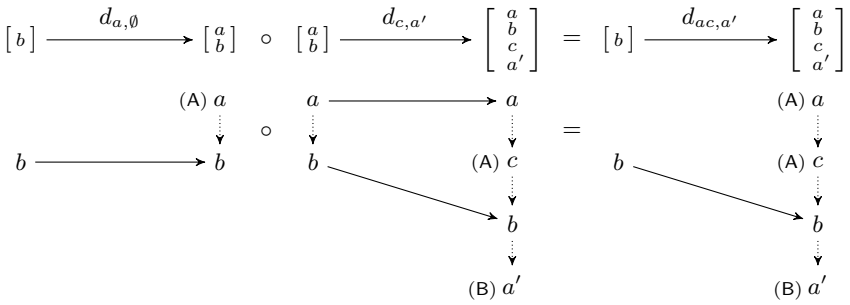


Figure 2 Composition of morphisms in $\square$. Letters (A) and (B) indicate events that become unstarted (A) or terminated (B), as defined in the triple $(\partial_{A\cup B}, A, B)$.

The *full labelled precube category with interfaces*, $\mathbb{I}\square$, has ilo-sets as objects and *coface maps* $d_{A,B} : U \to V$ as arrows. Each $d_{A,B}$ is a triple $(\partial_{A\cup B}, A, B)$ with $A, B \subseteq V$ disjoint, $A \cap S_V = \emptyset = B \cap T_V$, and $\partial_{A\cup B} : U \hookrightarrow V$ an ilo-map. The composite of $d_{A,B} : U \to V$ and $d_{C,D} : V \to W$ is defined as $d_{C,D} \circ d_{A,B} = (\partial_{C\cup D} \circ \partial_{A\cup B}, \partial_{C\cup D}(A) \cup C, \partial_{C\cup D}(B) \cup D)$.

As an instance, the *full labelled precube category* $\square$ with lo-sets as objects and coface maps $d_{A,B}$ based on lo-maps $\partial_{A\cup B}$ is trivially isomorphic to the full subcategory of $\mathbb{I}\square$ with empty interfaces S_U, T_U, for each ilo-set U.

We only need coface map compositions with pairwise disjoint $A, B, C, D \subseteq W$ and $V = W \setminus (C \cup D)$. In this case,

$$d_{C,D} \circ d_{A,B} = d_{A\cup C, B\cup D}. \tag{1}$$

See Fig 2 for an example. We write d^0_A for $d_{A,\emptyset}$ and d^1_B for $d_{\emptyset,B}$. Intuitively, $d^0_A : U \to V$ inserts events in A into U that are unstarted in U, whereas $d^1_B : U \to V$ inserts events B into U that are terminated in U. See again Fig. 2 for an example. With d^1_B, events in T_V cannot terminate in U as $T_V \cap B = \emptyset$; with d^0_A, those in S_V cannot be unstarted in U as $S_V \cap A = \emptyset$. Both interfaces are preserved in pre-images of $\partial_{A\cup B}$ and thus by coface map compositions.

It is standard to work with equivalence classes of events. The *labelled precube category with interfaces* $\mathbb{I}\square$ is the quotient of $\mathbb{I}\square$ with respect to $\simeq$. Its objects are words over $\Sigma_\bullet$. Its coface maps are equivalence classes of coface maps in $\mathbb{I}\square$, where $d_{A,B} : U \to V$, $d_{A',B'} : U' \to V'$ are equivalent in $\mathbb{I}\square$ if $U \simeq U'$ and $V \simeq V'$ (hence also $A \simeq A'$ and $B \simeq B'$). They insert letters into words, while remembering whether they correspond to unstarted or terminated actions.

The *labelled precube category* $\square$ is obtained from $\square$ in a similar way. The categories $\mathbb{I}\square$ and $\square$ are skeletal, and because isomorphisms are unique, the quotient functors $\mathbb{I}\square \to \mathbb{I}\square$ and $\square \to \square$ are equivalences of categories. We thus switch freely between full categories and their skeletons and identify morphisms $[U] \to [V]$ with representatives $d_{A,B} : U \to V$.

We often use the involutive *reversal* functor on these categories. It maps ${}_SU_T$ to ${}_TU_S$ and $d_{A,B}$ to $d_{B,A}$, thus swapping unstarted and terminated events.

Higher-dimensional automata. A *precubical set with interfaces* (*ipc-set*) is a presheaf on $\mathbb{I}\square$, that is, a functor $X : \mathbb{I}\square^{\mathrm{op}} \to \mathbf{Set}$. We write $X[U]$ for the value of X on object U of $\mathbb{I}\square$. We write $\delta_{A,B} = X[d_{A,B}] : X[U] \to X[U \setminus (A \cup B)]$ for the *face map* associated to coface map $d_{A,B} : U \setminus (A \cup B) \to U$. Elements of $X[U]$ are *cells* of X – we often view X as a set of cells.

We write $\mathrm{iev}(x) = {}_SU_T$ and $\mathrm{ev}(x) = U$ if $x \in X[{}_SU_T]$, as well as $\delta^0_A = X[d^0_A]$ and $\delta^1_B = X[d^1_B]$. Such face maps attach lower and upper faces to the cells in $X[U]$.

A *precubical set* is defined analogously as a presheaf X on $\square$. We may view it as an ipc-set X such that $X[{}_S U_T] = \emptyset$ whenever $S \neq \emptyset$ or $T \neq \emptyset$. A precubical set or ipc-set X is *finite* if it has finitely many cells. The *dimension* of a cell $x \in X[U]$ is $|U|$. The *dimension* of X is the maximal dimension among its cells.

A *higher-dimensional automaton with interfaces* (*iHDA*) is a finite ipc-set X with subsets $X_\perp$ of *start cells* and $X^\top$ of *accept cells*. These are required to satisfy $S = U$ for all $x \in X_\perp$ with $\mathrm{iev}(x) = {}_S U_T$, and $T = U$ for all $x \in X^\top$ with $\mathrm{iev}(x) = {}_S U_T$. $X_\perp$ and $X^\top$ are not necessarily precubical subsets.

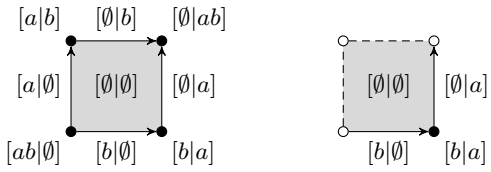
Analogously, a *higher-dimensional automaton* (*HDA*) is a finite precubical set X equipped with subsets $X_\perp$ and $X^\top$ of start and accept cells. HDAs are not simply special cases of iHDAs due to the above requirements on interfaces.

An *ipc-map* is a natural transformation $f : X \to Y$ of ipc-sets X, Y, an *iHDA-map* must preserve start and accept cells as well: $f(X_\perp) \subseteq Y_\perp$ and $f(X^\top) \subseteq Y^\top$. Precubical maps and HDA-maps are defined analogously. We write $\square\mathbf{Set}$, $\mathbb{I}\square\mathbf{Set}$, **HDA** and **iHDA** for the resulting categories (of precubical sets, ipc-sets, HDAs and iHDAs, respectively).

The reversal on $\square$ translates to ipc-sets, precubical sets, iHDAs and HDAs. It maps $\delta_{A,B}$ to $\delta_{B,A}$ and exchanges start and accept cells if present. The relationship between HDAs and iHDAs is studied in [7].

Standard cubes. The *standard U-cube* $\square^U$ of lo-set U is the precubical set represented by U (the Yoneda embedding): $\square^U = \square(-, U)$. For each lo-set V, $\square^U[V]$ is the set of all $d_{A,B} : V \to U$ with $A, B \subseteq U$. We write $[A|B]$ for such a cell. For $d_{A,B} : U \to V$ we denote by $\square^{d_{A,B}} : \square^U \to \square^V$ the induced map given by $\square^{d_{A,B}}([C|D]) = [A \cup C | B \cup D]$. The definition of $\mathbb{I}\square^U$ for an ilo-set U is analogous.

▶ **Example 1.** For $U = \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$, $\square^U$ has the cells occurring as pairs $[-|-]$ in the left-hand cube below. The first coordinate of the pair lists the unstarted events associated to a cell, the second one the terminated ones.



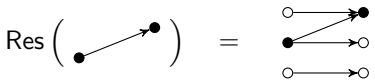
The maps $\square^{d_{A,B}}$ attach faces to cells. The lower face map $\square^{d^0_a}$, for instance, attaches $[a|\emptyset]$ as the left 1-cell to the 2-cell $[\emptyset|\emptyset]$ and $[ab|\emptyset]$ as the left 0-cell to the 1-cell $[b|\emptyset]$. Notation $[a|\emptyset]$ indicates that a has not yet started and no element has terminated in the associated face, while b is active. An analogous construction of the standard cube $\mathbb{I}\square^V$, for $V = \left[\begin{smallmatrix} \bullet a \\ b \bullet \end{smallmatrix}\right]$, is shown in the right-hand diagram above. Here $\bullet a$ and $b \bullet$ prevent a from starting and b from terminating on faces. All lower faces in the left-hand cube containing a on the left of the $|$ and all upper faces containing b on the right of the $|$ must thus be removed. (We have omitted some set braces and likewise.)

The notation $[A|B]$ for the cells of $\square^U$ is useful in later proofs. For any (i)HDA X and $x \in X$, there exists a unique precubical map, the Yoneda embedding $\iota_x : \square^U \to X$ ($\mathbb{I}\square^U \to X$), such that $\iota_x([\emptyset|\emptyset]) = x$.

Standard automata. One-dimensional HDAs X are slight generalisations of standard finite automata. Cells in $X[\emptyset]$ are states of the automaton, those in $X[a]$ are a-labelled transitions. The face maps $\delta^0_a, \delta^1_a : X[a] \to X[\emptyset]$ attach sources and targets to transitions. Yet transitions may serve as start and accept cells in X.

Resolution and closure. The forgetful functor $\mathsf{F} : \mathbb{I}\square \to \square$, ${}_SU_T \mapsto U$ induces two functors $\mathsf{Res} : \square\,\mathbf{Set} \to \mathbb{I}\square\,\mathbf{Set}$ and $\mathsf{Cl} : \mathbb{I}\square\,\mathbf{Set} \to \square\,\mathbf{Set}$. *Resolution* Res is given by $\mathsf{Res}(X) = X \circ \mathsf{F}$: the free functor induced by F, and *closure* Cl is left adjoint to Res. We extend them to functors between **HDA** and **iHDA** in a natural way.

Intuitively, closure fills in missing cells of ipc-sets: the standard cube $\square^U$ of Example 1, for instance, is the closure of $\mathbb{I}\square^V$. The cells of $\mathsf{Res}(X)$ are triples (x, S, T), where x is a cell of X and the subsets $S, T \subseteq \mathrm{ev}(x)$ are all possible assignments of interfaces. Every cell $x \in X[U]$ thus produces $4^{|U|}$ cells in $\mathsf{Res}(X)$, for example,



We give a detailed description of Res and Cl in [7].

3 Ipomsets

Pomsets are a standard model of non-interleaving concurrency. Ipomsets have been introduced to model the behaviours of a restricted class of HDAs [6]. Here we recall the basic definitions and adapt them to general HDAs. A notion of *rational ipomset language* is related with HDAs in the Kleene theorem of Section 5.

Ipomsets. A *labelled iposet* $(P, <, \dashrightarrow, S, T, \lambda)$ consists of a finite set P with two strict (partial) orders: the *precedence order* $<$ and the *event order* $\dashrightarrow$ such that each pair in P is comparable by either $\leq$ or $\dashrightarrow$. $\lambda : P \to \Sigma$ is a labelling function, and $S, T \subseteq P$ are the source and target *interfaces* of P. Elements of S must be $<$-minimal and those of T $<$-maximal, hence S and T are lo-sets. We write ε for the empty iposet.

A *subsumption* of labelled iposets P, Q is a bijection $f : P \to Q$ for which $f(S_P) = S_Q$, $f(T_P) = T_Q$, $f(x) <_Q f(y)$ implies $x <_P y$, and $x \dashrightarrow_P y$, $x \not<_P y$ and $y \not<_P x$ imply $f(x) \dashrightarrow_Q f(y)$. Subsumptions thus respect interfaces, reflect precedence and preserve essential event order. Our definition adapts the standard one [13] to the presence of event orders; intuitively, P has more order, and less concurrency, than Q.

An *isomorphism* of labelled iposets is a subsumption that is an order isomorphism. The event order makes isomorphisms unique [6, Lem. 34]. We write $P \sqsubseteq Q$ if there is a subsumption $P \to Q$ (Q subsumes P) and $P \cong Q$ if P and Q are isomorphic. Isomorphic iposets have the same order and label structure. An *ipomset* is an isomorphism class of labelled iposets. We switch freely between ipomsets and labelled iposets, which is safe due to uniqueness of isomorphisms.

An ipomset P is *discrete* if $<$ is empty and hence $\dashrightarrow$ total. It is an *identity* if $S = P = T$. Discrete ipomsets can therefore be identified with isomorphism classes of ilo-sets. The *singleton ipomsets* are the discrete ilo-sets $[a]$ $[\bullet\, a]$, $[a\, \bullet]$ and $[\bullet\, a\, \bullet]$ for all $a \in \Sigma$. They generate the rational ipomset languages in Section 5.

An ipomset P is an *interval ipomset* if it admits an interval representation [11]: a pair $b, e : P \to \mathbb{R}$ such that $b(x) \leq e(x)$ for all $x \in P$ and $x <_P y$ iff $e(x) < b(y)$ for all $x, y \in P$. We write iPoms and iiPoms for the sets of ipomsets and interval ipomsets, respectively.

Ipomset compositions. The *parallel composition* $P \parallel Q$ of labelled iposets P, Q has the disjoint union $P \sqcup Q$ as carrier set, $S_{P\parallel Q} = S_P \cup S_Q$, $T_{P\parallel Q} = T_P \cup T_Q$, $<_{P\parallel Q} = <_P \cup <_Q$, and $x \dashrightarrow_{P\parallel Q} y$ iff $x \dashrightarrow_P y$, $x \dashrightarrow_Q y$, or $x \in P$ and $y \in Q$. It is a straightforward generalisation of the pomset case, a coproduct of P and Q that extends $\dashrightarrow_P$ and $\dashrightarrow_Q$.

The *gluing composition* $P * Q$ is only defined if $T_P \simeq S_Q$. Its carrier set is the quotient $(P \sqcup Q)_{/x \equiv f(x)}$, where $f : T_P \to S_Q$ is the unique isomorphism. The interfaces are $S_{P*Q} = S_P$ and $T_{P*Q} = T_Q$, $\dashrightarrow_{P*Q}$ is the transitive closure of $\dashrightarrow_P \cup \dashrightarrow_Q$, and $x <_{P*Q} y$ iff $x <_P y$, $x <_Q y$, or $x \in P \setminus T_P$ and $y \in Q \setminus S_Q$. The structural inclusions $P \hookrightarrow P * Q \hookleftarrow Q$ preserve both precedence and event orders.

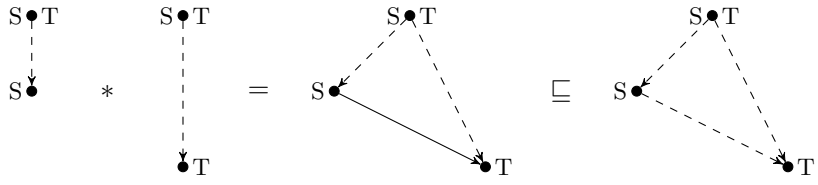
For ipomsets with empty interfaces, $*$ is serial pomset composition [13]. In general, matching interface points are glued, see [6,8] for examples. Both $*$ and $\parallel$ respect isomorphisms and lift to associative, non-commutative operations on ipomsets ($\parallel$ is non-commutative due to the event order). Ipomsets form a category with lo-sets as objects, ipomsets as arrows and $*$ as composition. Interval ipomsets are closed under $*$ [6], but not under $\parallel$ ($a \to b$ is an interval ipomset; $(a \to b) \parallel (a \to b)$ is not). Note that all (isomorphism classes of) interval ipomsets can be generated by gluing ilo-sets [6].

The *width* $\mathrm{w}(P)$ of ipomset P is the cardinality of a maximal $<$-antichain; its *size* is $\#(P) = |P| - \frac{1}{2}(|S| + |T|)$. We glue ipomsets along interfaces below and hence remove half of the interfaces when computing $\#$, which thus may be fractional.

▶ **Lemma 2.** *Let P and Q be ipomsets. Then*
(a) $\mathrm{w}(P \parallel Q) = \mathrm{w}(P) + \mathrm{w}(Q)$ *and* $\#(P \parallel Q) = \#(P) + \#(Q)$,
(b) *if $T_P = S_Q$, then also* $\mathrm{w}(P * Q) = \max(\mathrm{w}(P), \mathrm{w}(Q))$ *and* $\#(P * Q) = \#(P) + \#(Q)$,
(c) *if $P \sqsubseteq Q$, then* $\mathrm{w}(P) \le \mathrm{w}(Q)$.

▶ **Lemma 3.** *For lo-sets $W \subseteq V \subseteq U$, ${}_W V_V * {}_V U_U = {}_W U_U$. For lo-sets $Z \subseteq U, V \subseteq W$ with $Z = U \cap V$ and $W = U \cup V$, ${}_U U_Z * {}_Z V_V \sqsubseteq {}_U W_V$.* ◀

The second fact is illustrated by the following picture.



Languages. An *interval ipomset language* (a *language* for short) is a subset $L \subseteq \mathsf{iiPoms}$ that is *down-closed*: if $P \sqsubseteq Q$ and $Q \in L$, then $P \in L$. We introduce the rational operations $\cup$, $*$, $\parallel$ and (Kleene plus) ${}^+$ for languages:

$$L * M = \{P * Q \mid P \in L,\ Q \in M,\ T_P = S_Q\}{\downarrow}, \qquad L \parallel M = \{P \parallel Q \mid P \in L,\ Q \in M\}{\downarrow},$$
$$L^+ = \bigcup_{n \ge 1} L^n, \qquad \text{for } L^1 = L, L^{n+1} = L * L^n.$$

We write $A{\downarrow} = \{P \in \mathsf{iiPoms} \mid \exists Q \in A.\ P \sqsubseteq Q\}$ for the *down-closure* of a set $A \subseteq \mathsf{iPoms}$. Down-closure is needed because parallel compositions of interval ipomsets may not be interval and gluing and parallel compositions of down-closed languages may not be down-closed.

▶ **Example 4.** $\{[a] \parallel [b]\} = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}]\} = \{[\begin{smallmatrix} a & \bullet \\ b & \bullet \end{smallmatrix}] * [\begin{smallmatrix} \bullet & a \\ \bullet & b \end{smallmatrix}]\}$ is not down-closed.

Both $*$ and $\parallel$ are associative and non-commutative. The identity of $\parallel$ is $\{\varepsilon\}$, that of $*$ is the *identity language* $\mathsf{Id} = \{{}_U U_U \mid U \in \square\}$.

The class of *rational languages* is the smallest class that contains the empty language, the empty-pomset language and the singleton pomset languages

$$\emptyset,\ \{\varepsilon\},\ \{[a]\},\ \{[\bullet\, a]\},\ \{[a\, \bullet]\},\ \{[\bullet\, a\, \bullet]\},\ a \in \Sigma, \tag{2}$$

and that is closed under the rational operations $\cup$, $*$, $\parallel$ and ${}^+$.

The *width* of a language is the maximal width among its elements. Lemma 2 shows that rational languages have finite width; Id has infinite width and is thus not rational.

4 Paths and Languages of iHDAs

Computations or runs of (i)HDAs are higher-dimensional paths that keep track of the cells and face maps traversed [24]. In this section we recall their definition. As an important stepping stone towards a Kleene theorem we then relate paths of (i)HDAs with ipomsets – for a more general class than [6]. We also introduce notions of path equivalence and subsumption. The latter corresponds to ipomset subsumption. Finally, we introduce regular languages.

Paths. A *path* of length n in $X \in \mathbb{I}\square\,\mathbf{Set}$ is a sequence

$$\alpha = (x_0, \varphi_1, x_1, \varphi_2, \dots, \varphi_n, x_n),$$

where the $x_k \in X[U_k]$ are cells and, for all k, either
- $\varphi_k = d^0_A \in \mathbb{I}\square(U_{k-1}, U_k)$, $A \subseteq U_k$ and $x_{k-1} = \delta^0_A(x_k)$ (up-step), or
- $\varphi_k = d^1_B \in \mathbb{I}\square(U_k, U_{k-1})$, $B \subseteq U_{k-1}$, $\delta^1_B(x_{k-1}) = x_k$ (down-step).

We write $x_{k-1} \nearrow^A x_k$ for the up-steps and $x_{k-1} \searrow_B x_k$ for the down-steps in α, generally assuming $A \neq \emptyset \neq B$. We refer to the up- or down-steps (paths $(x_i, \varphi_{i+1}, x_{i+1})$, $0 \leq i < n$) as *steps* in α and write P_X for the set of all paths on X.

▶ **Example 5.** Figure 1 (on the right) in the introduction depicts the three paths

$$\alpha_1 = (\delta^0_{ab}(x) \nearrow^a \delta^0_b(x) \searrow_a \delta^1_{ac}(y)),$$
$$\alpha_2 = (\delta^0_{ab}(x) \nearrow^{ab} x \searrow_b \delta^1_b(x) \nearrow^c y \searrow_{ac} \delta^1_{ac}(y)),$$
$$\alpha_3 = (\delta^0_{ab}(x) \nearrow^b \delta^0_a(x) \searrow_b \delta^0_{ac}(y) \nearrow^{ac} y \searrow_{ac} \delta^1_{ac}(y)).$$

We equip paths with source and target maps as usual: $\ell(\alpha) = x_0$ and $r(\alpha) = x_n$ for path α as above. For $x, y \in X$, we write $\mathsf{P}_X(x,y) = \{\alpha \in \mathsf{P}_X \mid \ell(\alpha) = x, r(\alpha) = y\}$. Any ipc-map $f : X \to Y$ induces a map $f : \mathsf{P}_X \to \mathsf{P}_Y$. For α as above it is $f(\alpha) = (f(x_0), \varphi_1, f(x_1), \varphi_2, \dots, \varphi_n, f(x_n))$. For paths $\alpha = (x_0, \varphi_1, \dots, x_n)$, $\beta = (y_0, \psi_1, \dots, y_m)$ with $r(\alpha) = \ell(\beta)$ the *concatenation* $\alpha * \beta = (x_0, \varphi_1, \dots, x_n, \psi_1, \dots, y_m)$ is defined as expected.

Cell $y \in X$ is *reachable* from cell $x \in X$, denoted $x \preceq y$, if $\mathsf{P}_X(x,y) \neq \emptyset$. This preorder is generated by $\delta^0_A(x) \preceq x \preceq \delta^1_B(x)$ for $x \in X$ and $A, B \subseteq \mathrm{ev}(x)$. We call X *acyclic* if $\preceq$ is a partial order. The reversal on ipc-sets reverses paths and $\preceq$.

From paths to ipomsets. Next we introduce a map ev that computes ipomsets of paths. The interval ipomset $\mathrm{ev}(\alpha)$ of path $\alpha \in \mathsf{P}_X$ is computed recursively:
- If $\alpha = (x)$ has length 0, then $\mathrm{ev}(\alpha) = {}_{\mathrm{ev}(x)}\mathrm{ev}(x)_{\mathrm{ev}(x)}$.
- If $\alpha = (y \nearrow^A x)$, then $\mathrm{ev}(\alpha) = {}_{\mathrm{ev}(x)\setminus A}\mathrm{ev}(x)_{\mathrm{ev}(x)}$.
- If $\alpha = (x \searrow_B y)$, then $\mathrm{ev}(\alpha) = {}_{\mathrm{ev}(x)}\mathrm{ev}(x)_{\mathrm{ev}(x)\setminus B}$.
- If $\alpha = \beta_1 * \cdots * \beta_n$ is a concatenation of steps β_i, then $\mathrm{ev}(\alpha) = \mathrm{ev}(\beta_1) * \cdots * \mathrm{ev}(\beta_n)$.

Interfaces and gluings of ipomsets are essential for this construction. It would not have worked with regular pomsets.

▶ **Example 6.** The ipomset of path α_1 in Example 5 is computed recursively as $\mathrm{ev}(\alpha_1) = \mathrm{ev}(\delta^0_{ab}(x) \nearrow^a \delta^0_b(x)) * \mathrm{ev}(\delta^0_b(x) \searrow_a \delta^1_{ac}(y)) = {}_\emptyset a_a * {}_a a_\emptyset = a$. Those of the other two paths are $\mathrm{ev}(\alpha_2) = a \parallel (b \to c)$ and $\mathrm{ev}(\alpha_3) = b * \left[\begin{smallmatrix} a \\ c \end{smallmatrix}\right]$.

The following fact is immediate from the definition of ev.

▶ **Lemma 7.** *For $\alpha, \beta \in \mathsf{P}_X$ with $r(\alpha) = \ell(\beta)$ and an ipc-map $f : X \to Y$, $\mathrm{ev}(\alpha * \beta) = \mathrm{ev}(\alpha) * \mathrm{ev}(\beta)$ and $\mathrm{ev}(f(\alpha)) = \mathrm{ev}(\alpha)$.* ◀

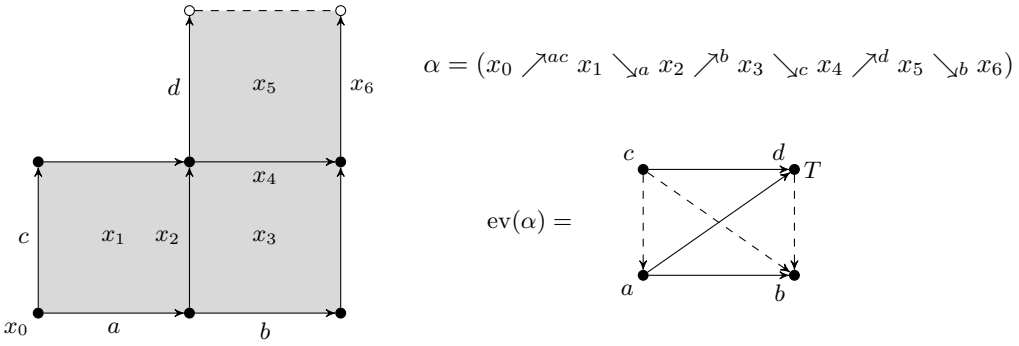


Figure 3 Example of a path α in an iHDA and its interval ipomset $\mathrm{ev}(\alpha)$. Solid arrows show the precedence on $\mathrm{ev}(\alpha)$, dashed arrows the event order. The start interface is empty, the target interface contains the single event d.

Path equivalence and subsumption. *Path equivalence* is the congruence $\simeq$ on P_X generated by $(z \nearrow^A y \nearrow^B x) \simeq (z \nearrow^{A\cup B} x)$, $(x \searrow_A y \searrow_B z) \simeq (x \searrow_{A\cup B} z)$ and $\gamma * \alpha * \delta \simeq \gamma * \beta * \delta$ whenever $\alpha \simeq \beta$. Further, *path subsumption* is the transitive relation $\sqsubseteq$ on P_X generated by $(y \searrow_A w \nearrow^B z) \sqsubseteq (y \nearrow^B x \searrow_A z)$, for disjoint $A, B \subseteq \mathrm{ev}(x)$, $\gamma * \alpha * \delta \sqsubseteq \gamma * \beta * \delta$ whenever $\alpha \sqsubseteq \beta$, and $\alpha \sqsubseteq \beta$ whenever $\alpha \simeq \beta$. We say that β *subsumes* α if $\alpha \sqsubseteq \beta$.

This means that β is more concurrent than α. Both relations preserve sources and targets of paths; they translate to ipomsets as follows (the proof uses Lemmas 3 and 7).

▶ **Lemma 8.** *If $\alpha, \beta \in \mathsf{P}_X$, then $\alpha \sqsubseteq \beta \Rightarrow \mathrm{ev}(\alpha) \sqsubseteq \mathrm{ev}(\beta)$ and $\alpha \simeq \beta \Rightarrow \mathrm{ev}(\alpha) = \mathrm{ev}(\beta)$.* ◀

▶ **Example 9.** It is easy to check that path α_3 in Example 5 is subsumed by α_2, and so are the corresponding pomsets in Example 6: $\mathrm{ev}(\alpha_3) = b * \left[\begin{smallmatrix} a \\ c \end{smallmatrix}\right] \sqsubseteq a \parallel (b \to c) = \mathrm{ev}(\alpha_2)$.

Regular languages. A path $\alpha \in \mathsf{P}_X$ of an (i)HDA X is *accepting* if $\ell(\alpha) \in X_\perp$ and $r(\alpha) \in X^\top$. Let $\mathsf{L}(X) = \{\mathrm{ev}(\alpha) \mid \alpha \in \mathsf{P}_X \text{ is accepting}\}$ be the set of ipomsets recognised by X.

▶ **Proposition 10.** *$\mathsf{L}(X)$ is a language (a down-closed set of interval ipomsets).*

▶ **Proposition 11.** *HDAs and iHDAs recognise the same languages.*

A language is *regular* if it is recognised by an HDA (or an iHDA).

Prop. 10 extends [6, Thm. 95]; see [7] for a proof. The proof of Prop. 11 in [7] uses resolution and closure.

Lem. 2 implies the following.

▶ **Lemma 12.** *Regular languages have finite width.* ◀

An (i)HDA map $f : X \to Y$ is a *weak equivalence* if for every accepting path $\beta \in \mathsf{P}_Y$ there exists an accepting path $\alpha \in \mathsf{P}_X$ with $f(\alpha) = \beta$. The next lemma is shown in [7].

▶ **Lemma 13.** *If $f : X \to Y$ is an (i)HDA-map, then $\mathsf{L}(X) \subseteq \mathsf{L}(Y)$. If f is a weak equivalence, then $\mathsf{L}(X) = \mathsf{L}(Y)$.*

5 Kleene Theorem

In this section we state the Kleene theorem for HDAs and ipomset languages and provide a road-map towards its proof. Technical details are explained in the rest of the paper.

▶ **Theorem 14** (Kleene theorem). *A language is regular if and only if it is rational.*

Proof. By Prop. 16 and Cor. 29 below. ◀

Regular languages are rational. This follows from a translation to the standard automata-theoretic Kleene theorem. It uses the following property which is proved in Section 6 after introducing cylinders.

▶ **Proposition 15.** *If L is regular, then so is $L \setminus \mathsf{Id}$.*

▶ **Proposition 16.** *Regular languages are rational.*

Proof. Suppose $L = (L \cap \mathsf{Id}) \cup (L \setminus \mathsf{Id})$ is regular. First, $L \cap \mathsf{Id}$ is a finite set of identity ipomsets and thus rational. Second, $L \setminus \mathsf{Id}$ is regular by Prop. 15; we show that it is rational. Let X be an HDA that recognises $L \setminus \mathsf{Id}$. Then $X_\perp \cap X^\top = \emptyset$ (otherwise, X accepts an identity ipomset). Let G be an automaton with alphabet $\mathbb{I}\square$ whose states are cells of X and whose transitions are, for all $x \in X[U]$, $A \subseteq U \in \square$,

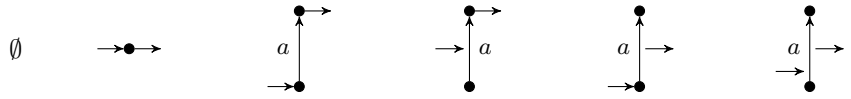
- $\delta^0_A(x) \to x$ labelled with ${}_{(U\setminus A)}U_U$
- $x \to \delta^1_A(x)$ labelled with ${}_U U_{(U\setminus A)}$.

Start and accept states of G are start and accept cells of X. Accepting runs of G are then exactly accepting paths of X. Every such run of G contains at least one transition. Hence, if $U_1 U_2 \cdots U_n \in \mathbb{I}\square^*$ is the word of a run of G, then $U_1 * U_2 * \cdots * U_n$ is the interval ipomset of the corresponding path in X. By the standard Kleene theorem, $L(G)$ is represented by a regular expression $w(U_1, \dots, U_n)$ with operations $\cup$, $*$ and $(-)^+$ for $U_i \in \mathbb{I}\square$. But each $U_i = e_i^1 \parallel \cdots \parallel e_i^{k(i)}$, a parallel composition of singleton ipomsets. Thus $\mathsf{L}(X)$ is represented by $w(e_1^1 \parallel \cdots \parallel e_1^{k(1)}, \dots, e_n^1 \parallel \cdots \parallel e_n^{k(n)})$ and therefore rational. ◀

Rational languages are regular. We need to show, as usual, that the generating languages are regular and that the rational operations preserve regularity.

▶ **Proposition 17.** *All languages in* (2) *are regular.*

Proof. The languages in (2) are recognised by the HDAs



◀

▶ **Proposition 18.** *Unions of regular languages are regular.*

Proof. $\mathsf{L}(X \sqcup Y) = \mathsf{L}(X) \cup \mathsf{L}(Y)$, where $X \sqcup Y$ is coproduct. ◀

In order to show the same for parallel compositions of regular languages, we introduce *tensor products* of HDAs. For HDAs X and Y, $X \otimes Y$ is the HDA defined, for $U, V, W \in \square$, $x \in X[V]$, $y \in Y[W]$, $A, B \subseteq U$, by

$$(X \otimes Y)[U] = \bigcup_{V \parallel W = U} X[V] \times Y[W], \qquad \delta_{A,B}(x,y) = (\delta_{A\cap V, B\cap V}(x), \delta_{A\cap W, B\cap W}(y)),$$

and $(X \otimes Y)_\perp = X_\perp \times Y_\perp$, $(X \otimes Y)^\top = X^\top \times Y^\top$.

▶ **Proposition 19.** $\mathsf{L}(X \otimes Y) = \mathsf{L}(X) \parallel \mathsf{L}(Y)$ *for all HDAs* X, Y*. As a consequence, parallel compositions of regular languages are regular.*

A proof for a restricted class of HDAs is in [6]; we show a complete proof in [7].

Analogous proofs for $*$ and $^+$ are much harder. They need gluing operations on HDAs and additional machinery.

An ipomset P is *separated* if $P \setminus (S_P \cup T_P) \neq \emptyset$ (some of its elements are not in an interface). A language is *separated* if its elements are separated.

An (i)HDA X is *start simple* if it has exactly one start cell, *accept simple* if it has exactly one accept cell, and *simple* if it is both start and accept simple. A regular language is *simple* if it is recognised by a simple iHDA. Yet this reduces expressivity.

▶ **Example 20.** The HDA X with a single 0-cell x, a 1-loop labelled a, and $X_\perp = X^\top = \{a\}$ is simple and recognises the language of all ipomsets $[\bullet\, a \cdots a\, \bullet]$, but no simple iHDA does.

Next we prove two lemmas about simple and separated languages.

▶ **Lemma 21.** *Regular languages are unions of simple regular languages.*

Proof. Prop. 11 allows us to work with an iHDA X, say. Denote $X_\perp = \{x^i_\perp\}_{i=1}^m$ and $X^\top = \{x^\top_j\}_{j=1}^n$. For each tuple (i,j) let X^j_i be the iHDA with the same underlying ipc-set as X and $(X^j_i)_\perp = \{x^i_\perp\}$, $(X^j_i)^\top = \{x^\top_j\}$. We have $\mathsf{L}(X) = \bigcup_{i,j} \mathsf{L}(X^j_i)$. ◀

▶ **Lemma 22.** *For L of finite width with $L \cap \mathsf{Id} = \emptyset$, and n sufficiently large, L^n is separated.*

Proof. For every $Q \in L^n$ there exists $P = P_1 * \ldots * P_n$ such that $P_k \in L$ and $Q \sqsubseteq P$. As $\#(P_k) \geq \frac{1}{2}$, additivity of size implies $\#(Q) = \#(P) = \#(P_1) + \ldots + \#(P_n) \geq \frac{n}{2}$ and therefore $|S_Q|, |T_Q| \leq \mathrm{w}(Q) \leq \mathrm{w}(P) = \max_k \mathrm{w}(P_k) \leq \mathrm{w}(L)$, since gluing composition does not increase width. Eventually, $|S_Q| + |T_Q| \leq 2\,\mathrm{w}(L) < n \leq 2\,\#(Q) = 2|Q| - |S_Q| - |T_Q|$ holds for $n \geq 2\,\mathrm{w}(L) + 1$ and therefore $|S_Q| + |T_Q| < |Q|$. ◀

Let X, Y be simple HDAs with $X^\top = \{x^\top\}$, $Y_\perp = \{y_\perp\}$, and $\mathrm{ev}(x^\top) = \mathrm{ev}(y_\perp) = U$. The *gluing composition* of X and Y is the HDA

$$X * Y = \operatorname{colim}\left(X \xleftarrow{\iota_{x^\top}} \square^U \xrightarrow{\iota_{y_\perp}} Y\right)$$

with $(X * Y)_\perp = X_\perp$, $(X * Y)^\top = Y^\top$. In other words, we identify the accept cell of X with the start cell of Y, as well as their corresponding faces. In general, $\mathsf{L}(X * Y)$ is a strict superset of $\mathsf{L}(X) * \mathsf{L}(Y)$; but in Sect. 6 we will introduce properties of *start* and *accept proper* which ensure the following.

▶ **Proposition 23.** *Let X, Y be simple iHDAs. If X is accept simple and accept proper, and Y is start simple and start proper, then* $\mathsf{L}(\mathsf{Cl}(X) * \mathsf{Cl}(Y)) = \mathsf{L}(X) * \mathsf{L}(Y)$.

Proof. This is a special case of Prop. 52 which is shown in [7]. ◀

▶ **Proposition 24.** *Every simple regular language is recognised by a start simple and start proper iHDA as well as by an accept simple and accept proper iHDA.*

The proof can be found in the next section.

▶ **Proposition 25.** *Gluing compositions of simple regular languages are regular.*

Proof. Let X, Y be simple iHDAs recognising L and M, respectively. We can assume that X is accept simple and accept proper and Y is start simple and start proper (Prop. 24). Thus, by Prop. 23, $\mathsf{L}(\mathsf{Cl}(X) * \mathsf{Cl}(Y)) = \mathsf{L}(X) * \mathsf{L}(Y) = L * M$. ◀

▶ **Proposition 26.** *The Kleene plus of a separated regular language is regular.*

The proof is in [7]. Below we show that the additional assumptions above may be removed, finishing the proof of the Kleene theorem.

▶ **Proposition 27.** *Gluing compositions of regular languages are regular.*

Proof. Suppose L and M are regular. By Lem. 21, $L = \bigcup_i L_i$ and $M = \bigcup_j M_j$ for simple regular languages L_i and M_j. Then $L * M = \left(\bigcup_i L_i\right) * \left(\bigcup_j M_j\right) = \bigcup_i \bigcup_j L_i * M_j$ is regular by Propositions 18 and 25. ◀

▶ **Proposition 28.** *The Kleene plus of a regular language is regular.*

Proof. Suppose L is regular. If $L \cap \mathsf{Id} = \emptyset$, then L^n is separated for sufficiently large n by Lem. 22. In this case, $L^+ = \bigcup_{i=1}^n L^i \cup \left(\bigcup_{i=1}^n L^i\right) * (L^n)^+$ is regular by Propositions 18, 27 and 26. Otherwise, if $L \cap \mathsf{Id} \neq \emptyset$, then $L^+ = ((L \cap \mathsf{Id}) \cup (L \setminus \mathsf{Id}))^+ = (L \cap \mathsf{Id}) \cup (L \setminus \mathsf{Id})^+$ is regular by Prop. 15 and Prop. 18. ◀

▶ **Corollary 29.** *Every rational language is regular.*

Proof. By Propositions 17, 18, 19, 27, and 28 ◀

The remaining proofs (Prop. 15, 25, 26). Prop. 15 needs translations between HDAs and iHDAs via resolution and closure, established in the next section.

Next we briefly explain the proof of Prop. 25. Sequential compositions $X * Y$ of standard automata require some care. When accept states of X have outgoing transitions or start states of Y have incoming ones, one cannot simply identify accept states of X with start states of Y. The language of the resulting automaton may contain more words than $\mathsf{L}(X) * \mathsf{L}(Y)$. To alleviate this, one usually replaces X and Y by equivalent "proper" automata without such transitions. We proceed along similar lines for iHDA. Suppose X, Y are simple iHDA with $X^\top = \{x^\top\}$, $Y_\perp = \{y_\perp\}$ and $\mathrm{ev}(x^\top) = \mathrm{ev}(y_\perp)$ ($\mathrm{ev}(x^\top) \neq \mathrm{ev}(y_\perp)$ implies $\mathsf{L}(X) * \mathsf{L}(Y) = \emptyset$).

We show that any iHDA may be converted into a *start proper* or *target proper* iHDA that recognises the same language. Start-properness implies that any start cell is $\preceq$-minimal and that different start cells do not share any faces. Target-properness is defined similarly.

We introduce *cylinders* in Sec 6 to perform this conversion. For ipc-sets X, Y, Z and image-disjoint maps $f : Y \to X$, $g : Z \to X$, we construct an ipc-set $\mathrm{C}(f, g)$ together with maps $\tilde{f} : Y \to \mathrm{C}(f, g)$, $\tilde{g} : Z \to \mathrm{C}(f, g)$ and $p : \mathrm{C}(f, g) \to X$, see Fig. 4. The image of $\tilde{f}$ is $\preceq$-minimal, that of $\tilde{g}$ is $\preceq$-maximal, and $p \circ \tilde{f} = f$ and $p \circ \tilde{g} = g$. This construction is inspired by algebraic topology: $\tilde{f}$ and $\tilde{g}$ are reminiscent of cofibrations and p of a trivial fibration. Here we only show that p has suitable lifting properties. We also use cylinders for Prop. 15.

We may therefore assume that X is accept proper and Y start proper. Gluing iHDAs is intricate due to the missing faces of start and accept cells. So, after rearranging iHDAs, we convert them back into HDAs using closure. Then we show that the gluing of $\mathsf{Cl}(X)$ and $\mathsf{Cl}(Y)$ along their accept and start cells yields an HDA that recognises $\mathsf{L}(X) * \mathsf{L}(Y)$.

Finally, the proof of Prop. 26 is similar but more sophisticated. Cylinders allow us to separate start cells from each other (same for accept cells), but we are not able to separate start cells from accept cells. Thus, we require the separability assumption.

The tools needed for gluing closures of proper iHDAs are developed in [7].

6 Cylinders

In the final technical section of this paper we describe the cylinder construction mentioned above, as it may be of independent interest. Omitted proofs are shown in [7].

Initial and final inclusions. A sub-ipc-set $Y \subseteq X$ is *initial* if it is down-closed with respect to the reachability preorder $\preceq$. Equivalently, $\delta^1_B(x) \in Y$ implies $x \in Y$ for all $x \in X[U]$ and $B \subseteq U \setminus T_U$. By reversal, Y is *final* if it is up-closed with respect to $\preceq$ or, equivalently, $\delta^0_A(x) \in Y$ implies $x \in Y$. An *initial* (*final*) *inclusion* is an injective ipc-map whose image is an initial (final) sub-ipc-set.

▶ **Lemma 30.** *If $f : Y \to X$ is an initial or final inclusion, then so is $\mathsf{Cl}(f) : \mathsf{Cl}(Y) \to \mathsf{Cl}(X)$.*

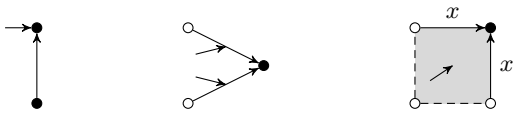
Proper iHDAs. The *start* and *accept maps* of iHDA X are the ipc-maps

$$\iota^X_\bot = \coprod_{x \in X_\bot} \iota_x : \coprod_{x \in X_\bot} \square^{\mathrm{iev}(x)} \to X, \qquad \iota^\top_X = \coprod_{x \in X^\top} \iota_x : \coprod_{x \in X^\top} \square^{\mathrm{iev}(x)} \to X.$$

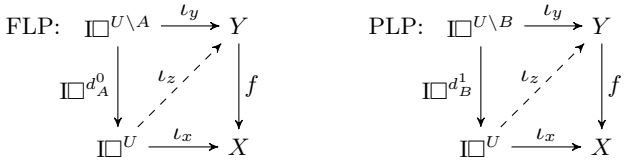
We call an iHDA *start proper* if its start map is an initial inclusion, *accept proper* if its accept map is a final inclusion, and *proper* if it is start proper, accept proper and the images of the start map and the accept map are disjoint.

▶ **Lemma 31.** *All start cells of start proper iHDAs are $\preceq$-minimal; all accept cells of accept proper iHDAs are $\preceq$-maximal.*

The condition of Lem. 31 is not sufficient for properness. The diagrams show examples of iHDAs that are not start proper; edges marked with x are identified.



Lifting properties. An ipc-map $f : Y \to X$ has the *future lifting property* (*FLP*) if for every up-step $\alpha = (\delta^0_A(x) \nearrow^A x)$ in X and every $y \in Y$ such that $f(y) = \delta^0_A(x)$ there is an up-step $\beta = (y \nearrow^A z)$ in Y such that $f(\beta) = \alpha$. The *past lifting property* (*PLP*) is defined by reversal. FLP and PLP are equivalent to the lifting property for the following diagrams.



The next lemma is immediate from the definitions.

▶ **Lemma 32.** *An ipc-map $f : Y \to X$ has the FLP if and only if, for every $\alpha \in \mathsf{P}_X$ and $y \in f^{-1}(\ell(\alpha))$, there exists a path $\beta \in \mathsf{P}_Y$ such that $\ell(\beta) = y$ and $f(\beta) = \alpha$. An analogous property holds for PLP.* ◀

Let $f : Y \to X$ be an ipc-map, let $S, T \subseteq X$ (subsets, but not necessarily sub-ipc-sets). Then f has the *total lifting property* (*TLP*) with respect to S and T if for every path $\alpha \in \mathsf{P}_X$ with $\ell(\alpha) \in S$ and $r(\alpha) \in T$ and every $y \in f^{-1}(\ell(\alpha))$ and $z \in f^{-1}(r(\alpha))$, there exists a path $\beta \in \mathsf{P}_Y(y, z)$ such that $f(\beta) = \alpha$.

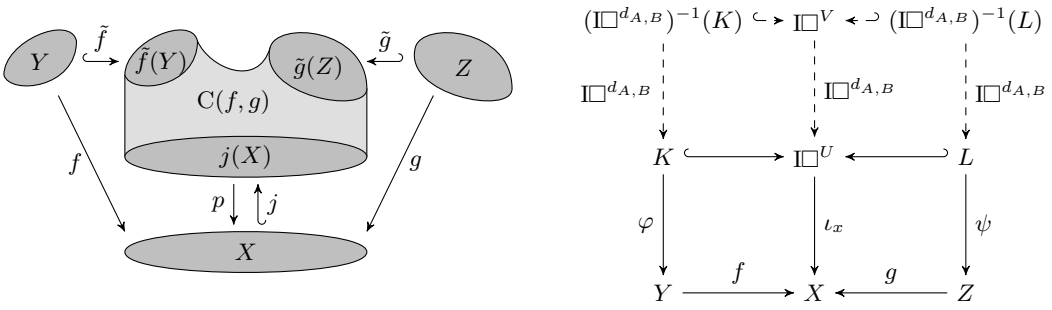


Figure 4 The cylinder $\mathrm{C}(f,g)$ and a diagram defining its cell.

▶ **Proposition 33.** *Let $f : Y \to X$ be an iHDA map such that the functions $Y_\perp \to X_\perp$ and $Y^\top \to X^\top$ induced by f are surjective. Assume that one of the following holds.*
(a) *f has the FLP and $Y^\top = f^{-1}(X^\top)$,*
(b) *f has the PLP and $Y_\perp = f^{-1}(X_\perp)$,*
(c) *f has the TLP with respect to $X_\perp$ and $X^\top$.*
Then f is a weak equivalence.

Cylinders. Let $X, Y, Z \in \mathrm{I}\square\,\mathbf{Set}$ and $f : Y \to X$, $g : Z \to X$ ipc-maps, assuming f and g to have disjoint images; this is not used directly in the construction, but crucial in proofs.

The *cylinder* $\mathrm{C}(f,g)$ is the ipc-set such that $\mathrm{C}(f,g)[U]$ is the set of (x, K, L, φ, ψ), where $x \in X[U]$, K is an initial and L is a final sub-ipc-set of $\mathrm{I}\square^U$, $\varphi : K \to Y$ and $\psi : L \to Z$ are ipc-maps satisfying $f \circ \varphi = \iota_x|_K$ and $g \circ \psi = \iota_x|_L$. For $d_{A,B} \in \mathrm{I}\square(V,U)$ and $(x, K, L, \varphi, \psi) \in \mathrm{C}(f,g)[U]$, we put

$$\delta_{A,B}(x, K, L, \varphi, \psi) = (\delta_{A,B}(x), (\mathrm{I}\square^{d_{A,B}})^{-1}(K), (\mathrm{I}\square^{d_{A,B}})^{-1}(L), \varphi \circ \mathrm{I}\square^{d_{A,B}}, \psi \circ \mathrm{I}\square^{d_{A,B}}).$$

Equivalently, $\mathrm{C}(f,g)[U]$ is the set of commutative diagrams of solid arrows in Fig. 4 and the face map $\delta_{A,B}$ composes the diagram with the dashed arrows. The following is then clear (recall that $f(Y) \cap g(Z) = \emptyset$).

▶ **Lemma 34.** *For every $(x, K, L, \varphi, \psi) \in \mathrm{C}(f,g)$ we have $K \subseteq (\iota_x)^{-1}(f(Y))$ and $L \subseteq (\iota_x)^{-1}(g(Z))$. Thus $K \cap L = \emptyset$, $x \in f(Y)$ implies $L = \emptyset$, and $x \in g(Z)$ implies $K = \emptyset$.*

$\mathrm{C}(f,g)$ is equipped with the ipc-maps shown in Fig. 4. They are defined by $j(x) = (x, \emptyset, \emptyset, \emptyset, \emptyset)$, $p(x, K, L, \varphi, \psi) = x$, $\tilde{f}(y) = (f(y), \mathrm{I}\square^{\mathrm{iev}(y)}, \emptyset, \iota_y, \emptyset)$, $\tilde{g}(z) = (g(z), \emptyset, \mathrm{I}\square^{\mathrm{iev}(z)}, \emptyset, \iota_z)$. Below we collect some of their properties.

▶ **Lemma 35.**
(a) *$p \circ \tilde{f} = f$, $p \circ \tilde{g} = g$, $p \circ j = \mathrm{id}_X$.*
(b) *$\tilde{f}$ is an initial inclusion and $\tilde{f}(Y) = \{(x, K, L, \varphi, \psi) \in \mathrm{C}(f,g) \mid K = \mathrm{I}\square^{\mathrm{iev}(x)},\ L = \emptyset\}$.*
(c) *$\tilde{g}$ is a final inclusion and $\tilde{g}(Z) = \{(x, K, L, \varphi, \psi) \in \mathrm{C}(f,g) \mid L = \mathrm{I}\square^{\mathrm{iev}(x)},\ K = \emptyset\}$.*
(d) *j is an inclusion and $j(X) = \{(x, \emptyset, \emptyset, \emptyset, \emptyset) \in \mathrm{C}(f,g)\}$.*
(e) *$\tilde{f}(Y)$, $\tilde{g}(Z)$ and $j(X)$ are pairwise disjoint.*

▶ **Proposition 36.** *The projection $p : \mathrm{C}(f,g) \to X$ has the FLP and PLP, and the TLP with respect to $f(Y)$ and $g(Z)$.*

Proof of Prop. 24. We show only the first claim; the second follows by reversal. Let L be recognised by a simple iHDA X with start cell $x_\bot \in X[U]$. Let Y be the iHDA with underlying precubical set $\mathrm{C}(\iota^X_\bot, \emptyset)$ ($\emptyset : \emptyset \to X$ is the empty map). Let $y_\bot = (x_\bot, \mathbb{I}\square^U, \emptyset, \mathrm{id}_{\mathbb{I}\square^U}, \emptyset)$ be the only start cell of Y and $Y^\top = p^{-1}(X^\top)$. Since $\iota^Y_\bot = \widetilde{\iota^X_\bot}$, Y is start proper (Lem. 35(b)). The projection $p : Y \to X$ has the FLP (Prop. 36); thus $\mathsf{L}(Y) = \mathsf{L}(X) = L$ (Prop. 33(a)). ◀

Proof of Prop. 15. Suppose first that L is simple. Let X be a start simple and start proper iHDA recognising L (Prop. 24) and let Y be the iHDA with same underlying ipc-set and start cells as X and accept cells $Y^\top = X^\top \setminus X_\bot$. By Lem. 31, an accepting path $\alpha \in \mathsf{P}_X$ is accepting in Y iff it has positive length ($\mathrm{ev}(\alpha)$ is not an identity), thus $\mathsf{L}(Y) = \mathsf{L}(X) \setminus \mathsf{Id} = L \setminus \mathsf{Id}$ is regular. If L is not simple, then let $L = \bigcup_i L_i$ be a finite sum of simple languages. Then $L \setminus \mathsf{Id} = (\bigcup_i L_i) \setminus \mathsf{Id} = \bigcup_i (L_i \setminus \mathsf{Id})$ is regular by the first case and Prop. 18. ◀

7 Conclusion

Automata accept languages, but higher-dimensional automata (HDAs) have so far been an exception to this rule. We have proved a Kleene theorem for HDAs, connecting models to behaviours through an equivalence between regular and rational languages.

Showing that regular languages are rational was quite direct, but the converse direction required some effort. One reason is that HDAs may be glued not only at states, but also at higher-dimensional cells. This in turn led us to consider languages of pomsets with interfaces (ipomsets) and to equip HDAs with interfaces (iHDAs), too. After showing that HDAs and iHDAs recognise the same languages, we used topology-inspired constructions to glue (i)HDAs and show that rational operations on languages can be reflected by operations on (i)HDAs.

Kleene theorems build bridges between machines and languages, and there is a vast literature on the subject. In non-interleaving concurrency, one school considers (Mazurkiewicz) trace languages. Zielonka introduces asynchronous automata and shows that languages are regular iff they are recognisable [26]. Droste's automata with concurrency relations have similar properties [4]. Yet not all rational trace languages (generated from singletons using union, concatenation and Kleene star) are recognisable [3]. Trace languages use a binary notion of independence and already 2-dimensional HDAs may exhibit behaviour that cannot be captured by trace languages [12].

Another school studies Kleene theorems for series-parallel pomset languages and automata models for these, such as branching and pomset automata [17, 19], and for Petri automata [2]. Series-parallel pomsets are incomparable to the interval orders accepted by Petri nets or HDAs, see [8, 25].

HDAs have been developed first of all with a view on operational, topological and geometric aspects, see [10] and the extensive bibliography of [24]. Languages have only been introduced recently [6]. Topological intuition has also guided our work, for example in the cylinder construction. Partial HDAs [9] were introduced for defining open maps and unfoldings on HDAs. HDAs with interfaces are a special case of partial HDAs, and our tools and techniques should be useful for those as well.

Our new formalisation of (i)HDAs as presheaves over a category of labelled ordered sets opens up connections to presheaf automata [22], coalgebra, and open maps [16], which we intend to explore. Finally, our introduction of iHDA morphisms akin to cofibrations and trivial fibrations hints at factorisation systems for HDAs. Weak factorisation systems and model categories have been considered in a bisimulation context, for example in [18], so we wonder about the connection.

References

1 Marc Bezem, Thierry Coquand, and Simon Huber. A model of type theory in cubical sets. In Ralph Matthes and Aleksy Schubert, editors, *TYPES*, volume 26 of *Leibniz International Proceedings in Informatics*, pages 107–128. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.TYPES.2013.107.

2 Paul Brunet and Damien Pous. Petri automata. *Logical Methods in Computer Science*, 13(3), 2017. doi:10.23638/LMCS-13(3:33)2017.

3 Christian Choffrut and Leucio Guerra. Logical definability of some rational trace languages. *Mathematical Systems Theory*, 28(5):397–420, 1995. doi:10.1007/BF01185864.

4 Manfred Droste. A Kleene theorem for recognizable languages over concurrency monoids. In Serge Abiteboul and Eli Shamir, editors, *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 388–399. Springer, 1994. doi:10.1007/3-540-58201-0_84.

5 Uli Fahrenberg. *Higher-Dimensional Automata from a Topological Viewpoint*. PhD thesis, Aalborg University, Denmark, 2005.

6 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.

7 Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. A Kleene theorem for higher-dimensional automata. *CoRR*, abs/2202.03791, 2022. arXiv:2202.03791.

8 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.

9 Uli Fahrenberg and Axel Legay. Partial higher-dimensional automata. In Lawrence S. Moss and Pawel Sobocinski, editors, *CALCO*, volume 35 of *Leibniz International Proceedings in Informatics*, pages 101–115. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CALCO.2015.101.

10 Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, and Martin Raussen. *Directed Algebraic Topology and Concurrency*. Springer, 2016. doi:10.1007/978-3-319-15398-8.

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

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

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

14 Marco Grandis. *Directed algebraic topology: models of non-reversible worlds*. New mathematical monographs. Cambridge University Press, 2009.

15 Marco Grandis and Luca Mauri. Cubical sets and their site. *Theory and Applications of Categories*, 11(8):185–211, 2003.

16 André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

17 Tobias Kappé, Paul Brunet, Bas Luttik, Alexandra Silva, and Fabio Zanasi. On series-parallel pomset languages: Rationality, context-freeness and automata. *Journal of Logic and Algebraic Methods in Programming*, 103:130–153, 2019. doi:10.1016/j.jlamp.2018.12.001.

18 Alexander Kurz and Jiří Rosický. Weak factorizations, fractions and homotopies. *Applied Categorical Structures*, 13(2):141–160, 2005.

19 Kamal Lodaya and Pascal 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.

20 Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Springer, 1992.

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

22 Pawel Sobocinski. Relational presheaves, change of base and weak simulation. *J. Comput. Syst. Sci.*, 81(5):901–910, 2015. doi:10.1016/j.jcss.2014.12.007.

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

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

25 Walter Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, volume 625 of *Lecture Notes in Computer Science*. Springer, 1992. doi:10.1007/3-540-55767-9.

26 Wiesław Zielonka. Notes on finite asynchronous automata. *RAIRO - Theoretical Informatics and Applications*, 21(2):99–135, 1987. doi:10.1051/ita/1987210200991.

A Definitions of HDAs

Precubical sets and HDAs are used in a few different incantations in the literature, all more or less equivalent. We expose some of these here in order to relate them to the setting in this paper; we make no claim as to completeness.

Precubical sets. according to Grandis [14, 15] are presheaves on a small category $\square_{\mathrm{G}}$ which is generated by the following data:

- objects are $\{0,1\}^n$ for $n \geq 0$;
- elementary coface maps $d_i^\nu : \{0,1\}^n \to \{0,1\}^{n+1}$, for $i = 1, \dots, n+1$ and $\nu = 0, 1$, are given by $d_i^\nu(t_1, \dots, t_n) = (t_1, \dots, t_{i-1}, \nu, t_i, \dots, t_n)$.

Elementary coface maps compose to coface maps $\{0,1\}^m \to \{0,1\}^n$ for $n \geq m$. See also [5].

$\square_{\mathrm{G}}$-sets, *i.e.*, elements $X \in \mathbf{Set}^{\square_{\mathrm{G}}^{\mathrm{op}}}$, are then graded sets $X = \{X_n\}_{n\geq 0}$ (where $X_n = X[\{0,1\}^n]$) together with face maps $X_n \to X_m$ for $n \geq m$. The elementary face maps are denoted $\delta_i^\nu = X[d_i^\nu]$, and they satisfy the *precubical identity*

$$\delta_i^\nu \delta_j^\mu = \delta_{j-1}^\mu \delta_i^\nu \quad (i < j) \tag{3}$$

for $\nu, \mu \in \{0,1\}$.

The above elementary description of $\square_{\mathrm{G}}$-sets may be taken as definition, allowing one to avoid any talk about presheaves. For example, van Glabbeek [24] defines a precubical set $Q = (Q, s, t)$ as a family of sets $(Q_n)_{n\geq 0}$ and maps $s_i : Q_n \to Q_{n-1}$, $1 \leq i \leq n$, such that $\alpha_i \circ \beta_j = \beta_{j-1} \circ \alpha_i$ for all $1 \leq i < j \leq n$ and $\alpha, \beta \in \{s, t\}$; this is evidently equivalent to the above.

The paper [6] introduces another base category, $\square_{\mathrm{Z}}$, given as follows:

- objects are totally ordered sets $(S, \dashrightarrow_S)$;
- morphisms $S \to T$ are pairs (f, ε), where $f : S \hookrightarrow T$ is an order preserving injection and $\varepsilon : T \to \{0, \lrcorner, 1\}$ fulfils $f(S) = \varepsilon^{-1}(\lrcorner)$.

(The element $\lrcorner$ stands for "executing".) Letting $A = \varepsilon^{-1}(0)$ and $B = \varepsilon^{-1}(1)$, the above notion of morphisms is equivalent to having triples (f, A, B) consisting of $f : S \hookrightarrow T$ (order preserving and injective) and $A, B \subseteq T$ such that $T = A \sqcup f(S) \sqcup B$ (disjoint union). Except for the labelling, this is the same as our definition of $\square$ in Section 2. ([6] also makes a connection to [1].)

Then [6] goes on to show that the full subcategory of $\square_{\mathrm{Z}}$ spanned by objects $\emptyset$ and $\{1, \dots, n\}$ for $n \geq 1$ is skeletal and equivalent to $\square_{\mathrm{Z}}$. Moreover, this subcategory, $\square_{\mathrm{Z}}$, is shown to be isomorphic to $\square_{\mathrm{G}}$, and that the presheaf categories on $\square_{\mathrm{Z}}$ and on $\square_{\mathrm{Z}}$ (and thus also on $\square_{\mathrm{G}}$) are uniquely naturally isomorphic. It is clear that $\square_{\mathrm{Z}}$ is a representative of the quotient of $\square_{\mathrm{Z}}$ under isomorphisms, so except for the labelling, this is again the same as our $\square$ of Section 2.

The advantage of $\square_{\mathrm{Z}}$, and of our $\square$, over the skeleton versions is that the precubical identity (3) is automatic and that there is a built-in notion of events, *i.e.*, in a $\square_{\mathrm{Z}}$-set X, a cell $x \in X[U]$ has events U.

HDAs. are, generally speaking, labelled precubical sets (on the alphabet Σ) with specified start and accept cells. The labelling may be obtained using the labelling object $!\Sigma$ [12]. This is the precubical set with $!\Sigma_n = \Sigma^n$ and $\delta^\nu_i((a_1, \dots, a_n)) = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$, and a labelled precubical set is then a precubical map $X \to !\Sigma$: an object of the slice category of precubical sets over $!\Sigma$.

A labelling $\lambda : X \to !\Sigma$ induces a function $\lambda_1 : X_1 \to \Sigma$ with the property that for all $x \in X_2$, $\lambda_1(\delta^0_1(x)) = \lambda_1(\delta^1_1(x))$ and $\lambda_1(\delta^0_2(x)) = \lambda_1(\delta^1_2(x))$. Conversely, any such function extends uniquely to a precubical map $X \to !\Sigma$ [6, Lem. 14], so that λ_1 may be taken as the definition of labelling instead. This is the approach taken in [24], where an HDA is defined as a precubical set Q equipped with a function $\lambda_1 \to \Sigma$ such that $\lambda_1(s_i(q)) = \lambda_1(t_i(q))$ for all $q \in Q_2$ and $i = 1, 2$, and subsets of start and accept states $I, F \subseteq Q_0$.

Another observation made in [6] is that regarded as a presheaf, $!\Sigma(S) = \mathbf{Set}(S, \Sigma)$, hence $!\Sigma$ is representable in $\mathbf{Set}$ via the forgetful functor $\square_{\mathrm{Z}} \to \mathbf{Set}$. This means that the labelling may be integrated into the base category, turning $\square_{\mathrm{Z}}$ into our $\square$ with objects being labelled totally ordered sets. Using $\square$ instead of $\square_{\mathrm{Z}}$ allows us to avoid working in the slice category: everything is labelled from the outset.

To sum up, let X be an HDA in our sense, then the corresponding HDA $(Q, s, t, \lambda_1, I, F)$ in the sense of [24] is given as follows.

- $Q_n = \coprod_{U \in \square,\ |U|=n} X[U]$.
- If $x \in X[U]$, then $s_i(x) = \delta^0_u(x)$ and $t_i(x) = \delta^1_u(x)$, where $u \in U$ is the i-th smallest element of U in the order $\dashrightarrow_U$.
- If $x \in X[U] \subseteq Q_1$ with $U = (\{e\}, \emptyset, \lambda(e) = a)$, then $\lambda_1(x) = a$.
- $I = X_\bot$, $F = X^\top$.

Conversely, let $(Q, s, t, \lambda_1, I, F)$ be an HDA as in [24]. There exist unique labelling functions $\lambda_n : Q_n \to \Sigma^n$ such that $\lambda_{n-1}(\alpha_i(q)) = \delta_i(\lambda_n(q))$ [6, Lem. 14], where $\alpha \in \{s, t\}$ and δ_i skips the i-th element of a sequence. We construct an HDA X as follows.

- $X[U] = \{q \in Q_n \mid \lambda_n(q) = U\}$ for $U \in \square$ and $|U| = n$.
- $\delta^0_a(q) = s_i(q)$ and $\delta^1_a(q) = t_i(q)$ for $q \in X[U]$ and $a \in U$ the i-th smallest element of U in the order $\dashrightarrow_U$. The remaining face maps are compositions of these.
- $X_\bot = I$, $X^\top = F$.

Finally, the only difference between van Glabbeek's HDAs and ours is that we allow start and accept cells that are not necessarily vertices. Our definition of HDAs, based on lo-sets, treats the elements of these sets as labelled events that are regarded either as unstarted, executed, or terminated. In this way, they incorporate events in a direct manner. This makes our definition of HDAs into a model that combines event-based and state-based models of concurrency.