Irrationality of Process Replication for Higher-Dimensional Automata

Thomas Baronner[1[0009−0006−7710−3085]], Henning Basold[2[0000−0001−7610−8331]], and Márton Hablicsek[3[0000−0001−7549−9713]]

[1] Student, Leiden University thomasbaronner@gmail.com
[2] LIACS, Leiden University h.basold@liacs.leidenuniv.nl
[3] MI, Leiden University m.hablicsek@math.leidenuniv.nl

Abstract Higher-dimensional automata (HDA) are a formalism to faithfully model the behaviour of concurrent systems. For ordinary automata, there is a correspondence between regular expressions, regular languages and finite automata, which provides a powerful link between algebraic proofs and operational behaviour. It has been shown by Fahrenberg et al. that finite HDA correspond with interfaced interval pomset languages generated by sequential and parallel composition and non-empty iteration, and thereby to a variant of Kleene algebras (KA) with parallel composition. It is known that this correspondence cannot be extended to concurrent KA, which additionally have process replication. An alternative to finite HDA are locally finite HDA, in which every state can only reach finitely many other states, and finitely branching HDA. In this paper, we show that both classes of HDA are closed under process replication and thus models of concurrent KA. To achieve this, we prove that the category of HDA is locally finitely presentable, where the finite HDA generate all other HDA. We then prove that this has the unfortunate side-effect that all HDA are locally finite, which means that the correspondence with concurrent KA trivialises. Similarly, we also show that, even though finitely branching HDA are closed under process replication, the resulting HDA necessarily have infinitely many initial states.

Keywords: Higher-dimensional automata · Process replication · Concurrent Kleene Algebras · Locally finitely presentable categories

1 Introduction

Automata theory has as a core goal that problems, like deciding language membership, should be solved by finitary means. With this goal in mind, research on automata typically strives for a correspondence between certain kinds of finitary automata, languages, syntactic expressions, and algebras. The classical example of this correspondence is between finite (non)deterministic automata, regular languages, free Kleene algebras (aka. regular expressions), and finite syntactic monoids. In the area of concurrency, such correspondences have been sought as well [7,9,13,23,25]. Several automata models have emerged from this as did the notion of concurrent Kleene algebras [15,16], which extend Kleene algebras with parallel computation and process replication (also called parallel closure). Concurrent Kleene algebras (CKA) correspond to several automata models [23,25].

Parallel to automata models for CKA, several operational models of true concurrency have been developed, such as Petri nets and higher-dimensional automata (HDA). These are models that can faithfully represent parallel computation without having to resort to sequentialisation [32]. HDA have received a lot of attention recently because of the geometric view on concurrency that they offer [10,11,12,19,29,30,32]. Fahrenberg et al. showed that there is a correspondence between finite HDA and Kleene algebras (KA) with parallel composition and that KA with process replication cannot be given semantics in terms of finite HDA [9, Lemma 12]. We show in this paper that process replication can also not be realised as neither locally finite HDA, in which every state can only reach finitely many other states [4,27,28], nor as finitely branching HDA with finitely initial states. Our approach is to prove that the category of HDA is locally finitely presentable, which allows us to define the language of HDA in terms of languages of finite HDA [8], prove that any HDA is locally finite HDA and that process replication cannot be realised in any finitary way over HDA.

But what are HDA in the first place? The idea is that they generalise labelled transition systems to allow for n actions to be active simultaneously by modelling transitions as n-cells in higher-dimensional cubes. For instance, fig. 1 shows a graphical representation of a HDA over an alphabet with actions $\{a, b, c, d\}$. The

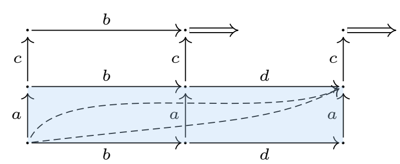


Figure 1. Event a may happen in parallel with b and d (filled squares), while c is in conflict with b and d (not filled); two parallel executions of a and b, and a and d are indicated by the dashed homotopic paths; cells with double arrows are accepting cells

dots indicate 0-cells, in which no action is active, solid arrows are 1-cells that are transitions with one active action, and the blue shaded areas are 2-cells with two active actions. Starting from the bottom left, first a and b may be active in parallel and any execution path through the shaded area is allowed. In the square above that, the action c and b have to be executed sequentially because the square is not filled. The HDA in fig. 1 accepts a run if one of the 0-cells with a double arrow is reached. For instance, the (sequential) path $a \to b \to c$ is accepted. More generally, HDA accept pomset languages [8]. In the case of fig. 1, the accepted language is given by the following set consisting of ten pomsets.

$$\left\{ (a \to b \to c), (a \to c \to b), (b \to a \to c), (a \to b \to d \to c), \begin{pmatrix} a & & \\ & \searrow & \\ & & c \\ & \nearrow & \\ b & & \end{pmatrix}, \right.$$

$$\left. (b \to a \to d \to c), (b \to d \to a \to c), \begin{pmatrix} & a & & \\ & & \searrow & \\ & & & c \\ & & \nearrow & \\ b \to d & & & \end{pmatrix}, \begin{pmatrix} a & & & \\ & \searrow & & \\ & & d \to c & \\ & \nearrow & & \\ b & & & \end{pmatrix}, \begin{pmatrix} & & a & & \\ & \nearrow & & \searrow & \\ b & & & & c \\ & \searrow & & \nearrow & \\ & & d & & \end{pmatrix} \right\}$$

The first six are purely sequential runs, while the last four use the concurrent capabilities of the HDA to run a, b, c and d in parallel. Pomset languages can be composed with the operations of concurrent Kleene algebras, and one may then ask which of these operations carry over to HDA and may result in a correspondence between (locally) finite HDA and rational pomset languages constructed from these operations.

Outline and Contributions We show in section 3.3 that the category of HDA is locally finitely presentable (lfp) and that finite HDA are exactly the compact (or finitely presentable) objects. This allows the reduction of arguments to finite HDA. In section 4.2, we show that languages of coproducts and filtered colimits of HDA are given directly by the languages of the HDA in the corresponding diagrams, and that this fails for general colimits. We also give in section 3.2 a novel characterisation of the tensor product of HDA, and then use this and the lfp property to show that the tensor product yields the parallel composition of languages. In section 5 we present two possible local finiteness conditions for HDA that are stable under process replication. We then show that both notions involve some infinite branching and we end with a proof that it is impossible to realise process replication without infinite branching. We begin with a recap of the theory of pomset languages in section 2 and of HDA in section 3.

Related Work The work of Lodaya and Weil [25] offers another automaton model for concurrency, called branching automata, as well as an algebraic perspective. Interestingly, their correspondence is restricted to languages of bounded width. Our result in section 5 could be extended to show that finitely branching HDA correspond to languages of bounded width, but we do not explore this further, as bounded width languages can be realised without process replication.

Ésik and Németh [7] prove a correspondence between rational languages of *series-parallel biposets*, which are essentially pomsets, and finite parenthesising automata. Such automata have two kinds of states and transition relations that can be thought of as 0- and 1-cells, and transitions among them (respectively 1- and 2-cells) and transitions up and down one dimension and that are guarded by parentheses. Thus, they make HDA more flexible in that they allow dimension change but also restrict the dimensions.

Jipsen and Moshier [18] reiterate on branching automata [25] but improve them by adding a bracketing condition akin to parenthesising automata [7].

Kappé et al. [21,22,23] have shown that finite well-nested pomset automata correspond to concurrent Kleene algebras and, what they call, series-parallel rational expressions. Pomset automata have two transition functions, one for sequential and one for parallel computation. The latter can branch out to finitely many parallel states and synchronise after each has completed their work. This allows them to implement process replication because the number of parallel processes can grow arbitrarily during execution, while the dimension of a cell in a HDA fixes the number of parallel processes. We will discuss this in section 6.

Finally, our work builds on the work by Fahrenberg et al. [9]. For the most part, we follow their work [9] in our definitions of HDA and languages, but also deviate in some choices, like the definition of the cube category and the tensor product of HDA. We have also followed them in giving up on event consistency [8], as the category of HDA would otherwise not be cocomplete [3].

2 Concurrent Words via Ipomsets

In this section, we recap the theory of interval ipomsets and their languages, sequential composition, parallel composition and parallel Kleene closure [8].

2.1 Ipomsets

Definition 1. *A* labelled iposet *P is a tuple* $(|P|, <_P, \dashrightarrow_P, S_P, T_P, \lambda_P)$ *where*

- $|P|$ *is a finite set,*
- $<_P$ *is a strict partial order on* $|P|$ *called* precedence order*,*
- $\dashrightarrow_P$ *is a strict partial order on* $|P|$*, called* event order*, that is linear on* $<_P$*-antichains,*
- $\lambda_P \colon |P| \to \Sigma$ *is a labelling map to an alphabet* Σ*,*
- $S_P \subseteq |P|$ *is a set of* $<_P$*-minimal elements called the* source set*, and*
- $T_P \subseteq |P|$ *is a set of* $<_P$*-maximal elements called the* target set*.*

Note that the condition that $\dashrightarrow_P$ is linear on $<_P$-antichains implies that $\dashrightarrow_P$ and $<_P$ together form a total order.

Definition 2. *We say that a labelled iposet P is* subsumed *by a labelled iposet Q, written* $P \sqsubseteq Q$*, if there exists a bijection* $f \colon |P| \to |Q|$ *with* $f(S_P) = S_Q$*,* $f(T_P) = T_Q$ *and such that for all* $x, y \in |P|$ *we have*

1. $f(x) <_Q f(y) \implies x <_P y$
2. $x \dashrightarrow_P y,\ x \not<_P y,\ y \not<_P x \implies f(x) \dashrightarrow_Q f(y)$
3. $\lambda_P(x) = \lambda_Q \circ f(x)$

The labelled iposets P and Q are isomorphic if f is an isomorphism for both orders. An ipomset *is an isomorphism class of labelled iposets.*

$P \sqsubseteq Q$ intuitively means that P is more ordered by the precedence order $<$ than Q, which means that P has less "concurrency". Isomorphisms between labelled iposets are unique, which means that any skeleton of the category of labelled iposets and subsumptions isomorphic to the quotient by isomorphisms.

Definition 3. *An ipomset P is an* interval ipomset *if there is a pair of functions* $b, e \colon |P| \to \mathbb{R}$ *to the real numbers, such that* $b(x) \leq e(x)$ *for all* $x \in |P|$ *and* $x <_P y \iff e(x) < b(y)$ *for all* $x, y \in |P|$*. The pair of functions* (b, e) *is called an* interval representation *of P. We let* `iiPom` *be the set of all interval ipomsets.*

The simplest example of an ipomset that is not interval is the ipomset P with $|P| = \{a, b, c, d\}$ with $a <_P b$ and $c <_P d$ but where a and b are incomparable with c and d. This is the ipomset variant of the $(2+2)$-poset. Given a set of interval ipomsets $A \subseteq$ `iiPom`, the down-closure of A is defined as usual by $A^{\downarrow} = \{P \in \texttt{iiPom} \mid \exists Q \in A.\, P \sqsubseteq Q\}$.

Definition 4. *A* language L *of interval ipomsets is a down-closed set of interval ipomsets, that is, if* $L^{\downarrow} \subseteq L$ *holds. We denote by* **Lang** *the thin category with languages as objects and subset inclusions as morphisms.*

2.2 Composition of ipomsets and languages

Definition 5. *We say that ipomsets* P *and* Q sequentially match *if there is a (necessarily unique) isomorphism* $f\colon (T_P, \dashrightarrow_P) \to (S_Q, \dashrightarrow_Q)$ *with* $\lambda_Q \circ f = \lambda_P$*. If* P *and* Q *match sequentially, then we define the* gluing composition *by*

$$P * Q = (|P * Q|, <_{P*Q}, \dashrightarrow_{P*Q}, S_P, T_Q, \lambda_{P*Q}),$$

where $(|P * Q|, \dashrightarrow_{P*Q})$ *is the pushout* $\operatorname{colim}\Big((|P|, \dashrightarrow_P) \hookleftarrow T_P \xrightarrow{f} (|Q|, \dashrightarrow_Q)\Big)$ *of posets of* f *along the inclusion. The precedence order* $<_{P*Q}$ *is the union of the images of* $<_P$*,* $<_Q$ *and* $(|P| \setminus T_P) \times (|Q| \setminus S_Q)$ *in* $|P * Q|$*. Finally, the labelling function* $\lambda_{P*Q}\colon |P * Q| \to \Sigma$ *is defined as the copairing* $[\lambda_P, \lambda_Q]$ *on the pushout using that* f *preserves labelling.*

If P and Q are interval ipomsets, then their gluing composition $P * Q$ is an interval ipomset as well [8, Lem. 41]. This uses that the map f, which attaches the interfaces, is an order isomorphism and that the event order is linear.

If the interfaces T_P and S_Q are empty, then $P * Q$ is the coproduct of $(|P|, \dashrightarrow_P)$ and $(|Q|, \dashrightarrow_Q)$, and at the same time the join of $(|P|, <_P)$ and $(|Q|, <_Q)$ considered as categories. This amounts to the serial pomset composition [9], which is the generalisation of concatenation of words to pomsets.

Definition 6. *The sequential composition of languages* L_1 *and* L_2 *is defined as*

$$L_1 * L_2 = \{P * Q \mid P \in L_1, Q \in L_2, \text{and } P \text{ and } Q \text{ match sequentially}\}^{\downarrow}$$

Definition 7. *We define the* parallel composition *of ipomsets* P *and* Q *by*

$$P \parallel Q = \big(|P| + |Q|, <_{P\parallel Q}, \dashrightarrow_{P\parallel Q}, S_{P\parallel Q}, T_{P\parallel Q}, \lambda_{P\parallel Q}\big)$$

Let $i_P : |P| \to |P| + |Q|$ *and* $i_Q : |Q| \to |P| + |Q|$ *be the canonical injection maps. Using these injection maps we define* $<_{P\parallel Q} = i_P(<_P) \cup i_Q(<_Q)$*,* $S_{P\parallel Q} = i_P(S_P) \cup i_Q(S_Q)$*,* $T_{P\parallel Q} = i_P(T_P) \cup i_Q(T_Q)$ *and* $\lambda_{P\parallel Q} = [\lambda_P, \lambda_P]$*. Then* $\dashrightarrow_{P\parallel Q}$ *is defined as the ordered sum of the event orders, in other words,* i_P *preserves the order* $\dashrightarrow_P$ *as* $\dashrightarrow_{P\parallel Q}$ *and* i_Q *preserves* $\dashrightarrow_Q$ *as* $\dashrightarrow_{P\parallel Q}$ *and for all* $x \in |P|$*,* $y \in |Q|$ *we have* $i_P(x) \dashrightarrow_{P\parallel Q} i_Q(y)$*.*

Differently said, the event order $\dashrightarrow_{P\parallel Q}$ on the parallel composition $P \parallel Q$ is defined as the join of $(|P|, \dashrightarrow_P)$ and $(|Q|, \dashrightarrow_Q)$ thought of as categories.

Definition 8. *The parallel composition of languages* L_1 *and* L_2 *is defined as*

$$L_1 \parallel L_2 = \{P \parallel Q \mid P \in L_1,\ Q \in L_2\}^{\downarrow}$$

and the parallel Kleene closure *of a language* L *as*

$$L^{(*)} = \bigcup_{n\in\mathbb{N}} L^{\parallel n} \quad \text{where} \quad L^{\parallel 0} = \{\varepsilon\} \text{ and } L^{\parallel(n+1)} = L \parallel \big(L^{\parallel n}\big)$$

Down-closure is needed in Definitions 6 and 8, since sequential or parallel compositions of down-closed languages may not result in a down-closed language. However, we can form unions of languages.

Lemma 1. *Languages are closed under arbitrary unions.*

We conclude this section by showing that the parallel composition of languages respects small colimits.

Lemma 2. *For small diagrams $M : D \to \mathbf{Lang}$ and $N : E \to \mathbf{Lang}$ of languages we have*

$$\bigcup_{(d,e)\in D\times E} M_d \parallel N_e = \Big(\bigcup\nolimits_{d\in D} M_d\Big) \Big\| \Big(\bigcup\nolimits_{e\in E} N_e\Big)$$

3 Higher-Dimensional Automata

In this section we first recall the definition of HDA, then discuss the monoidal structure of HDA to model parallel computation and finally show in section 3.3 that the category of HDA is locally finitely presented by finite HDA.

3.1 The Category of HDA

Higher-dimensional automata are modelled as labelled precubical sets, which in turn are presheaves over a category of basic hypercubes. Such cubes can be represented as ordered sets, where the size of the set corresponds to the dimension of the cube, and the morphism of the ordered sets determine how the faces of $(n+1)$-cells in a precubical set match with n-dimensional faces. We fix from now on an alphabet Σ in which HDA are labelled.

Definition 9. *A labelled linearly ordered set or lo-set $(U, \dashrightarrow, \lambda)$ is a finite set U with a strict linear order $\dashrightarrow$ and a labelling map $\lambda\colon U \to \Sigma$. We write ε for the unique empty lo-set. A lo-map is a map between lo-sets that preserves the order and the labelling. Lo-sets and -maps form a category $\ell\mathbf{SLO}$.*

The category $\ell\mathbf{SLO}$ is monoidal with $U \star V$ being the join of U and V considered as thin categories and the monoidal unit being the empty set. Explicitly, the underlying set of $U \star V$ is the coproduct $U + V$, the order is given by $x \dashrightarrow_{U\star V} y$ iff $x \dashrightarrow_U y$, $x \dashrightarrow_V y$, or $x \in U$ and $y \in V$. The labelling $\lambda_{U\star V}$ is given by the copairing $[\lambda_U, \lambda_V]\colon U + V \to \Sigma$.

Note that lo-maps are necessarily injective, which means that morphisms $f\colon U \to V$ in $\ell\mathbf{SLO}$ are equivalently defined by their image $f(U)$ or their complement $V \setminus f(U)$. Moreover, f is an isomorphism iff f is surjective, i.e. if $V \setminus f(U) = \emptyset$. Since isomorphisms in $\ell\mathbf{SLO}$ are unique, we can safely identify it with a skeleton that has as objects pairs $(\mathbf{n}, w)$ where $n \in \mathbb{N}$, $\mathbf{n}$ is the finite ordinal $\{0 < \cdots < n-1\}$ with n elements and $w \in \Sigma^n$ is a word of length n.

Definition 10. *A* coface map $d\colon U \to V$ *between lo-sets* U *and* V *is a triple* (f, A, B)*, where* $f\colon U \to V$ *is a lo-map and* $\{A, B\}$ *is a partition of the complement image of* f*, that is,* $V \setminus f(U) = A \cup B$ *and* $A \cap B = \emptyset$*. We write* $d(x)$ *for the application of the underlying map* f *to* x *to simplify notation. For* $A, B \subseteq U$ *that are disjoint, we denote by* $d_{A,B}\colon U \setminus (A \cup B) \to U$ *the coface map* (i, A, B)*, where* $i\colon U \setminus (A \cup B) \to U$ *is the inclusion.*

The monoidal structure on $\ell\mathbf{SLO}$ induces a monoidal structure on the category of lo-sets and coface maps, which is the *full precube category* $\boxdot$.

Lemma 3. *The lo-sets and coface maps form a monoidal category* $(\boxdot, \oplus, I)$*.*

Since isomorphisms in $\ell\mathbf{SLO}$ are unique, they are in $\boxdot$ as well and we can use the same skeleton as for $\ell\mathbf{SLO}$ only with the morphisms of $\boxdot$. We denote this small skeleton by $\square$.

Definition 11. *A* precubical set *is a presheaf* $X\colon \square^{\mathrm{op}} \to \mathbf{Set}$ *and a morphism of precubical sets is a natural transformation. They form a category* $\mathbf{PSh}(\square)$*. We write* よ *for the Yoneda embedding* $\square \to \mathbf{PSh}(\square)$ *with* $よ_U = \square(-, U)$*.*

We refer to the elements of $X[U]$ as *cells* and to the cardinality of U as the *dimension* of those cells. If for some U of cardinality n the set $X[U]$ is inhabited and for all V with cardinality greater than n the sets $X[U]$ are empty, then we say that X has finite dimension n. A precubical set X is finite if it has finite dimension and if for all $U \in \square$ the set $X[U]$ is finite.

To ease notation, we denote the face map $X[d_{A,B}]\colon X[U] \to X[U \setminus (A \cup B)]$, induced by a coface map $d_{A,B}\colon U \setminus (A \cup B) \to U$, by $\delta_{A,B}$. The face maps $\delta_{A,\emptyset}$ and $\delta_{\emptyset,B}$ will be suggestively abbreviated to δ^0_A and δ^1_B.

Definition 12. *A* higher-dimensional automaton (HDA) *is a tuple* $\left(X, X_\perp, X^\top\right)$ *where* X *is a precubical set,* $X_\perp$ *and* $X^\top$ *are families of sets indexed by lo-sets of starting and accepting cells with* $(X_\perp)_U \subseteq X_U$ *and* $(X^\top)_U \subseteq X_U$*. A HDA map* $f\colon \left(X, X_\perp, X^\top\right) \to \left(Y, Y_\perp, Y^\top\right)$ *is a precubical map* $f\colon X \to Y$ *that preserves the starting and accepting cells, that is,* $f(X_\perp) \subseteq Y_\perp$ *and* $f\left(X^\top\right) \subseteq Y^\top$*. We denote by* $\mathbf{HDA}$ *the category of higher-dimensional automata and their maps.*

We usually leave out the index on $X_\perp$ and $X^\top$ for better readability.

Lemma 4. *The forgetful functor* $\mathcal{F} : \mathbf{HDA} \to \mathbf{PSh}(\square)$ *has left and right adjoints* N *and* T *given, respectively, by* $NX = (X, \emptyset, \emptyset)$ *and* $TX = (X, |X|, |X|)$ *where* $|X|$ *is the family obtained by forgetting the action of* X *on morphisms. Thus, the left adjoint* N *stipulates no starting or accepting cells, while* T *considers all cells as starting and accepting.*

3.2 Monoidal Structure on HDA

Our main interest in this paper is to realise (repeated) parallel composition of languages as HDA. In this section we briefly discuss how HDA can be synchronised in parallel via a monoidal product on **HDA**.

Definition 13. *The tensor product of HDA is the Day convolution [6,17,26], which is given for HDA X and Y on the precubical sets by the following coend.*

$$X \otimes Y = \int^{V,W} \square(-, V \oplus W) \times X[V] \times Y[W]$$

The starting cells $(X \otimes Y)_\perp$ are given as the image of all inclusions

$$(X_\perp \cap X[V]) \times (Y_\perp \cap Y[W]) \longrightarrow \square(V \oplus W, V \oplus W) \times X[V] \times Y[W] \longrightarrow X \otimes Y$$

and analogously for accepting cells $(X \otimes Y)^\top$. A diagram chase shows that $\otimes$ is well-defined on HDA morphisms. For any $U \in \square$, we can make $よ_U$ an HDA by taking all cells to be starting and accepting. The monoidal unit I is the Yoneda embedding $よ_\varepsilon$ of the empty lo-set with the only 0-cell being starting and accepting.

By this definition, the Yoneda embedding is a strong monoidal functor and $\otimes$ preserves colimits [17]. Moreover, $\mathcal{F}$ is clearly a strict monoidal functor. Usually, the tensor product of (pre)cubical sets is defined as a coproduct [5,9,14,20]. Indeed, one can prove that $(X \otimes Y)(U) \cong \coprod_{U=V\oplus W} X[V] \times Y[W]$.

3.3 Filtered Colimits and Compact HDA

Compact (or finitely presentable) objects in a category can be thought of as the analogue of finite sets, relative to what morphisms in that category perceive as finite. For instance, compact objects in the category $\mathbf{Vec}_\mathbb{R}$ of $\mathbb{R}$-vector spaces are vector spaces with finite dimension. In **Set** and $\mathbf{Vec}_\mathbb{R}$, arguments can be reduced to arguments about compact objects because *all* objects in those categories are given as nice colimits of a set of chosen compact objects. For instance, each set U is given as a colimit of finite sets by taking the unit of the finite subsets of U. This process is given by so-called filtered colimits. The advantage of breaking down objects to filtered colimits of compact objects is that constructions on objects can be carried out on a set of compact objects instead. Categories that admit these kind of reduction are called locally finitely presentable (lfp).

In what follows, we recall the definition of lfp categories, show that the category of HDA is lfp and that the compact objects are precisely the finite HDA.

We first provide the basics of lfp categories [1,31]. A category $\mathcal{C}$ is called *essentially small* if it is equivalent to a small category. We call a category D *filtered* if any finite diagram in D has a cocone, or equivalently if (1) D is inhabited, (2) for any two objects $c, d \in D$ there exists an object $e \in D$ and two morphisms $c \to e \leftarrow d$, and (3) for any two morphisms $f, g \colon c \to d$ there exist an object $e \in D$ and a morphism $h \colon d \to e$ with $h \circ f = h \circ g$. A *filtered colimit* in a category $\mathcal{C}$ is a colimit of a diagram $F \colon D \to \mathcal{C}$ where D is filtered. We say that an object $X \in \mathcal{C}$ is *compact* if the hom-functor $\mathcal{C}(X, -) \colon \mathcal{C} \to \mathbf{Set}$ preserves filtered colimits. Finally, the category $\mathcal{C}$ is called *locally finitely presentable (lfp)* if it is cocomplete, the subcategory $\mathcal{C}_\mathrm{c}$ of compact objects is essentially small, and every object in $\mathcal{C}$ is isomorphic to a filtered colimit of compact objects.

Many calculations are simplified by the fact that the category $\mathcal{C}_c$ is closed under finite colimits [1, Prop. 1.3]. One of the important examples of a lfp category is the functor category of precubical sets $\mathbf{PSh}(\square)$ [1, Ex. 1.12] and that the hom-functor $よ_U$ is compact in $\mathbf{PSh}(\square)$ for all $U \in \square$. Similarly, the category **HDA** is also lfp, as we show now.

Lemma 5. *The forgetful functor $\mathcal{F}\colon \mathbf{HDA} \to \mathbf{PSh}(\square)$ creates colimits [31, Sec. 3.3] and the category of HDA is thus cocomplete.*

Theorem 1. *A HDA is compact if and only if it is finite.*

Let $I\colon \mathbf{HDA}_c \to \mathbf{HDA}$ be the inclusion functor of the full subcategory of compact HDA in **HDA**. For a HDA X, we denote by $I \downarrow X$ the comma category that has as objects morphisms $Y \to X$ from a compact HDA Y into X, and morphisms are the evident commutative triangles. The comma category $I \downarrow X$ is essentially small and closed under finite colimits, thus it is a filtered category. We write $U_X : I \downarrow X \to \mathbf{HDA}_c$ for the domain projection functor.

Lemma 6. *Every HDA X can be canonically expressed as the filtered colimit of finite HDA, that is, we have $X \cong \operatorname{colim} U_X$.*

Theorem 2. *The category of HDA is locally finitely presentable.*

An alternative proof that **HDA** is lfp is to show that it is equivalent to a reflective subcategory $\mathcal{H}$ of a presheaf category that is closed under filtered colimits. This implies that $\mathcal{H}$ and **HDA** are lfp [1, Sec. 1C]. We thank an anonymous referee of a previous version of this paper for the suggestion.

4 Languages of Higher-Dimensional Automata

Computations as modelled by HDA can be expressed as higher-dimensional paths running through the HDA from a starting cell to an accepting cell. Each of these accepting paths corresponds to an interval ipomset, which allows us to define the languages of HDA as the set of interval ipomsets it accepts. We expand here on previous work [9] by also including infinite HDA and by showing that HDA languages preserve coproducts and filtered colimits.

4.1 Paths and languages

Let us start by defining paths and their labelling.

Definition 14. *A path (of length n) in a precubical set or HDA X is a (finite) sequence*

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

where $x_k \in X[U_k]$ are cells for $U_k \in \square$ and for all $1 \le k \le n$ we have an

– *up-step: $\varphi_k = d_{A,\emptyset} \in \square(U_{k-1}, U_k)$, $x_{k-1} = \delta^0_A(x_k)$ and $A = U_k \backslash U_{k-1}$, or*

– *down-step:* $\varphi_k = d_{\emptyset,B} \in \square(U_k, U_{k-1})$, $\delta^1_B(x_{k-1}) = x_k$ *and* $B = U_{k-1} \backslash U_k$.

The elements x_k define cells while the φ_k define how these cells are connected. Since for a path we cannot have $\delta^0_A(x_{k-1}) = x_k$ or $x_{k-1} = \delta^1_B(x_k)$ it can only move along the direction of the arrows. Two paths where the first ends at the cell the other starts in can be composed in the following intuitive manner.

Definition 15. *Let* $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$ *and* $\beta = (y_0, \psi_1, y_1, ..., \psi_m, y_m)$ *be two paths in a precubical set or HDA* X *with* $x_n = y_0$. *Then we define their concatenation* $\alpha * \beta$ *as the following path in* X.

$$\alpha * \beta = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n, \psi_1, y_1, ..., \psi_m, y_m)$$

Every path $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$ can therefore be broken down into paths of length 1, called steps. We can denote a step $(x_{k-1}, \varphi_k, x_k)$ with $x_{k-1} \nearrow^A x_k$ if $\varphi_k = d_{A,\emptyset}$ (an *up step*) or with $x_{k-1} \searrow_B x_k$ if $\varphi_k = d_{\emptyset,B}$ (a *down step*). We get the unique representation $(x_0, \varphi_1, x_1) * (x_1, \varphi_2, x_2) * ... * (x_{n-1}, \varphi_n, x_n)$ for the path α. Using this we define the labelling of paths recursively.

Definition 16. *Let* X *be a precubical set or HDA. Let* α *be a path in* X, *let* U *and* V *be objects in* $\square$ *and let* $x \in X[U]$, $y \in X[V]$. *Then the labelling* $\mathtt{ev}(\alpha)$ *of* α *is the ipomset that is computed as follows:*

– *If* $\alpha = (x)$ *is a path of length 0 then its label is* $\mathtt{ev}(\alpha) = (U, \emptyset, \dashrightarrow_U, U, U, \lambda_U)$.
– *If* $\alpha = (x, \varphi, y)$ *is a path with* $x \nearrow^A y$ *then its label is*

$$\mathtt{ev}(\alpha) = (V, \emptyset, \dashrightarrow_V, V \backslash A, V, \lambda_V)$$

– *If* $\alpha = (x, \varphi, y)$ *is a path with* $x \searrow_B y$ *then its label is*

$$\mathtt{ev}(\alpha) = (U, \emptyset, \dashrightarrow_U, U, U \backslash B, \lambda_U)$$

– *If* $\alpha = \beta_1 * \beta_2 * ... * \beta_n$ *the concatenation of steps* $\beta_1, \beta_2, ..., \beta_n$ *then its label is the gluing composition of ipomsets* $\mathtt{ev}(\alpha) = \mathtt{ev}(\beta_1) * \mathtt{ev}(\beta_2) * ... * \mathtt{ev}(\beta_n)$.

The labels of paths of length 0 or 1 are trivially interval ipomsets, since the relation $<$ is empty. Since the labelling of paths of length greater than 1 is defined as the gluing of the labels of its steps it follows that they are interval ipomsets as well.

For a precubical set or HDA X we define P_X as the set of paths in X. For a path $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$ we call $s(\alpha) = x_0$ the source and $t(\alpha) = x_n$ the target of the path. We can now define the languages of HDA.

Definition 17. *The language of an HDA* X *is the set of interval ipomsets*

$$L(X) = \left\{ \mathtt{ev}(\alpha) \mid \alpha \in P_X,\ s(\alpha) \in X_\perp,\ t(\alpha) \in X^\top \right\}$$

We refer to a path α with $s(\alpha) \in X_\perp$ and $t(\alpha) \in X^\top$ as an accepting path. In lemma 11 we will prove that for each HDA X the language $L(X)$ of X is a down-closed interval ipomset language as defined in definition 4. Let X

and Y be precubical sets with the precubical map $f : X \to Y$. For each path $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$ in X with $x_k \in X[U_k]$ we define
$f(\alpha) = (f_{U_0}(x_0), \varphi_1, f_{U_1}(x_1), ..., \varphi_n, f_{U_n}(x_n))$ which by definition of the precubical maps is a path in Y. With this we get two lemmas regarding the way precubical maps and HDA maps preserve paths and languages.

Lemma 7. *Let X and Y be precubical sets and let $f : X \to Y$ be a precubical map. Suppose that we have $\alpha, \beta \in P_X$ with $s(\alpha) = t(\beta)$. Then we have $\mathtt{ev}(\alpha * \beta) = \mathtt{ev}(\alpha) * \mathtt{ev}(\beta)$ and $\mathtt{ev}(f(\alpha)) = \mathtt{ev}(\alpha)$.*

Lemma 8. *Let X and Y be HDA and let $f : X \to Y$ be a HDA map. Then we have $L(X) \subseteq L(Y)$. If f is an isomorphism then we have $L(X) = L(Y)$.*

4.2 Composition of HDA and their languages

We want to know the relation between the languages of diagrams of HDA and the languages of their colimits. We start with a theorem that is relevant for all colimits and cocones.

Lemma 9. *Let (X, ϕ) be a cocone of the small diagram $F : D \to$ **HDA**. Then we have $\bigcup_{d \in D} L(F(d)) \subseteq L(X)$.*

We get equality in the case that (X, ϕ) is a coproduct or a filtered colimit, as we will prove with the next two theorems.

Lemma 10. *Let $F \colon D \to$ **HDA** be a small discrete diagram of HDA with coproduct $(X, \phi) = \operatorname{colim} F$. Then we have $\bigcup_{d \in D} L(F(d)) = L(X)$.*

Theorem 3. *Let $F \colon D \to$ **HDA** be a small filtered diagram of HDA with filtered colimit $(X, \phi) = \operatorname{colim} F$. Then we have $\bigcup_{d \in D} L(F(d)) = L(X)$.*

Proof. Suppose that $P \in L(X)$. Then there exists a path α in X with $s(\alpha) \in X_\perp$ and $t(\alpha) \in X^\top$ such that $\mathtt{ev}(\alpha) = P$. Let $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$. Lemma 17 gives us that there exists an index $d \in D$ and a path $\alpha' = (y_0, \varphi_1, y_1, ..., \varphi_n, y_n)$ such that $\phi_d(\alpha') = \alpha$ (a path in this case can be seen as a finite set S). Because of lemma 15 we can assume that this path is accepting. This gives us that $\mathtt{ev}(\alpha') = P \in \bigcup_{d \in D} L(F(d))$, which proves with lemma 9 the statement. □

The theorem above together with lemma 6 shows that all infinite HDA can be expressed using finite HDA respecting the corresponding languages. This powerful tool allows us to prove statements about the languages of HDA in a simple way by using the filtered colimits of finite HDA demonstrated by the following theorem.

Lemma 11. *The languages of HDA are down-closed interval ipomset languages.*

Since **Lang** is the category with as objects down-closed interval ipomset languages and as morphisms the subset inclusion maps, the theorem above and lemma 8 allow us to see L as a functor $L :$ **HDA** $\to$ **Lang**. Since the colimit of a diagram of languages is the union, lemma 10 and theorem 3 give us that L preserves coproducts and filtered colimits. However, it does not preserve all colimits as we show with the next theorem.

Proposition 1. *There is a small diagram $F\colon D \to \mathbf{HDA}$, whose colimit accepts more than the HDA in the diagram together:* $\bigcup_{d\in D} L\left(F(d)\right) \subsetneq L(\operatorname{colim} F)$.

Proof. We use for D the category of shape $1 \leftarrow 2 \rightarrow 3$. Consider the following pushout of HDA, which is a colimit over a diagram of shape D.

$$\begin{array}{ccc}
(\circ) & \xrightarrow{\quad i_1 \quad} & (\Rightarrow \bullet \xrightarrow{a} \circ) \\
{\scriptstyle i_2}\big\downarrow & \ulcorner & \big\downarrow \\
(\circ \xrightarrow{c} \bullet \Rightarrow) & \longrightarrow & (\Rightarrow \bullet \xrightarrow{a} \bullet \xrightarrow{c} \bullet \Rightarrow)
\end{array}$$

The inclusions i_k map $\circ$ to $\circ$ and the double arrows indicate starting and accepting cells. Note that the languages of the HDA at the corners are all empty, except of the HDA at the bottom right corner, which accepts the word $(a \to c)$. Thus the pushout of these HDA with empty languages has a non-empty language. □

Finally, we prove that the language of the tensor product of two HDA is the same as the parallel composition of their two individual languages.

Theorem 4. *The functor L is a strict monoidal* $(\mathbf{HDA}, \otimes, I) \to (\mathbf{Lang}, \parallel, \{\varepsilon\})$.

Proof. Let X and Y be HDA. We have to show that $L\left(X \otimes Y\right) = L\left(X\right) \parallel L\left(Y\right)$. Lemma 6 provides use with filtered diagrams $F\colon D \to \mathbf{HDA}$ and $G\colon E \to \mathbf{HDA}$ of finite HDA with X and Y being their respective filtered colimits. This allows us to generalise [9, Prop. 19], where $L(X \otimes Y) = L(X) \parallel L(Y)$ is proved for finite HDA, to arbitrary HDA.

$$\begin{aligned}
L\left(X \otimes Y\right) &= L\Big(\operatorname*{colim}_{(d,e)\in D\times E} F(d) \otimes G(e)\Big) && \text{tensor product preserves colimits} \\
&= \bigcup\nolimits_{(d,e)\in D\times E} L\left(F(d) \otimes G(e)\right) && \text{by theorem 3} \\
&= \bigcup\nolimits_{(d,e)\in D\times E} L\left(F(d)\right) \parallel L\left(G(e)\right) && \text{[9, Prop. 19] for finite HDA} \\
&= \bigcup\nolimits_{d\in D} L\left(F(d)\right) \parallel \bigcup\nolimits_{e\in E} L\left(G(e)\right) && \text{by lemma 2} \\
&= L(X) \parallel L(Y) && \text{by theorem 3}
\end{aligned}$$

This shows that even for arbitrary HDA the parallel composition of their languages is given by tensoring the HDA. That $L(I) = \{\varepsilon\}$ is obvious. □

5 Process Replication as Rational HDA

In this section, we seek to complete the correspondence between concurrent Kleene algebras and HDA, which requires us to identify a notion of *rational HDA* that can capture finitary behaviour. This has almost been accomplished [9] but the parallel closure could not be realised as finite HDA. For regular languages, linear weighted languages and various other languages without true concurrency, the correspondence between languages and automata has been studied from a coalgebraic perspective [4,27,28]. We make in section 5.1 a first attempt, where we follow these ideas by studying locally compact HDA and by showing how to realise the parallel closure as locally compact HDA. However, we will see that this model is too powerful and will restrict to finitely branching HDA in section 5.2. These can realise the parallel Kleene star as well, but will require an infinite choice at the start. Thus, none of these choices is satisfactory to act as rational HDA and we show that it is impossible to realise the parallel closure as finitely branching HDA with finitely many starting cells.

5.1 Locally Compact HDA

Let us first define what we mean by locally compact HDA. This follows work on rational coalgebraic behaviour [28,27] and can be seen as axiomatisation of the factorisation property that filtered colimits enjoy in lfp categories.

Definition 18. *A HDA $\left(X, X_\perp, X^\top\right)$ is* locally compact *if the forgetful functor $\mathcal{F}\colon \mathbf{HDA}_c \downarrow X \to \mathbf{PSh}(\square)_c \downarrow X$ is cofinal. Explicitly, this means [1, 0.11] that 1) for all compact precubical set P and $f\colon P \to X$ there is a factorisation of f into $P \xrightarrow{f'} Y \xrightarrow{h} X$, where $h\colon \left(Y, Y_\perp, Y^\top\right) \to \left(X, X_\perp, X^\top\right)$ is a HDA morphism and $\left(Y, Y_\perp, Y^\top\right) \in \mathbf{HDA}_c$; and 2) for all $\left(Y', Y'_\perp, Y'^\top\right) \in \mathbf{HDA}_c$, $h'\colon \left(Y', Y'_\perp, Y'^\top\right) \to \left(X, X_\perp, X^\top\right)$ and $f''\colon P \to Y'$ with $h' \circ f'' = f$, there exists $\left(R, R_\perp, R^\top\right) \in \mathbf{HDA}_c$ and an HDA morphisms $e\colon \left(Y', Y'_\perp, Y'^\top\right) \to \left(R, R_\perp, R^\top\right)$ and $e'\colon \left(Y, Y_\perp, Y^\top\right) \to \left(R, R_\perp, R^\top\right)$ such that $e' \circ f' = e \circ f''$.*

The following lemma shows that the second condition is redundant, which follows from **HDA** being an lfp category.

Lemma 12. *An HDA $\left(X, X_\perp, X^\top\right)$ is locally compact if and only if all presheaf morphisms $f\colon P \to X$ factor as $P \xrightarrow{f'} Y \xrightarrow{h} X$ into a presheaf morphism f' and a HDA morphism $h\colon \left(Y, Y_\perp, Y^\top\right) \to \left(X, X_\perp, X^\top\right)$ from $\left(Y, Y_\perp, Y^\top\right) \in \mathbf{HDA}_c$.*

Since morphisms into filtered colimits factor essentially uniquely through the colimit inclusion, HDA given by a filtered colimit of compact HDA are locally compact. The other way around this is also true. Combined with lemma 6 this gives us the following result.

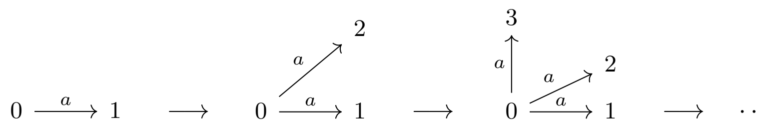
Theorem 5. *All HDA are locally compact.*

Proof. For one direction, we use that if $D \to \mathbf{HDA}_c$ is a filtered diagram, then $\operatorname{colim}(D \to \mathbf{HDA}_c \to \mathbf{HDA})$ is locally compact because filtered colimits in lfp categories factor essentially uniquely through colimit inclusions.

For the other direction, we use that every $x \in X[U]$ generates a compact sub-precubical set $\langle x \rangle \hookrightarrow X$ that contains x and all its boundary cells. This inclusion factors essentially uniquely into an inclusion of a compact HDA, as X is locally compact. This gives us inclusions HDA into $\operatorname{colim} U_X$ for every U and $x \in X[U]$. It is easy to see that these inclusions jointly set up an isomorphism. ◻

This shows that local compactness is no restriction in the case of HDA, contrary to other computational models. Let us, nevertheless, apply the lessons of local compactness to get closer to an HDA that models process replication in a reasonably finitary way. Before that, let us warm up and construct a HDA as a filtered colimit with infinite branching.

Example 1. Let $F\colon \mathcal{D} \to \mathbf{HDA}_c$ be the diagram given by



This is a chain and thus filtered, and its colimit a 1-dimensional HDA with infinitely many branches coming out of 0. Nevertheless, since each HDA in the chain is compact, $\operatorname{colim} F$ is locally compact.

Example 2. Similarly to example 1, we can also branch with higher dimensions and thus realise process replication as filtered colimit of compact HDA. For the purpose of this example it is simpler to ignore starting cells, but it is easy to see that tensor product and colimits are not affected by this.

Let A be the HDA with one 1-cell labelled with a and the endpoint of this 1-cell taken as accepting. This is illustrated in fig. 2 on the left, where the double arrows mark accepting cells. The maps $d_n\colon A_n \to A_{n+1}$ in fig. 2, where $A_1 = A$,

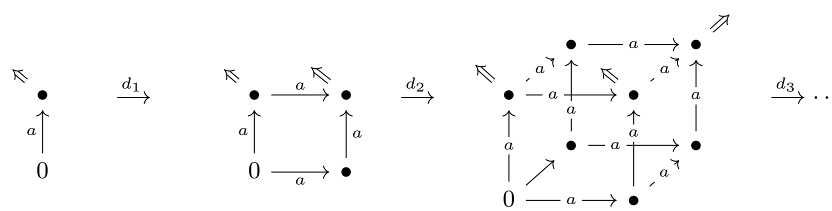
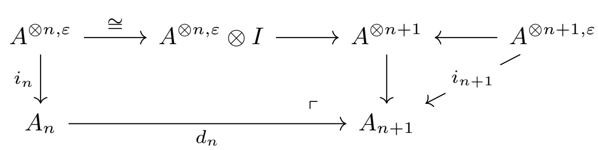


Figure 2. Chain of HDA to construct process replication of the HDA A on the left. The starting cell named 0 serves as orientation as to how d_n embeds the cells matching with the accepting cells.

are constructed as in the following pushout diagram. In this diagram, we denote by $A^{\otimes n}$ the n-fold tensor product of A with itself, where $A^{\otimes 0} = I$. For an HDA X, we write X^{ε} for the HDA that has the same underlying precubical set but no starting and accepting states.



Intuitively, the HDA A_{n+1} is given by extending A_n to a full $n+1$-dimensional cube, where A_n is included via d_n as the “front face”. In fig. 2, this inclusion is indicated by the vertex 0, which is identified via d_n. The indicated maps d_n form a chain of compact HDA and thus a filtered diagram $F : \mathcal{D} \to \mathbf{HDA}_c$. By taking the colimit of F and declaring the cell marked 0 as starting cell, we obtain an HDA that accepts $L(A)^{(*)}$, the parallel Kleene closure of the language of A. That this is the case follows directly from theorem 4 and theorem 3. Since each HDA in the chain is compact, $\operatorname{colim} F$ is locally compact, but this colimit is a HDA with infinitely many branches coming out of 0.

5.2 Finitely Branching HDA

The HDA that we constructed in example 2 has the pleasant property that during execution many a-processes can be spawned, as one would expect from a process replication operator that occurs in process algebra. However, the HDA in example 2 has infinitely many cells branching out of any cell. This makes it impossible to realise this HDA on a physical machine and motivates another possible definition of what one may consider rational HDAs.

Definition 19. *A HDA X is* finitely branching *if for all lo-sets $U \cup \{a\}$ and all $x \in X_U$ the set $\{y \in X_{U \cup \{a\}} \mid \delta_{A,B}(y) = x\}$ is finite. We denote by $\mathbf{HDA}_{\mathrm{fb}}$ the full subcategory of* **HDA** *that consists of finitely branching HDA.*

Clearly, finitely branching HDA are not closed under filtered colimits, as example 2 shows. However, they are closed under coproducts.

Lemma 13. *Let $F : \mathcal{D} \to \mathbf{HDA}_{\mathrm{fb}}$ a diagram on a small discrete category D. Then the colimit (coproduct)* $\operatorname{colim} F$ *exists in* $\mathbf{HDA}_{\mathrm{fb}}$.

The parallel Kleene star of a finitely branching HDA X, also known as process replication, can be realised as finitely branching HDA. We write $X^{\otimes n}$ for the n-fold tensor product of X with itself, where $X^{\otimes 0} = I$, and define the parallel replication of X to be $!X = \coprod_{n \in \mathbb{N}} X^{\otimes n}$.

Theorem 6. *The HDA $!X$ is finitely branching and we have $L(!X) = L(X)^{(*)}$.*

Proof. By lemma 10 and theorem 4 we have

$$L(!X) = L\Big(\coprod\nolimits_{n \in \mathbb{N}} X^{\otimes n}\Big) = \bigcup\nolimits_{n \in \mathbb{N}} L\big(X^{\otimes n}\big) = \bigcup\nolimits_{n \in \mathbb{N}} L(X)^{\| n} = L(X)^{(*)}$$

That $!X$ is finitely branching is given by lemma 13. □

The caveat of this theorem, and the definition of finitely branching in general, is that we do not make any restrictions on the number of starting cells. In fact, $!X$ will have infinitely many starting cells, if X has at least one.

Example 3. Let A again be the HDA as in example 2. The HDA $!A$ looks as in fig. 3. Notice that it consists of little finite islands, each with a starting cell. The HDA has to make at the beginning of an execution a choice on the number of parallel executions of the action a. This means that this HDA is not realisable, as such a guess requires knowledge about how many parallel processes will be needed. For instance, a web server would need to know *when it is started* how many clients will connect during its life time. This is clearly impossible.

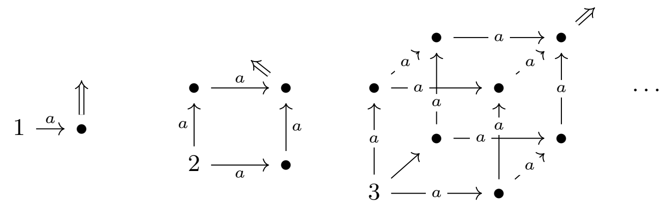


Figure 3. Finitely branching HDA for replication of A constructed as coproduct, where the cells $1, 2, 3, \ldots$ are starting cells and double arrows mark accepting cells

The examples 2 and 3 show that either way of realising process replication, as locally compact HDA or as finitely branching HDA, leads to operational problems. In fact, it is not possible to realise process replication as finitely branching HDA with finite starting cells.

Theorem 7. *There is no HDA $X \in \mathbf{HDA}_{\text{fb}}$ with finitely many initial states, such that X would realise the parallel Kleene star of $L(A) = \{(a)\}$, where A is the HDA with only one a-transition, as in example 2.*

Proof. Suppose there is an HDA $X \in \mathbf{HDA}_{\text{fb}}$ with finite initial states, such that $L(X) = L(A)^{(*)} = \{(a)\}^{(*)}$. We partition $L(X)$ into languages L_x for $x \in X_\perp$. Since $X_\perp$ is finite, some L_x must be infinite. Thus for every $\underbrace{(a) \parallel \cdots \parallel (a)}_{n} \in L_x$ there must be an n-cell of which x is a boundary. But then X has infinitely many branches at x, and thus X cannot exist with the proclaimed properties. □

Since the identity language has infinite width [9, Example 4], it cannot be represented by a finite HDA. One can provide a finitely branching HDA that accepts the identity language, but again with infinitely many starting cells. Thus, even this simple language does not fit into any reasonable restriction of HDA.

6 Conclusion

What does this leave us with? The problem is that HDA combine state space and transitions into one object, a precubical set. Intuitively, this prevents us from having transitions and cycles among cells of higher dimension. More technically, the locally compact HDA allow infinite branching, while finite branching limits the number of active parallel events to be finite. This can be compared to the coalgebras for the finite powerset functor, also known as finitely branching transition systems. Here, locally compact transition systems may only have finite branching and thus realise locally the behaviour of finite transition systems, as one would expect. Therefore, one is led to the conclusion that HDA as a computational model are unsuited to model process replication and another model for true concurrency has to be sought. This is not say that topological or geometrical models, like HDA, are inherently flawed but rather that they have to be expanded to allow for the dynamic spawning of processes, in contrast to the static nature of HDA.

References

1. Adámek, J., Rosický, J.: Locally Presentable and Accessible Categories. London Mathematical Society Lecture Note Series, Cambridge University Press, Cambridge (1994). https://doi.org/10.1017/CBO9780511600579
2. Awodey, S.: Category Theory. Oxford Logic Guides, Ebsco Publishing (2006), https://books.google.nl/books?id=IK_sIDI2TCwC
3. Baronner, T.: Finite Accessibility of Higher-Dimensional Automata and Unbounded Parallelism of Their Languages. Bachelor's Thesis, Leiden University (Dec 2022), https://theses.liacs.nl/2532
4. Bonsangue, M.M., Milius, S., Silva, A.: Sound and Complete Axiomatizations of Coalgebraic Language Equivalence. ACM Trans. Comput. Logic **14**(1), 7:1–7:52 (Feb 2013). https://doi.org/10.1145/2422085.2422092
5. Brown, R., Higgins, P.J.: Tensor products and homotopies for ω-groupoids and crossed complexes. Journal of Pure and Applied Algebra **47**(1), 1–33 (Jan 1987). https://doi.org/10.1016/0022-4049(87)90099-5
6. Day, B.J.: Construction of Biclosed Categories. Ph.D. thesis, University of New South Wales (Sep 1970), http://web.science.mq.edu.au/~street/DayPhD.pdf
7. Ésik, Z., Németh, Z.L.: Higher Dimensional Automata. Journal of Automata **9**(1), 329 (2004). https://doi.org/10.25596/JALC-2004-003
8. Fahrenberg, U., Johansen, C., Struth, G., Ziemianski, K.: Languages of Higher-Dimensional Automata. Math. Struct. Comput. Sci. **31**(5), 575–613 (2021). https://doi.org/10.1017/S0960129521000293
9. Fahrenberg, U., Johansen, C., Struth, G., Ziemiański, K.: A Kleene Theorem for Higher-Dimensional Automata. In: Klin, B., Lasota, S., Muscholl, A. (eds.) CONCUR 2022. Leibniz International Proceedings in Informatics (LIPIcs), vol. 243, pp. 29:1–29:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.CONCUR.2022.29
10. Fahrenberg, U., Legay, A.: History-Preserving Bisimilarity for Higher-Dimensional Automata via Open Maps. In: Proceedings of MFPS 29. pp. 165–178 (2013). https://doi.org/10.1016/j.entcs.2013.09.012
11. Fajstrup, L., Goubault, E., Raußen, M.: Detecting Deadlocks in Concurrent Systems. In: CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings. pp. 332–347 (1998). https://doi.org/10.1007/BFb0055632
12. Goubault, E.: Geometry and concurrency: A user's guide. Math. Struct. Comput. Sci. **10**(4), 411–425 (2000), http://journals.cambridge.org/action/displayAbstract?aid=54593
13. Grabowski, J.: On partial languages. Fundam. Informaticae **4**(2), 427 (1981)
14. Grandis, M.: Directed Algebraic Topology: Models of Non-Reversible Worlds. New Mathematical Monographs, Cambridge University Press, Cambridge (2009). https://doi.org/10.1017/CBO9780511657474
15. Hoare, C.A.R.T., Möller, B., Struth, G., Wehrman, I.: Concurrent Kleene Algebra. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009 - Concurrency Theory. pp. 399–414. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04081-8_27
16. Hoare, T., Möller, B., Struth, G., Wehrman, I.: Concurrent Kleene Algebra and its Foundations. The Journal of Logic and Algebraic Programming **80**(6), 266–296 (Aug 2011). https://doi.org/10.1016/j.jlap.2011.04.005

17. Im, G.B., Kelly, G.M.: A universal property of the convolution monoidal structure. Journal of Pure and Applied Algebra **43**(1), 75–88 (Nov 1986). https://doi.org/10.1016/0022-4049(86)90005-8

18. Jipsen, P., Moshier, M.A.: Concurrent Kleene algebra with tests and branching automata. Journal of Logical and Algebraic Methods in Programming **85**(4), 637–652 (Jun 2016). https://doi.org/10.1016/j.jlamp.2015.12.005

19. Kahl, T.: Labeled homology of higher-dimensional automata. J. Appl. Comput. Topol. **2**(3-4), 271–300 (2018). https://doi.org/10.1007/s41468-019-00023-0

20. Kan, D.M.: Abstract Homotopy. I. Proceedings of the National Academy of Sciences of the United States of America **41**(12), 1092–1096 (1955), http://www.jstor.org/stable/89108

21. Kappé, T.: Concurrent Kleene Algebra: Completeness and Decidability. Doctoral, UCL (University College London) (Sep 2020), https://discovery.ucl.ac.uk/id/eprint/10109361/

22. Kappé, T., Brunet, P., Luttik, B., Silva, A., Zanasi, F.: Brzozowski Goes Concurrent - A Kleene Theorem for Pomset Languages. In: Meyer, R., Nestmann, U. (eds.) 28th International Conference on Concurrency Theory (CONCUR 2017). LIPIcs, vol. 85, pp. 25:1–25:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2017). https://doi.org/10.4230/LIPIcs.CONCUR.2017.25

23. Kappé, T., Brunet, P., Luttik, B., Silva, A., Zanasi, F.: On series-parallel pomset languages: Rationality, context-freeness and automata. JLAMP **103**, 130–153 (Feb 2019). https://doi.org/10.1016/j.jlamp.2018.12.001

24. Leinster, T.: Basic category theory, vol. 143. Cambridge University Press (2014)

25. Lodaya, K., Weil, P.: Series–parallel languages and the bounded-width property. Theoretical Computer Science **237**(1), 347–380 (Apr 2000). https://doi.org/10.1016/S0304-3975(00)00031-1

26. Loregian, F.: Coend calculus (Dec 2020). https://doi.org/10.48550/arXiv.1501.02503

27. Milius, S.: A Sound and Complete Calculus for Finite Stream Circuits. In: Proceedings of LICS 2010. pp. 421–430 (2010). https://doi.org/10.1109/LICS.2010.11

28. Milius, S., Bonsangue, M.M., Myers, R.S.R., Rot, J.: Rational Operational Models. In: Proceedings of MFPS 29. pp. 257–282 (2013). https://doi.org/10.1016/j.entcs.2013.09.017

29. Pratt, V.R.: Modeling Concurrency with Geometry. In: Conference Record of the Eighteenth Annual ACM Symposium on Principles of Programming Languages (POPL). pp. 311–322 (1991). https://doi.org/10.1145/99583.99625

30. Raussen, M.: Connectivity of spaces of directed paths in geometric models for concurrent computation. CoRR **abs/2106.11703** (2021), https://arxiv.org/abs/2106.11703

31. Riehl, E.: Category Theory in Context. Aurora: Dover Modern Math Originals, Dover Publications (2016), http://www.math.jhu.edu/~eriehl/context/

32. van Glabbeek, R.J.: On the expressiveness of higher dimensional automata. Theor. Comput. Sci. **356**(3), 265–290 (2006). https://doi.org/10.1016/j.tcs.2006.02.012

A Notation

Notation	Meaning
$\mathbf{C}$	Standard or specific categories
$\mathbf{Set}$	Category of sets
$\mathbf{Top}$	Category of topological spaces
よ	Yoneda embedding
Σ	Fixed alphabet
$\lvert P\rvert$	Carrier of iposet P
$A^{\downarrow}$	Downwards closure
ε	empty lo-set
$\ell\mathbf{SLO}$	category of labelled strict linear orders
$\star$	monoidal product of $\ell\mathbf{SLO}$
$\mathbf{n}$	finite ordinal with n elements (possibly empty!)
$[n]$	finite ordinal with $n+1$ elements (spine of n-simplex)
$\boxdot$	Full labelled precube category
$\square$	Labelled precube category (skeletal)
$d_{A,B}$	Coface map arising from the inclusion $U \setminus (A \cup B) \to U$
$\mathbf{HDA}$	Category of HDA
$\mathcal{C}$	Generic category
$\mathcal{C}^{\mathrm{op}}$	Opposite category
$\mathbf{PSh}(\mathcal{I})$	$\mathbf{Set}$-Valued presheaves indexed by $\mathcal{I}$
$X_{\perp}$	Starting cells of HDA
$X^{\top}$	Accepting cells of HDA
$\left(X, X_{\perp}, X^{\top}\right)$	Tuple that makes an HDA
$\mathbf{Lang}$	Category of languages
`iiPom`	The set of interval ipomsets
$s(\alpha)$	Source of path α in an HDA
$t(\alpha)$	Source of path α in an HDA

B Convolution Product on HDA

B.1 Day Convolution Precubical Sets is Coproduct

In definition 13 we defined the tensor products of HDA as extending the tensor product of precubical sets given by Day convolution with appropriate starting and accepting cells. We show here that the coend formula

$$X \otimes Y = \int^{V,W} \square(-, V \oplus W) \times X[V] \times Y[W] \tag{1}$$

for Day convolution reduces to a coproduct formula

$$(X \otimes Y)(U) \cong \coprod_{U = V \oplus W} X[V] \times Y[W] \tag{2}$$

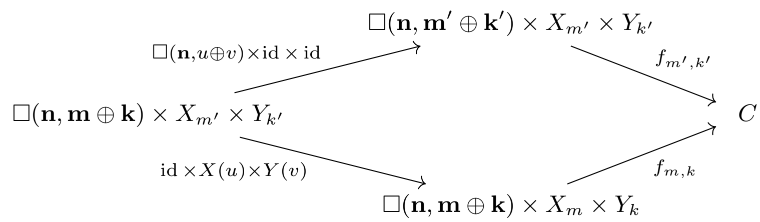
and thus reduces to the standard definition [5,14,20]

Recall that objects in $\ell\mathbf{SLO}$ are pairs $(\mathbf{n}, w)$ where $n \in \mathbb{N}$ and w is a word of length n over Σ. Let us write $i_{n,j} \colon \mathbf{n} \to \mathbf{n+1}$ for the unique map that does not have j in its image. Clearly, any map $(\mathbf{n}, w) \to (\mathbf{n+1}, w')$ is determined by the embedding maps $i_{n,j}$. Therefore, we will leave out in the remainder the words w and pretend that $\ell\mathbf{SLO}$ consists of unlabelled finite ordinals $\mathbf{n}$. Further, a map $d \colon \mathbf{n} \to \mathbf{n+1}$ in $\square$ comes with a partition of the complement image and is therefore given by either $(i_{n,j}, \{j\}, \emptyset)$ or $(i_{n,j}, \emptyset, \{j\})$. For what follows, this duplication of morphisms also makes no difference and we focus attention on the maps $i_{n,j}$.

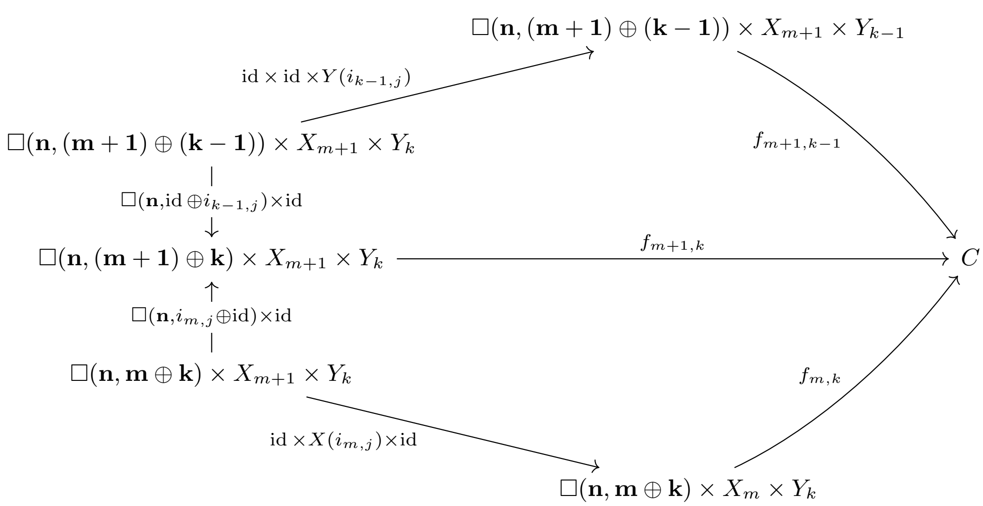
The strategy to show that eq. (2) holds is to show that any cowedge for the coend in eq. (1) is uniquely determined by a cocone for the coproduct in eq. (2). Write $F_{n,X,Y} \colon \square \times \square \times \square^{\mathrm{op}} \times \square^{\mathrm{op}} \to \mathbf{Set}$ for the functor given by

$$F_{n,X,Y}(\mathbf{m}, \mathbf{k}, \mathbf{m}', \mathbf{k}') = \square(\mathbf{n}, \mathbf{m} \oplus \mathbf{k}) \times X_{m'} \times Y_{k'}$$

on objects, which gives us $(X \otimes Y)_n = \int^{\mathbf{m},\mathbf{k}} F_{n,X,Y}(\mathbf{m}, \mathbf{k}, \mathbf{m}, \mathbf{k})$. Suppose now that $f \colon F \to C$ is a cowedge, which means that it consists of maps $f_{m,k} \colon \square(\mathbf{n}, \mathbf{m} \oplus \mathbf{k}) \times X_m \times Y_k \to C$ in $\mathbf{Set}$, such that the following diagram commutes for all $u \colon \mathbf{m} \to \mathbf{m}'$ and $v \colon \mathbf{k} \to \mathbf{k}'$.



Suppose now that $n = m + k$ and consider the following diagram, which commutes for all appropriate choices of j since f is a cowedge.



But then $f_{m+1,k}$ is determined from $f_{m+1,k-1}$ and $f_{m,k}$, since any map $\mathbf{n} \to (\mathbf{m}+\mathbf{1}) \oplus \mathbf{k}$ is uniquely determined by the only number j that is not in its image. These are exactly the maps obtained as the image of the maps $\square(\mathbf{n}, i_{m,j} \oplus \mathrm{id})$ and $\square(\mathbf{n}, \mathrm{id} \oplus i_{k-1,j})$. Hence, the parts in the coend of eq. (1) where $n < k+m$ do not contribute and it suffices to consider splittings of $n = m+k$. This gives us eq. (2).

C Proofs

C.1 Proofs for section 2

Proof (Proof of lemma 1 on page 6). Let $L : D \to \mathbf{Lang}$ be a small diagram of down-closed interval ipomset languages and let $L_\cup = \bigcup_{d \in D} L_d$ be their union. Then for every $Q \in L_\cup$ there exists at least one $d \in D$ such that $Q \in L_d$, which means that Q has to be an interval ipomset. Moreover for every $P \in \texttt{iiPom}$ with $P \sqsubseteq Q$ we by definition have $P \in L_d$ which means that we have to have $P \in L_\cup$ as well. Therefore $L_\cup$ is down-closed as well.

Proof (Proof of Lemma 2 on page 6). Suppose that $L_1 = \bigcup_{(d,e) \in D \times E} M_d \parallel N_e$ and $L_2 = \left(\bigcup_{d \in D} M_d\right) \parallel \left(\bigcup_{e \in E} N_e\right)$.

Suppose that $R \in L_1$. Then there exist $d \in D$ and $e \in E$ such that $R \in M_d \parallel N_e$. Then there exists a $P \in M_d$ and a $Q \in N_e$ such that $R \sqsubseteq P \parallel Q$. Since $P \in \bigcup_{d \in D} M_d$ and $Q \in \bigcup_{e \in E} N_e$ this means that $P \parallel Q \in L_2$ and therefore $R \in L_2$. This gives us $L_1 \subseteq L_2$.

Suppose that $R \in L_2$. Then there exists a $P \in \bigcup_{d \in D} M_d$ and a $Q \in \bigcup_{e \in E} N_e$ such that $R \sqsubseteq P \parallel Q$. Therefore there exist $d \in D$ and $e \in E$ such that $P \in M_d$ and $Q \in N_e$, which means that $P \parallel Q \in M_d \parallel N_e$ and therefore $P \parallel Q \in L_1$. This gives us $R \in L_1$ and therefore $L_1 \supseteq L_2$ which means that we have $L_1 = L_2$.

C.2 Proofs for section 3.1

Proof (Proof of lemma 3 on page 7). Composition of $(e, C, D) \colon V \to W$ and $(d, A, B) \colon U \to V$ is given by $(e, C, D) \circ (d, A, B) = (e \circ d, e(A) \cup C, e(B) \cup D)$. That $\{e(A) \cup C, e(B) \cup D\}$ form a partition of the complement image of $e \circ d$ follows from injectivity of e, properties of the image and the given partitions. The identity is given by $(\mathrm{id}, \emptyset, \emptyset)$, and the unit and associativity axioms follow from colimit preservation of the image. The monoidal structure in inherited from $\ell\mathbf{SLO}$: on objects we use $\star$ and on morphisms we take $(d_1, A_1, B_1) \oplus (d_2, A_2, B_2) = (d_1 \star d_2, A_1 \star A_2, B_1 \star B_2)$, where we write $A_1 \star A_2$ for the application of $\star$ to the inclusions $A_k \subseteq V$. Finally, the associator and unitor isomorphisms have empty complement image that can be trivially partitioned.

Definition 20. *Let D be a small category and let $F : D \to \mathbf{PSh}(\square)$ be a small diagram of precubical sets. For each object U in $\square$ we define the relation $\sim$ on $\coprod_{d \in D} F(d)[U]$ as the transitive closure of*

$$\left\{ (x, y) \,\middle|\, \begin{matrix} d, e \in D,\ x \in F(d)[U],\ y \in F(e)[U] \\ \exists c \in D,\ f : d \to c,\ g : e \to c \text{ s.t. } \left(F(f)[U]\right)(x) = \left(F(g)[U]\right)(y) \end{matrix} \right\}$$

Note that if D is a filtered category the above is already transitive.

Lemma 14. *Let D be a small category and let $F : D \to \mathbf{PSh}(\square)$ be a small diagram of precubical sets. Then for each object U in $\square$ we have*

$$\left(\operatorname*{colim}_{d\in D} F(D)\right)[U] \cong \operatorname*{colim}_{d\in D}\left(F(d)[U]\right) \cong \left(\coprod_{d\in D}\left(F(d)[U]\right)\right) \Big/ \sim$$

where $\sim$ is the relation defined in definition 20.

Proof. Proposition 8.8 from [2] gives us the first isomorphism and the second isomorphism follows from the description of colimits in the category of sets (see, for instance, Example 5.2.16 of [24]).

Theorem 8. *Let (X, ϕ) be a colimit of the small diagram $F : D \to \mathbf{PSh}(\square)$ of precubical sets. Then for all objects U in $\square$, all $d, e \in D$, $x \in F(d)[U]$ and $y \in F(e)[U]$ we have*

$$x \sim y \iff \phi(d)[U](x) = \phi(e)[U](y)$$

Proof. lemma 14 gives us that for all objects U in $\square$ there exists a bijection $q[U] : X[U] \to \left(\coprod_{d\in D}\left(F(d)[U]\right)\right) / \sim$. For all $d \in D$ and every object U in $\square$ there also exists a unique set map $\psi_{d,U} : F(d)[U] \to \left(\coprod_{d\in D}\left(F(d)[U]\right)\right) / \sim$. We then have $q[U] \circ \phi(d)[U] = \psi_{d,U}$ which because $q[U]$ is a bijection gives us

$$x \sim y \iff \psi_{d,U}(x) = \psi_{e,U}(y) \iff \phi(d)[U](x) = \phi(e)[U](y)$$

which proves the statement.

Lemma 15. *Let $F : D \to \mathbf{HDA}$ be a small diagram of HDA with the colimit (X, ϕ). Then for all $U \in \square$ and all $x \in X[U]$ there exists a $d \in D$ and a $y \in F(d)[U]$ such that $\phi_d[U](y) = x$ and*

$$x \in X_\perp \iff y \in F(d)_\perp$$

$$x \in X^\top \iff y \in F(d)^\top$$

If D is discrete then this $y \in F(d)[U]$ is unique.

Proof. The fact that for each $x \in X[U]$ there exists a $d \in D$ and a $y \in F(d)[U]$ with $\phi_d[U](y) = x$ follows from theorem 8. Suppose that we have $x \in X_\perp$ but $y \notin F(d)_\perp$ for all $y \in F(d)[U]$ with $\phi_d[U](y) = x$. Then we can define (X', ϕ') as the cocone of F with the same underlying precubical set and maps as (X, ϕ) but with $x \notin X'_\perp$. Then there exists no unique HDA map $q : X \to X'$ as per the universal property, which is in contradiction with X being the colimit. Combined with the above working analogously for the accepting cells gives us that there must exist a $y \in F(d)[U]$ which reflects the starting and accepting cells of $\phi_d[U](y) = x$.

Since a discrete category D contains no morphisms for all $d_1, d_2 \in D$, $y_1 \in F(d_1)[U]$, $y_2 \in F(d_2)[U]$ with $\phi_{d_1}[U](y_1) = \phi_{d_2}[U](y_2)$ because of theorem 8 we have $y_1 \sim y_2$ and therefore $d_1 = d_2$ and $y_1 = y_2$.

Lemma 16. *Let (X, ϕ) be a cocone of the small diagram $F : D \to \mathbf{PSh}(\square)$ of precubical sets such that for all objects U in $\square$, all $d, e \in D$, $x \in F(d)[U]$ and $y \in F(e)[U]$ we have*

$$x \sim y \iff \phi(d)[U](x) = \phi(e)[U](y)$$

and suppose that for all $x \in X[U]$ there exists a $d \in D$ and a $y \in F(d)[U]$ such that $\phi_d[U](y) = x$. Then (X, ϕ) is a colimit.

Proof. Suppose that (Y, ψ) is a colimit of $F : D \to \mathbf{PSh}(\square)$ and let $q : Y \to X$ be the unique precubical map with $q \circ \psi_d = \phi_d$ for all $d \in D$. Because of the first property of X and lemma 15 this map is injective, and because of the second property it is surjective. Therefore (X, ϕ) is isomorphic to (Y, ψ) through the cocone map $q : Y \to X$ which means that (X, ϕ) is a colimit.

C.3 Proofs for section 3.3

Proof (Proof of lemma 5 on page 9). Let $F\colon D \to \mathbf{HDA}$ be a small diagram of HDA. We write $F'\colon D \to \mathbf{PSh}(\square)$ for $\mathcal{F} \circ F$. Since $\mathbf{PSh}(\square)$ is a cocomplete category there exists a colimit (L', ϕ) of this diagram.

We can then convert this colimit of precubical sets back to a HDA. Let L be the HDA with the underlying precubical set L'. The starting and accepting cells $L_\perp$ and $L^\top$ we define as follows: For every object U in $\square$, every $d \in D$ and every $x \in F(d)[U]$ we have

$$x \in F(d)_\perp \implies \phi(d)[U](x) \in L_\perp$$

$$x \in F(d)^\top \implies \phi(d)[U](x) \in L^\top$$

The precubical maps $\phi(d) : F(d) \to L$ then by definition preserve starting and accepting cells making them HDA maps. Therefore (L, ϕ) is a cocone of the diagram $F : D \to \mathbf{HDA}$.

In fact, we define the sets of starting and accepting cells of $L[U]$ as the colimits of the sets of starting and accepting cells of $F(d)[U]$. It is clear from the construction that $(L, L_\perp, L^\top)$ is the colimit.

Lemma 17. *Let $F : D \to \mathbf{PSh}(\square)$ be a filtered diagram with the filtered colimit (X, ϕ). Let S be a finite set of pairs (U, x) with $U \in \square$ and $x \in X[U]$. Then there exists a $d \in D$ and a finite set S' of pairs (U, y) with $U \in \square$ and $y \in F(d)[U]$ such that the universal map of the colimit provides a bijection $q : S' \to S$ that maps (U, y) to $(U, \phi_d(y))$ with the property that for all $(U, y) \in S'$ if $(V, \delta_{A,B} \circ \phi_d[U](y)) \in S$ for a certain $V \in \square$ then $(V, \delta_{A,B}(y)) \in S'$.*

Proof. For each $U \in \square$ and $x \in X[U]$ such that $(U, x) \in S$ there exists a $d_x \in D$ and a $y_x \in F(d_x)[U]$ such that $\phi_{d_x}[U](y_x) = x$. Because D is filtered there exists a $d \in D$ and morphisms $g_x : d_x \to d$ for each $d_x \in D$ corresponding to a $x \in X[U]$ for a certain $U \in \square$. Therefore we can assume that each y_x resides in the same precubical set $F(d)$. Here we have that for all $(U, x) \in S$ there exists a $y_x \in F(d)[U]$ such that $\phi_d[U](y_x) = x$. We can define the set map q^{-1} that sends (U, x) to (U, y_x). This then automatically gives us our finite set S' and our bijection $q : S' \to S$.

Let $(U, y) \in S'$ and suppose that $(V, \delta_{A,B} \circ \phi_d[U](y)) \in S$ for a certain $V \in \Box$. Then there exists a $(V, y') \in S'$ such that $\phi_d[V](y') = \delta_{A,B} \circ \phi_d[U](y) = \phi_d[V] \circ \delta_{A,B}(y)$, which gives us $y' \sim \delta_{A,B}(y)$. Therefore there exists a $e \in D$ and a morphism $f : d \to e$ such that $F(f)[V](y') = F(f)[V](\delta_{A,B}(y))$.

Since there are only a finite amount of elements in S' and only a finite amount of elements that can be reached form a certain element by the face maps this means that there exists a $d \in D$ and a finite set S' with the bijection $q : S' \to S$ for which we have that for all $(U, y) \in S'$ if $(V, \delta_{A,B} \circ \phi_d[U](y)) \in S$ for a certain $V \in \Box$ then $(V, \delta_{A,B}(y)) \in S'$.

Lemma 18. *Let X be a finite HDA, let $F : D \to$ **HDA** be a filtered diagram with the colimit (Y, ϕ) and let $f : X \to Y$ be a HDA map. Then there exists a $d \in D$ such that there exists a HDA map $g : X \to F(d)$ with $\phi_d \circ g = f$.*

Proof. Let S be the set of pairs $(U, f[U](x))$ with $U \in \Box$ and $x \in X[U]$. Then, lemma 17 says that there exists a $d \in D$ with a set S' of pairs (U, y), $y \in F(d)[U]$ such that if $(U, y) \in S'$ and $(V, \delta_{A,B} \circ \phi_d(y)) \in S$ then $(V, \delta_{A,B}(y)) \in S'$. This means that for each $x \in X[U]$ there exists a certain $y_x \in F(d)[U]$ such that $f[U](x) = \phi_d[U](y_x)$ and such that for all $V \in \Box$ and all face maps $\delta_{A,B}$ we have $f[V] \circ \delta_{A,B}(x) = \phi_d[V] \circ \delta_{A,B}(y_x) = \phi_d[V]\left(y_{\delta_{A,B}(x)}\right)$. This in turn gives us the precubical map $g : X \to F(d)$ with $\phi_d \circ g = f$. By lemma 15 we can also assume that $g : X \to F(d)$ is a HDA map, by choosing the y_x reflecting the starting and accepting cells of $\phi_d[U](y_x) = x$.

Differently stated, lemma 18 says that if X is a finite HDA and $F : D \to$ **HDA** is a filtered diagram with the colimit (Y, ϕ), then any HDA map $f : X \to Y$ factors through some $F(d)$.

Lemma 19. *Let X be a finite HDA, let $F : D \to$ **HDA** be a filtered diagram with the colimit (Y, ϕ) and let $f_1, f_2 : X \to F(d)$ be HDA maps for a certain $d \in D$. Then we have $\phi_d \circ f_1 = \phi_d \circ f_2$ if and only if there exists a $e \in D$ and a morphism $g : d \to e$ such that $F(g) \circ f_1 = F(g) \circ f_2$.*

Proof. Suppose that there exists a $e \in D$ and a morphism $g : d \to e$ such that $F(g) \circ f_1 = F(g) \circ f_2$. Then we have $\phi_e \circ F(g) \circ f_1 = \phi_e \circ F(g) \circ f_2$ which automatically gives us $\phi_d \circ f_1 = \phi_d \circ f_2$, since for all $U \in \Box$ and all $x \in X[U]$ we have

$$\phi_d \circ f_1[U](x) = \phi_e \circ F(g) \circ f_1[U](x) = \phi_e \circ F(g) \circ f_2[U](x) = \phi_d \circ f_2[U](x)$$

For the other direction, suppose that we have $\phi_d \circ f_1 = \phi_d \circ f_2$. Then for all $U \in \Box$ and all $x \in X[U]$ we have $\phi_d \circ f_1[U](x) = \phi_d \circ f_2[U](x)$. By theorem 8 there exist $e_x \in D$ and morphisms $g_1, g_2 : d \to e_x$ such that $F(g_1) \circ f_1[U](x) = F(g_2) \circ f_2[U](x)$. Because D is filtered there exists a $e'_x \in D$ and a $h : e_x \to e'_x$ such that $h \circ g_1 = h \circ g_2$. For the sake of convenience we say that for all

$U \in \square$ and all $x \in X[U]$ there exists a $e_x \in D$ and a $g_x : d \to e_x$ such that $F(g_x) \circ f_1[U](x) = F(g_x) \circ f_2[U](x)$.

Since X is finite this gives us only a finite amount of $e_x \in D$. Therefore there exists a $e \in D$ and morphisms $h_x : e_x \to e$ for each $U \in \square$ and each $x \in X[U]$. This gives us the morphisms $h_x \circ g_x : d \to e$ which then because of D being a filtered category gives us a morphism $h : e \to e'$ such that $h \circ h_x \circ g_x = h \circ h_y \circ g_y$ for all $U, V \in \square$ and all $x \in X[U]$, $y \in X[V]$.

Therefore for all $U \in \square$ and all $x \in X[U]$ we have a morphism $h \circ h_x \circ g_x : d \to e'$. This morphism is the same for all $U \in \square$ or $x \in X[U]$. Renaming e' to e and $h \circ h_x \circ g_x$ to g gives us the required morphism.

Lemma 20. *All finite precubical sets or HDA are compact*

Proof. Since a precubical set can be seen as a special case of HDA (one with empty starting and accepting cells) we will just consider the HDA.

Let X be a finite HDA and let $F : D \to \mathbf{HDA}$ be a small filtered diagram with the colimit (Y, ϕ). This gives us the small filtered diagram $\mathrm{Hom}(X, F(-)) : D \to \mathbf{Set}$ which has the filtered colimit $(\mathrm{colim}_{d \in D} \mathrm{Hom}(X, F(d)), \Phi)$ and the cocone $(\mathrm{Hom}(X, Y), \mathrm{Hom}(X, \phi_d))$ with the unique cocone map $q : \mathrm{colim}_{d \in D} \mathrm{Hom}(X, F(d)) \to \mathrm{Hom}(X, Y)$.

Suppose that $f \in \mathrm{Hom}(X, Y)$. Then from lemma 18 it follows that there exists a $d \in D$ and a $g \in \mathrm{Hom}(X, F(d))$ such that $\phi_d \circ g = f$ and therefore $\mathrm{Hom}(X, \phi_d)(g) = f$. Since we have $g \circ \Phi_d = \mathrm{Hom}(X, \phi_d)$ this means that q is surjective.

Suppose that $f_1, f_2 \in \mathrm{colim}_{d \in D} \mathrm{Hom}(X, F(d))$ such that $q(f_1) = q(f_2)$. Then by definition there exists a $d \in D$ and $g_1, g_2 \in \mathrm{Hom}(X, F(d))$ such that $\Phi_d(g_1) = f_1$ and $\Phi_d(g_2) = f_2$ (we can assume that g_1 and g_2 are in the same set due to D being filtered). Then $q \circ \Phi_d(g_1) = q(f_1) = q(f_2) = q \circ \Phi_d(g_2)$ which gives us $\phi_d \circ g_1 = \phi_d \circ g_2$. Then lemma 19 gives us that there exists an object $e \in D$ and a morphism $h : d \to e$ such that $F(h) \circ g_1 = F(h) \circ g_2$. This then gives us the morphism $\mathrm{Hom}(X, F(h)) : \mathrm{Hom}(X, F(d)) \to \mathrm{Hom}(X, F(d))$ for which we have $\mathrm{Hom}(X, F(h))(g_1) = \mathrm{Hom}(X, F(h))(g_2)$, which means that we have to have $\Phi_d(g_1) = \Phi_d(g_2)$. Therefore q is injective as well, which means that it is an isomorphisms which therefore gives us that X is compact.

Since every representable precubical set is finite by definition this means that they are compact as well.

Definition 21. *Let X be a precubical set or HDA. Then the category of elements el(X) is the category where*

- *an object is a pair (U, x) with $U \in \square$ an object and $x \in X[U]$.*
- *A morphism $(U, x) \to (V, y)$ consists of a coface map $d_{A,B} : U \to V$ such that $\delta_{A,B}(y) = x$.*

The category comes with a forgetful functor $p :$ el$(X) \to \square$ with $p \circ (U, x) = U$.

Lemma 21. *Let X be a precubical set and let $el(X)$ be the category of elements. We have the Yoneda embedding $よ : \square \to \mathbf{PSh}(\square)$ that sends each object of $\square$ to its respective representable precubical set. Then X is a colimit of the diagram $よ \circ p : el(X) \to \mathbf{PSh}(\square)$ of finite precubical sets.*

Proof. This is the density theorem applied on precubical sets.

Lemma 22. *Let X be a precubical set. Then X can be canonically expressed as the colimit of a diagram $F : el(X) \to \mathbf{PSh}(\square)$ of representable precubical sets. Suppose that we have $y_1 \in F(d_1)[U]$, $y_2 \in F(d_2)[U]$ with $y_1 \sim y_2$ for certain $d_1, d_2 \in el(X)$ and an object $U \in \square$. Then there exists a $d_3 \in el(X)$ and morphisms $f_1 : d_3 \to d_1$ and $f_2 : d_3 \to d_2$ in $el(X)$ such that there exists a $x \in F(d_3)[U]$ with $F(f_1)[U](x) = y_1$ and $F(f_2)[U](x) = y_2$.*

Proof. From lemma 21 we get the diagram $F : \mathrm{el}(X) \to \mathbf{PSh}(\square)$ of which (X, ϕ) is a colimit. Since $y_1 \sim y_2$ theorem 8 gives us that $\phi_{d_1}[U](y_1) = \phi_{d_2}[U](y_2) = x \in X[U]$. Then there exists an object $d_3 = (U, x)$ in $\mathrm{el}(X)$. Then there also exists a $x' \in F(d_3)[U]$ such that $\phi_{d_3}[U](x') = x$.

Let $d_1 = (V_1, z_1)$ and $d_2 = (V_2, z_2)$. Let the unique element of $F(d_1)[V_1]$ be z_1' and let the unique element of $F(d_2)[V_2]$ be z_2'. Then there exist coface maps $d_{A_1,B_1} : V_1 \to U$ and $d_{A_2,B_2} : V_2 \to U$ such that $\delta_{A_1,B_1}(z_1') = y_1$ and $\delta_{A_2,B_2}(z_2') = y_2$.

Therefore we have $\phi_{d_1}[U] \circ \delta_{A_1,B_1}(z_1') = \phi_{d_1}[U](y_1) = x$ and $\phi_{d_2}[U] \circ \delta_{A_2,B_2}(z_2') = \phi_{d_2}[U](y_2) = x$. This then means that $\delta_{A_1,B_1}(z_1) = x = \delta_{A_2,B_2}(z_2)$. By definition of $\mathrm{el}(X)$ this means that there exist morphisms $f : (U, x) \to (V_1, z_1)$ and $g : (U, x) \to (V_2, z_2)$ such that $F(f)[U](x') = y_1$ and $F(g)[U](x') = y_2$, which proves the statement.

Proof (Proof of lemma 6 on page 9). Let $(X, X_\perp, X^\top)$ be a HDA and suppose that X is empty (for all objects U of $\square$ we have $X[U] = \emptyset$). Then we can express X as the filtered colimit of the diagram $H : D \to \mathbf{HDA}$ where D is a discrete category containing only a single object d (and therefore also a filtered category) with $F(d) = X$.

Let $(X, X_\perp, X^\top)$ be a non-empty HDA. By the density theorem, every precubical set can be expressed canonically as the colimit of finite precubical sets, i.e, there exists a diagram $F : D \to \mathbf{PSh}(\square)$, so that $X \cong \mathrm{colim}_{d \in D} F(d)$. We convert this diagram into a diagram of finite HDA $F : D \to \mathbf{HDA}$ where $x \in F(d)_\perp \iff \phi_d(x) \in X_\perp$ and $x \in F(d)^\top \iff \phi_d(x) \in X^\top$. The colimit of this diagram of HDA is exactly $(X, X_\perp, X^\top)$ which is by definition of the colimit of HDA.

The category D used in the density theorem is the category of elements $\mathrm{el}(X)$ of X. Let S be a finite full subcategory of $\mathrm{el}(X)$ and let $G_S : S \to \mathbf{HDA}$ be the finite diagram of HDA where $G_S(d) = F(d)$ for every object d of S and $G_S(f) = F(f)$ for every morphism $f : d \to e$ in S.

Let E be the (small) category of finite full subcategories of $\mathrm{el}(X)$ where the morphisms are the canonical inclusion functors. The category E is filtered since it is not empty, has no parallel morphisms and for each pair of objects S_1

and S_2 of E there exists a third object S_3 (the full subcategory of $\mathrm{el}(X)$ with $\mathrm{obj}\,(S_3) = \mathrm{obj}\,(S_1) \cup \mathrm{obj}\,(S_2)$) and morphisms $f_1 : S_1 \to S_3,\ f_2 : S_2 \to S_3$.

Let $H : E \to \mathbf{HDA}$ be the filtered diagram with $H(S) = \mathrm{colim}_{s\in S}\, G_S(s)$ for all $S \in E$. Because $G_S : S \to \mathbf{HDA}$ is a finite diagram of finite HDA its colimit $H(S)$ must be a finite HDA as well. For all $S_1, S_2 \in E$ there exists a morphism $f : S_1 \to S_2$ if and only if S_1 is a full subcategory of S_2. In this case $\mathrm{colim}_{s\in S_2}\, G_{S_2}(s)$ is a cocone of the diagram $G_{S_1} : S \to \mathbf{HDA}$ which gives us the unique HDA map $H(f) : H(S_1) \to H(S_2)$. This makes $H : D \to \mathbf{HDA}$ a well-defined filtered diagram of finite HDA.

Each $S \in E$ is a full subcategory of $\mathrm{el}(X)$ with $G_S(d) = F(d)$ for all $d \in S$ and $G_S(f) = F(f)$ for all morphisms f in E. Therefore X is a cocone of each $G_S : S \to \mathbf{HDA}$ which gives us the unique HDA maps $\varphi_S : H(S) \to X$. Due to the properties of cocone maps we get that for each pair of objects $S_1, S_2 \in E$ with the morphism $f : S_1 \to S_2$ we have $\varphi_{S_2} \circ H(f) = \varphi_{S_1}$, which makes (X, φ) a cocone of $H : E \to \mathbf{HDA}$.

Suppose that we have an object $U \in \square$ and an element $x \in X[U]$. Since (X, ϕ) is a colimit of $F : \mathrm{el}(X) \to \mathbf{HDA}$ there by definition exists a $y \in F((U, x))[U]$ such that $\phi_x[U](y) = x$. By definition there is a category S_x in E containing only the object (U, x) which means that we have $H(S_x) = \mathrm{colim}_{d\in S_x}\, G_{S_x} = F((U, x))$. In this case the cocone map φ_{S_x} is the same as the injection map $\phi_{(U,x)}$, which then gives us $\varphi_{S_x}[U](y) = x$.

Suppose that we have $S_1, S_2 \in E$ and $x_1 \in H(S_1)[U]$, $x_2 \in H(S_2)[U]$ for a certain object $U \in \square$ such that $\varphi_{S_1}[U](x_1) = \varphi_{S_2}[U](x_2)$. Since E is filtered we can simply assume that $S = S_1 = S_2$.

Per definition we have the colimit $(H(S), \theta)$ of $G_S : S \to \mathbf{HDA}$. Then lemma 15 gives us that there exist $d_1, d_2 \in S$ such that there exist $y_1 \in G_S(d_1)[U]$ and $y_2 \in G_S(d_2)[U]$ such that $\theta_{d_1}[U](y_1) = x_1$ and $\theta_{d_2}[U](y_2) = x_2$.

Then because (X, ϕ) is a cocone of $G_S : S \to \mathbf{HDA}$ with the cocone map $\varphi_S : H(S) \to X$ we get

$$\phi_{d_1}(y_1) = \varphi_S \circ \theta_{d_1}[U](y_1) = \varphi_S[U](x_1) = \varphi_S[U](x_2) = \varphi_S \circ \theta_{d_2}[U](y_2) = \phi_{d_2}(y_2)$$

This gives us $\phi_{d_1}(y_1) = \phi_{d_2}(y_2)$ and therefore because of theorem 8 we get $y_1 \sim y_2$ in $F : \mathrm{el}(X) \to \mathbf{HDA}$.

Then because of lemma 22 there exists a $d_3 \in \mathrm{el}(X)$ and morphisms $f : d_3 \to d_1$ and $g : d_3 \to d_2$ in $\mathrm{el}(X)$ such that there exists a $y_3 \in F(d_3)[U]$ with $F(f)[U](y_3) = y_1$ and $F(g)[U](y_3) = y_2$. We have $d_3 = (V, z)$ for some object $V \in \square$ and some $z \in X[V]$.

This gives us that there exists a $S' \in E$ with $\mathrm{obj}\,(S') = S \cup \{(V, z)\}$ and a morphism $h : S \to S'$. S' by definition includes d_1, d_2 and d_3 and the morphisms f and g which gives us that

$$\begin{aligned} H(h)[U](x_1) &= H(h) \circ \theta_{d_1}[U](x_1) = \theta'_{d_1}[U](y_1) \\ &= \theta'_{d_2}[U](y_2) = H(h) \circ \theta'_{d_2}[U](y_2) = H(h)[U](x_2) \end{aligned}$$

with $(H(S'), \theta')$ being the colimit of $G_{S'} : S' \to \mathbf{HDA}$. This gives us that for all $x_1 \in H(S_1)[U]$ and $x_2 \in H(S_2)[U]$ we have $x_1 \sim x_2 \iff \varphi_{d_1}[U](x_1) = \varphi_{d_2}[U](x_2)$.

From lemma 16 it then follows that (X, ϕ) is a filtered colimit of $H : E \to \mathbf{HDA}$ assuming that the starting and accepting cells are correct. Because of the way we defined $F : \mathrm{el}(X) \to \mathbf{HDA}$ this is the case. If $x \in X[U]$ and $x \in X_\bot$ then $F(d)$ with $d = (U, x)$ is defined such that for the element $y \in F(d)[U]$ with $\phi_d[U](y)$ we have $y \in F(d)_\bot$. For $S_x \in E$ the full subcategory containing only $d = (U, x)$ we then have $H(S_x) = F(d)$ such that $\varphi_{S_x}[U](y) = x$. Analogously the same is true for the accepting cells.

Lemma 23. *Every compact precubical set or HDA is finite.*

Proof. We will again only consider the HDA. Let X be a compact HDA and let $F : D \to \mathbf{HDA}$ be a filtered diagram of finite HDA with the filtered colimit (X, ϕ) as per lemma 6. Then, since X is compact, we have

$$\operatorname*{colim}_{d \in D} \mathrm{Hom}\,(X, F(d)) \cong \mathrm{Hom}\left(X, \operatorname*{colim}_{d \in D} F(d)\right) \cong \mathrm{Hom}\,(X, X)$$

As a consequence, we get that the identity map id_X factors through a map $X \to F(d)$. Since $F(d)$ is a finite HDA, X has to be finite as well.

Proof (Proof of theorem 1 on page 9). This follows from lemma 20 and lemma 23.

Proof (Proof of theorem 2 on page 9). **HDA** is cocomplete by lemma 5. Lemma 6 shows that any HDA is given as filtered colimit of compact HDA. Since by theorem 1 the compact HDA are finite, we have that $\mathbf{HDA}_c$ is essentially small. Thus, **HDA** is a lfp category.

Alternative Proofs for section 3.3 An alternative proof that the category of HDA is lfp goes as follows. The idea is to construct a reflective subcategory $\mathcal{H}$ of a presheaf category that is closed under filtered colimits and equivalent to **HDA**. This implies then that $\mathcal{H}$ and thus **HDA** is lfp [1, Sec. 1C]. We thank an anonymous referee of a previous version of the paper for this suggestion.

We define a category $\ell\mathbf{SLO}^{\triangleright}$ with objects

$$|\ell\mathbf{SLO}^{\triangleright}| = |\ell\mathbf{SLO}| \cup \{\top, \bot\} \times |\ell\mathbf{SLO}|$$

and morphisms between objects are given as follows.

$$\ell\mathbf{SLO}^{\triangleright}(X, Y) = \begin{cases} \ell\mathbf{SLO}(X, Y), & X, Y \in |\ell\mathbf{SLO}| \\ \{*_\top\}, & X \in |\ell\mathbf{SLO}|, Y = (\top, X) \\ \{*_\bot\}, & X \in |\ell\mathbf{SLO}|, Y = (\bot, X) \\ \emptyset, & \text{otherwise} \end{cases}$$

We will write $U_\top$ and $U_\bot$ instead of $(\top, U)$ and $(\bot, U)$ for $U \in \ell\mathbf{SLO}$. Let $\mathcal{H}$ be the full subcategory of $\mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright})$ of presheaves P for which $P(*_\top)$ and $P(*_\bot)$ are injective. The idea is that $P(U^\top)$ and $P(U_\bot)$ contain the starting and accepting cells of dimension U. We now have to show that $\mathcal{H}$ is a reflective subcategory of $\mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright})$, closed under filtered colimits and equivalent to **HDA**.

Reflective subcategory Let $I \colon \mathcal{H}\mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright})$ be the inclusion functor. We construct a left-adjoint T to I using the orthogonal epi-mono factorisation system (E, M) on **Set** as follows. For a presheaf $P \colon (\ell\mathbf{SLO}^{\triangleright})^{\mathrm{op}} \to \mathbf{Set}$ and $U, V \in \ell\mathbf{SLO}$, we define a presheaf TP by $(TP)(U) = P(U)$, $(TP)(k) = Pk$ for $k \colon V \to U$ and $(TP)(U_\top)$ and $(TP)(U_\bot)$ by the following factorisations into a surjection followed by an injection.

$$P(U_\bot) \xrightarrow{\eta_{P,U_\bot}} (TP)(U_\bot) \xrightarrow{(TP)(*_\bot)} P(U)$$
$$P(U_\top) \xrightarrow{\eta_{P,U_\top}} (TP)(U_\top) \xrightarrow{(TP)(*_\top)} P(U)$$

Since the epi-mono factorisation system is functorial, this assignment defines a functor $T \colon \mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright}) \to \mathcal{H}$. If we put $\eta_{P,U} = \mathrm{id}_{P(U)}$ for $U \in \ell\mathbf{SLO}$, then this yields together with the above factorisation a natural transformation $\eta \colon \mathrm{Id} \to IT$. Let now $f \colon P \to K$ be map natural transformation with $K \in \mathcal{H}$. Since (E, M) is an orthogonal factorisation system, there is for all U a unique map $\bar{f}_{U_\bot}$ filling the following diagram.

$$\begin{array}{ccccc}
P(U_\bot) & \xrightarrow{\eta_{P,U_\bot}} & (TP)(U_\bot) & \xrightarrow{T(*_\bot)} & P(U) \\
\downarrow{\scriptstyle f_{U_\bot}} & & \downarrow{\scriptstyle \bar{f}_{U_\bot}} & & \downarrow{\scriptstyle f_U} \\
K(U_\bot) & \xrightarrow[\mathrm{id}]{} & K(U_\bot) & \xrightarrow[K(*_\bot)]{} & K(U)
\end{array}$$

Similarly, there is a unique map $\bar{f}_{U_\top} \colon (TP)(U_\top) \to K(U_\top)$ with $\bar{f}_{U_\top} \circ \eta_{P,U_\top} = f_{U_\top}$. If we put $\bar{f}_U = f_U$, then we obtain a unique natural transformation $\bar{f} \colon TP \to K$ with $\bar{f} \circ \eta_P = f$. Thus, (TP, η) is a reflection of P along I and thus $T \dashv I$. The inclusion I is by definition full and thus $\mathcal{H}$ is a reflective subcategory of the presheaf category $\mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright})$.

$\mathcal{H}$ is closed under filtered colimits Let $\mathcal{C}$ be filtered and $D \colon \mathcal{C} \to \mathcal{H}$ a diagram. Since $\mathbf{PSh}(\ell\mathbf{SLO}^{\triangleright})$ is a presheaf category, the colimit $\operatorname{colim} ID$ is computed point-wise. Thus, it remains to prove that $(\operatorname{colim} ID)(*_\top)$ and $(\operatorname{colim} ID)(*_\bot)$ are injective, where we only prove the first and the second is analogous. We note that the following diagram is a pullback for all $c \in \mathcal{C}$ and $U \in \ell\mathbf{SLO}$ because $D_c(*_\top)$ is a monomorphism (injective).

$$\begin{array}{ccc}
D_c(U_\top) & \xrightarrow{\mathrm{id}} & D_c(U_\top) \\
\downarrow{\scriptstyle \mathrm{id}} & & \downarrow{\scriptstyle D_c(*_\top)} \\
D_c(U_\top) & \xrightarrow[D_c(*_\top)]{} & D_c(U)
\end{array}$$

Since filtered colimits commute with finite limits and because colim preserves identities, the following is also a pullback.

$$\begin{array}{ccc}
(\operatorname{colim} D)(U_\top) & \xrightarrow{\mathrm{id}} & (\operatorname{colim} D)(U_\top) \\
\downarrow{\scriptstyle \mathrm{id}} & \lrcorner & \downarrow{\scriptstyle (\operatorname{colim} D)(*_\top)} \\
(\operatorname{colim} D)(U_\top) & \xrightarrow[(\operatorname{colim} D)(*_\top)]{} & D_c(U)
\end{array}$$

Therefore, $(\operatorname{colim} D)(*_\top)$ is a monomorphism and $\operatorname{colim} D \in \mathcal{H}$.

$\mathcal{H}$ is equivalent to **HDA** This is obvious by mapping $P \in \mathcal{H}$ to the HDA $(X, X_\perp, X_\top)$ with $X(U) = P(U)$, $X_\perp = \bigcup_U P(*_\perp)(U_\perp)$ and $X_\top = \bigcup_U P(*_\top)(U_\top)$. This mapping induces clearly a fully faithful functor that is essentially surjective, and is thus part of an equivalence.

C.4 Proofs for section 4

Proof (Proof of lemma 7 on page 11). This follows directly from the definition of $\mathtt{ev}$.

Proof (Proof of lemma 8 on page 11). If $P \in L(X)$ then there exists a path α in X with $s(\alpha) \in X_\perp$ and $t(\alpha) \in X^\top$ such that $\mathtt{ev}(\alpha) = P$. lemma 7 gives us that $f(\alpha)$ is a path in Y and because HDA maps preserve starting and accepting cells we have $s(f(\alpha)) \in X_\perp$ and $t(f(\alpha)) \in X^\top$ and therefore $P = \mathtt{ev}(\alpha) = \mathtt{ev}(f(\alpha)) \in L(Y)$.

In the case that $f : X \to Y$ is an isomorphism there exists an inverse map $f^{-1} : Y \to X$, which gives us $L(Y) \subseteq L(X)$ as well and therefore $L(X) = L(Y)$.

Proof (Proof of lemma 9 on page 11). For every $d \in D$ we have the HDA map $\phi(d) : F(d) \to X$. Lemma 8 then gives us that $L(F(d)) \subseteq L(X)$, from which the statement follows.

Proof (Proof of lemma 10 on page 11). Suppose that we have $P \in L(X)$. Then there exists an accepting path $\alpha = (x_0, \varphi_1, x_1, ..., \varphi_n, x_n)$ in X with $s(\alpha) \in X_\perp$ and $t(\alpha) \in X^\top$ such that $\mathtt{ev}(\alpha) = P$.

Lemma 15 gives us that for each $x_k \in X[U_k]$ for $1 \le k \le n$ and the object $U_k \in \square$ there exists a unique $d_k \in D$ and a unique $y_k \in F(d)[U_k]$ such that $\phi_{d_k}[U_k](y_k) = x_k$. It also gives us that $y_1 \in F(d_1)_\perp$ and $y_n \in F(d_n)_\perp$.

Suppose that we have $x_k = \delta^0_A(x_{k+1})$. Because we have

$$\phi_{d_k}[U_k](y_k) = x_k = \delta^0_A(x_{k+1}) = \delta^0_A \circ \phi_{d_{k+1}}[U_{k+1}](y_{k+1}) = \phi_{d_k}[U_k] \circ \delta^0_A(y_{k+1})$$

we get $y_k \sim \delta^0_A(y_{k+1})$ (with $\sim$ as defined in definition 20) which because of lemma 15 gives us $d_k = d_{k+1}$ and $y_k = \delta^0_A(y_{k+1})$. Analogously the same works for if we have $\delta^1_B(x_k) = x_{k+1}$.

Therefore there exists an accepting path $\alpha' = (y_0, \varphi_1, y_1, ..., \varphi_n, y_n)$ in $F(d)$ with $d = d_1 = d_2 = ... = d_n$ such that $\phi_d(\alpha') = \alpha$. Lemma 7 gives us that $P = \mathtt{ev}(\alpha) = \mathtt{ev}(\alpha')$ and therefore $\mathtt{ev}(\alpha') \in L(F(d))$. As a result we have that $P \in L(X) \implies P \in \bigcup_{d \in D} L(F(d))$. Combined with lemma 9 this proves the statement.

Proof (Proof of lemma 11 on page 11). For finite HDA X, $L(X)$ is a language by [9, Prop. 10]. Suppose that X is an arbitrary HDA. From lemma 6 we get a filtered diagram $F : D \to \mathbf{HDA}$ of finite HDA such that $X \cong \operatorname{colim}_{d \in D} F(d)$. lemma 8 and theorem 3 give us that

$$L(X) = L\Big(\operatorname*{colim}_{d \in D} F(d)\Big) = \bigcup\nolimits_{d \in D} L(F(d))$$

The result follows because languages are closed arbitrary unions, see lemma 1.

C.5 Proofs for section 5

Proof (Proof of lemma 12 on page 13). Suppose that we are given compact HDA $\mathcal{Y} = (Y, Y_\perp, Y^\top)$ and $\mathcal{Y}' = (Y', Y'_\perp, Y'^\top)$ with morphisms $h\colon \mathcal{Y} \to (X, X_\perp, X^\top)$, $h'\colon \mathcal{Y}' \to (X, X_\perp, X^\top)$, $g\colon P \to Y$ and $g'\colon P \to Y'$, such that $h \circ g = f$ and $h' \circ g' = f$. Since $\mathbf{HDA}_c$ is closed under finite colimits, we can form the coproduct $\mathcal{Y} + \mathcal{Y}'$ with inclusions κ and κ'. Let $[h, h']\colon \mathcal{Y} + \mathcal{Y}' \to \mathcal{X}$ be the copairing of h and h', where $\mathcal{X} = (X, X_\perp, X^\top)$. Because $\mathbf{HDA}$ is lfp, we can factor $[h, h']$ into an epimorphism q and a monomorphism m: $[h, h'] = \mathcal{Y} + \mathcal{Y}' \xrightarrow{e} \mathcal{R} \xrightarrow{m} \mathcal{X}$. We then define $e = q \circ \kappa$ and $e' = q \circ \kappa'$. Note that because q is an epimorphism, $\mathcal{R}$ is a compact HDA. With this notation set up, we have

$$me'g' = mq\kappa'g' = [h, h']\kappa'g' = h'g' = hg = [h, h']\kappa g = mq\kappa g = meg$$

and thus, since m is mono, $e'g' = eg$. □