Myhill-Nerode Theorem for Higher-Dimensional Automata

Uli Fahrenberg
EPITA Research Laboratory (LRE), France

Krzysztof Ziemiański
University of Warsaw, Poland

Abstract. We establish a Myhill-Nerode type theorem for higher-dimensional automata (HDAs), stating that a language is regular if and only if it has finite prefix quotient. HDAs extend standard automata with additional structure, making it possible to distinguish between interleavings and concurrency. We also introduce deterministic HDAs and show that not all HDAs are determinizable, that is, there exist regular languages that cannot be recognised by a deterministic HDA. Using our theorem, we develop an internal characterisation of deterministic languages. Lastly, we develop analogues of the Myhill-Nerode construction and of determinacy for HDAs with interfaces.

Keywords: higher-dimensional automaton; Myhill-Nerode theorem; concurrency theory; determinism

1. Introduction

Higher-dimensional automata (HDAs), introduced by Pratt and van Glabbeek [21, 25, 26], extend standard automata with additional structure that makes it possible to distinguish between interleavings and concurrency. That puts them in a class with other non-interleaving models for concurrency such as Petri nets [20], event structures [19], configuration structures [29,30], asynchronous transition systems [3, 24], and similar approaches [18, 22, 23, 28], while retaining some of the properties and intuition of automata-like models. As an example, Figure 1 shows Petri net and HDA models for a system with two events, labelled a and b. The Petri net and HDA on the left side model the (mutually exclusive) interleaving of a and b as either $a.b$ or $b.a$; those to the right model concurrent execution of a and b. In the HDA, this independence is indicated by a filled-in square.

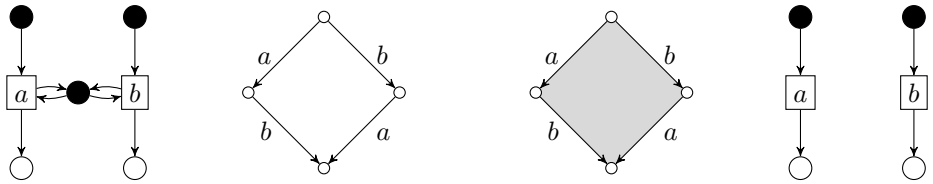


Figure 1. Petri net and HDA models distinguishing interleaving from non-interleaving concurrency. Left: Petri net and HDA models for $a.b + b.a$; right: HDA and Petri net models for $a \parallel b$.

We have recently introduced languages of HDAs [8], which consist of partially ordered multisets with interfaces (ipomsets), and shown a Kleene theorem for them [9, 10]. Here we continue to develop the language theory of HDAs. Our first contribution is a Myhill-Nerode type theorem for HDAs, stating that a language is regular if and only if it has finite prefix quotient. This provides a necessary and sufficient condition for regularity. Our proof is inspired by the standard proofs of the Myhill-Nerode theorem, but the higher-dimensional structure introduces some difficulties. For example, we cannot use the standard prefix quotient relation but need to develop a stronger one which takes concurrency of events into account.

As a second contribution, we give a precise definition of deterministic HDAs and show that there exist regular languages that cannot be recognised by deterministic HDAs. Our Myhill-Nerode construction will produce a deterministic HDA for such deterministic languages, and a non-deterministic HDA otherwise. Our definition of determinism is more subtle than for standard automata as it is not always possible to remove non-accessible parts of HDAs. We develop a language-internal characterisation of deterministic languages.

Thirdly, we develop a variant of the Myhill-Nerode construction and of determinism which uses higher-dimensional automata with interfaces (iHDAs). These were introduced in [9] and allow for some components to be missing which in HDAs would have to exist solely for structural reasons. In iHDAs, non-accessible parts may be removed, which allows for a more principled Myhill-Nerode construction. HDAs and iHDAs are related via mappings called resolution and closure which preserve languages.

We start this paper by introducing languages of ipomsets in Section 2. Section 3 develops important decomposition properties of ipomsets needed in the sequel, and HDAs are introduced in Section 4. Section 5 then states and proves our Myhill-Nerode theorem, and Section 6 introduces deterministic HDAs. HDAs with interfaces are defined in Section 7, the Myhill-Nerode theorem using iHDAs is in Section 8, and deterministic iHDAs are treated in Section 9. This paper is based on [13] which was presented at the 44th International Conference on Application and Theory of Petri Nets and Concurrency. Compared to this conference paper, proofs have been added and errors corrected, and the material in Sections 8 and 9 is new.

2. Pomsets with Interfaces

HDAs model systems in which labelled events have duration and may happen concurrently. Every event has a time interval during which it is active: it starts at some point, then remains active until its termination and never reappears. Events may be concurrent, that is, their activity intervals may overlap; otherwise, one of the events precedes the other. We also need to consider executions in which some events are already active at the beginning (*source events*) or are still active at the end (*target events*).

At any moment of an execution we observe a list of currently active events (such lists are called *conclists* below). The relative position of any two concurrent events on these lists remains the same, regardless of the point in time. This provides a secondary relation between events, which we call *event order*.

To make the above precise, let Σ be a finite alphabet. A *conclist* (for "concurrency list") $(U, \dashrightarrow, \lambda)$ is a finite set U with a total order $\dashrightarrow$ called the *event order* and a labelling function $\lambda : U \to \Sigma$. Conclists (or rather their isomorphism classes) are effectively strings but consist of concurrent, not subsequent, events.

A *labelled poset with event order* (*lposet*) $(P, <, \dashrightarrow, \lambda)$ consists of a finite set P with two relations: *precedence* $<$ and *event order* $\dashrightarrow$, together with a labelling function $\lambda : P \to \Sigma$. Note that different events may carry the same label: we do *not* exclude autoconcurrency. We require that both $<$ and $\dashrightarrow$ are strict partial orders, that is, they are irreflexive and transitive (and thus asymmetric). We also require that for each $x \neq y$ in P, at least one of $x < y$ or $y < x$ or $x \dashrightarrow y$ or $y \dashrightarrow x$ must hold; that is, if x and y are concurrent, then they must be related by $\dashrightarrow$.

Conclists may be regarded as lposets with empty precedence relation; the last condition enforces that their elements are totally ordered by $\dashrightarrow$. A temporary state of an execution is described by a conclist, while the whole execution provides an lposet of its events. The precedence order expresses that one event terminates before the other starts. The event order of an lposet is generated by the event orders of temporary conclists. Hence any two events which are active concurrently are unrelated by $<$ but related by $\dashrightarrow$.

In order to accommodate source and target events, we need to introduce lposets with interfaces (iposets). An *iposet* $(P, <, \dashrightarrow, S, T, \lambda)$ consists of an lposet $(P, <, \dashrightarrow, \lambda)$ together with subsets $S, T \subseteq P$ of source and target *interfaces*. Elements of S must be $<$-minimal and those of T $<$-maximal; hence both S and T are conclists. We often denote an iposet as above by ${}_S P_T$ or (S, P, T), ignoring the orders and labelling, or use $S_P = S$ and $T_P = T$ if convenient. Source and target events will be marked by "•" at the left or right side, and if the event order is not shown, we assume that it goes downwards.

Example 2.1. Figure 2 shows some simple examples of activity intervals of events and the corresponding iposets. The left iposet consists of three totally ordered events, given that the intervals do not overlap; the event a is already active at the beginning and hence in the source interface. In the other iposets, the activity intervals do overlap and hence the precedence order is partial (and the event order non-trivial).

Given that the precedence relation $<$ of an iposet represents activity intervals of events, it is an

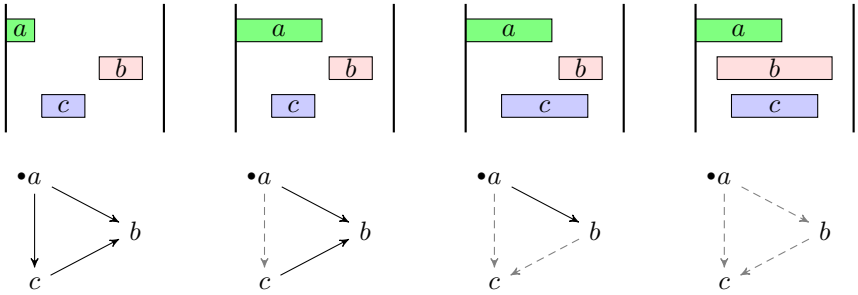


Figure 2. Activity intervals (top) and corresponding iposets (bottom), see Example 2.1. Full arrows indicate precedence order; dashed arrows indicate event order; bullets indicate interfaces.

interval order [14]. In other words, any of the iposets we will encounter admits an *interval representation*: functions b and e from P to real numbers such that $b(x) \leq e(x)$ for all $x \in P$ and $x <_P y \iff e(x) < b(y)$ for all $x, y \in P$. We will only consider interval iposets in this paper and hence omit the qualification "interval". This is *not* a restriction, but rather induced by the semantics. The following property is trivial, but we will make heavy use of it later.

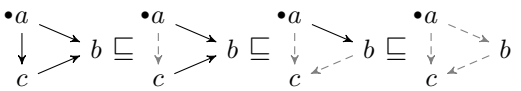
Lemma 2.2. If P is an (interval) iposet and $A \subseteq P$, then the set difference $P - A$ is an (interval) iposet as well.

Iposets may be refined by shortening the activity intervals of events, so that some events stop being concurrent. This corresponds to expanding the precedence relation $<$ (and, potentially, removing event order). The inverse to refinement is called subsumption and defined as follows. For iposets P and Q, we say that Q *subsumes* P (or that P is a *refinement* of Q) and write $P \sqsubseteq Q$ if there exists a bijection $f : P \to Q$ (a *subsumption*) which

- respects interfaces and labels: $f(S_P) = S_Q$, $f(T_P) = T_Q$, and $\lambda_Q \circ f = \lambda_P$;
- reflects precedence: $f(x) <_Q f(y)$ implies $x <_P y$; and
- preserves essential event order: $x \dashrightarrow_P y$ implies $f(x) \dashrightarrow_Q f(y)$ whenever x and y are concurrent (that is, $x \not<_P y$ and $y \not<_P x$).

(Event order is essential for concurrent events, but by transitivity, it also appears between non-concurrent events; subsumptions may ignore such non-essential event order.)

Example 2.3. In Figure 2, there is a sequence of refinements from right to left, each time shortening some activity intervals. Conversely, there is a sequence of subsumptions from left to right:



Interfaces need to be preserved across subsumptions, so in our example, the left endpoint of the a-interval must stay at the boundary.

Iposets and subsumptions form a category. The isomorphisms in that category are invertible subsumptions, and isomorphism classes of iposets are called *ipomsets*. Concretely, an isomorphism $f : P \to Q$ of iposets is a bijection which

- respects interfaces and labels: $f(S_P) = S_Q$, $f(T_P) = T_Q$, and $\lambda_Q \circ f = \lambda_P$;
- respects precedence: $x <_P y \iff f(x) <_Q f(y)$; and
- respects essential event order: $x \dashrightarrow_P y \iff f(x) \dashrightarrow_Q f(y)$ whenever $x \not<_P y$ and $y \not<_P x$.

Isomorphisms between iposets are unique (because of the requirement that all elements be ordered by $<$ or $\dashrightarrow$), hence we may switch freely between ipomsets and concrete representations, see [9] for details. We write $P \cong Q$ if iposets P and Q are isomorphic and let iiPoms denote the set of (interval) ipomsets.

Ipomsets may be *glued*, using a generalisation of the standard serial composition of pomsets [15]. For ipomsets P and Q, their *gluing* $P * Q$ is defined if the targets of P match the sources of Q: $T_P \cong S_Q$. In that case, its carrier set is the quotient $(P \sqcup Q)_{/x \equiv f(x)}$, where $f : T_P \to S_Q$ is the unique isomorphism, the interfaces are $S_{P*Q} = S_P$ and $T_{P*Q} = T_Q$, $\dashrightarrow_{P*Q}$ is the transitive closure of $\dashrightarrow_P \cup \dashrightarrow_Q$, and $x <_{P*Q} y$ if and only if $x <_P y$, or $x <_Q y$, or $x \in P - T_P$ and $y \in Q - S_Q$. We will often omit the "$*$" in gluing compositions. For ipomsets with empty interfaces, $*$ is serial pomset composition; in the general case, matching interface points are glued, see [8, 11] or below for examples.

A *language* is, a priori, a set of ipomsets $L \subseteq \mathsf{iiPoms}$. However, we will assume that languages are closed under refinement (inverse subsumption), so that refinements of any ipomset in L are also in L:

Definition 2.4. A *language* is a subset $L \subseteq \mathsf{iiPoms}$ such that $P \sqsubseteq Q$ and $Q \in L$ imply $P \in L$.

Using interval representations, this means that languages are closed under shortening activity intervals of events. The set of all languages is denoted $\mathscr{L} \subseteq 2^{\mathsf{iiPoms}}$.

For $X \subseteq \mathsf{iiPoms}$ an arbitrary set of ipomsets, we denote by

$$X{\downarrow} = \{P \in \mathsf{iiPoms} \mid \exists Q \in X : P \sqsubseteq Q\}$$

its downward subsumption closure, that is, the smallest language which contains X. Then

$$\mathscr{L} = \{X \subseteq \mathsf{iiPoms} \mid X{\downarrow} = X\}.$$

3. Step Decompositions

An ipomset P is *discrete* if $<_P$ is empty and $\dashrightarrow_P$ total. Conclists are discrete ipomsets with empty interfaces. Discrete ipomsets ${}_UU_U$ are identities for gluing composition and written id_U. A *starter* is an ipomset ${}_{U-A}U_U$, a *terminator* is ${}_UU_{U-A}$; these will be written ${}_A{\uparrow}U$ and $U{\downarrow}_A$, respectively.

Any ipomset can be presented as a gluing of starters and terminators [11, Proposition 21]. (This is related to the fact that a partial order is interval if and only if its antichain order is total, see [14, 16, 17]). Such a presentation we call a *step decomposition*; if starters and terminators are alternating, the decomposition is *sparse*.

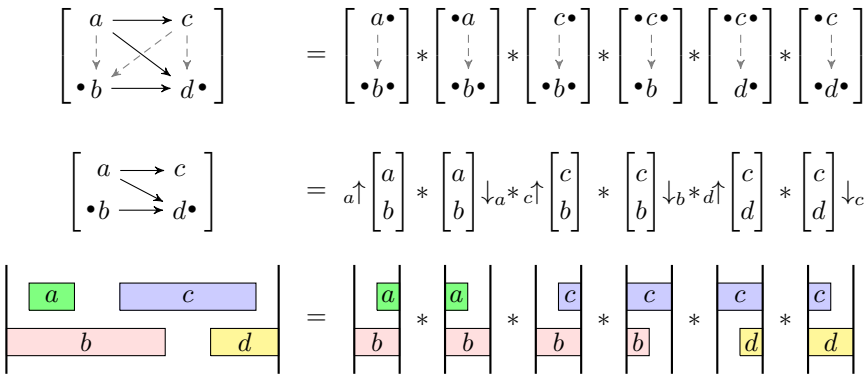


Figure 3. Sparse decomposition of ipomset into starters and terminators.

Example 3.1. Figure 3 shows a sparse decomposition of an ipomset into starters and terminators. The top line shows the graphical representation, in the middle the representation using the notation we have introduced for starters and terminators, and the bottom line shows activity intervals.

We show that sparse step decompositions of ipomsets are unique. For an ipomset P, we denote by $P^m \subseteq P$ the subset of $<$-minimal elements and

$$P^s = \{p \in P \mid \forall\, p' \in P - P^m :\ p < p'\}.$$

That is, P^s contains precisely those minimal elements which have arrows to all non-minimal elements. Clearly, both P^m and P^s are conclists, and $P^s \subseteq P^m \supseteq S_P$. We need a few technical lemmas.

Lemma 3.2. Let P be an ipomset and $A \subseteq U \in \square$.

1. Assume that $U \cong S_P$ and $P' = {}_A{\uparrow}U * P$. Then P' and P are isomorphic as pomsets, $T_P \cong T_{P'}$ and $S_{P'} \cong S_P - A$.
2. Assume that $U - A \cong S_P$ and $P' = U{\downarrow}_A * P$. Then $P' \cong P \cup A$ as sets, and $P \cong P' - A$ as pomsets, $T_P \cong T_{P'}$ and $S_{P'} \cong U$.

Proof:
Simple calculations. $\square$

Consider a presentation $P \cong QR$. From the definition follows that $P^m \cong Q^m$ and $S_P \cong S_Q$ This implies:

Lemma 3.3. Assume that $P \cong QR$ and Q is either a (non-identity) starter or a terminator. Then Q is a starter iff $S_P \subsetneq P^m$, and Q is a terminator iff $S_P = P^m$.

Proof:
We have $P^m \cong Q = Q^m$ and $S_P \cong S_Q$. But Q is a terminator if and only if $S_Q = Q$, and a (non-identity) starter if and only if $S_Q \subsetneq Q$. $\square$

Lemma 3.4. Assume that $P \cong QQ'R$.

1. If Q is a non-identity starter and Q' is a non-identity terminator, then $Q \cong {}_{P^m - S_P}\!\uparrow P^m$.
2. If Q is a non-identity terminator and Q' is a non-identity starter, then $Q \cong P^m{\downarrow}_{P^s}$.

Proof:
Consider the first case. Then P and $Q'R$ are isomorphic as pomsets, and

$$Q = T_Q \cong S_{Q'R} \overset{\text{Lemma 3.3}}{=} (Q'R)^m \overset{\text{Lemma 3.2}}{\cong} P^m.$$

Equality $S_Q = S_P$ follows immediately from the definition.

In the second case, we have $Q = S_Q \cong S_P \overset{\text{Lemma 3.3}}{=} P^m$, and $Q'R \cong P - (Q - T_Q)$ as pomsets (Lemma 3.2). By Lemma 3.3 we have

$$P^m \cap (Q'R) = Q \cap (Q'R) = T_Q \cong S_{Q'R} \overset{\text{Lemma 3.3}}{\subsetneq} (Q'R)^m.$$

Hence there exists an element $p \in Q'R$ that is minimal in $Q'R$ but not in P. For every $p' \in P^s$ we have $p' < p$ and, therefore, $p' \notin Q'R$. As a consequence, $P^s \subseteq P - (Q'R) = Q - T_Q$ (Lemma 3.2).

On the other hand, if $p' \in P^m - P^s$, then there exists $p \in P - P^m = P - Q$ such that $p' \not< p$. Thus, p' must belong to T_Q. □

Proposition 3.5. Every ipomset P has a unique sparse step decomposition.

Proof:
Let $P = P_1 * \cdots * P_n = Q_1 * \cdots * Q_m$ be two sparse presentations. If $n = 1$, then $m = 1$ and equality follows trivially, so assume $n, m \geq 2$ and write $P_2 * \cdots * P_n = P'$ and $Q_2 * \cdots * Q_m = Q'$.

Assume first that P_1 is a starter. By Lemma 3.4, $P_1 \cong {}_{P^m - S_P}\!\uparrow P^m$. By Lemma 3.2, $S_P \cong S_{P'} - (P^m - S_P)$. Hence $S_{P'} \cong P^m$, implying $S_P \subsetneq P^m$. By Lemma 3.3, Q_1 is a starter. By Lemma 3.4, $Q_1 \cong {}_{P^m - S_P}\!\uparrow P^m$. Thus $P_1 \cong Q_1$, and we may proceed inductively with $P' = Q'$.

Now assume instead that P_1 is a terminator. By Lemma 3.4, $P_1 \cong P^m{\downarrow}_{P^s}$. By Lemma 3.2, $S_P \cong P^m$. By Lemma 3.3, Q_1 is a terminator. By Lemma 3.4, $Q_1 \cong P^m{\downarrow}_{P^s}$. Thus $P_1 \cong Q_1$, and we may proceed inductively with $P' = Q'$. □

4. Higher-Dimensional Automata and Their Languages

An HDA is a collection of *cells* which are connected according to specified *face maps*. Each cell has an associated list of *labelled events* which are interpreted as being executed in that cell, and the face maps may terminate some events or, inversely, indicate cells in which some of the current events were not yet started. Additionally, some cells are designated *start* cells and some others *accept* cells; computations of an HDA begin in a start cell and proceed by starting and terminating events until they reach an accept cell.

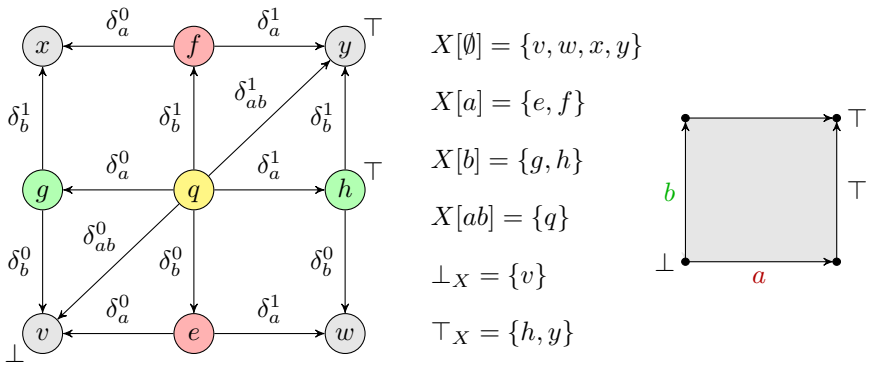


Figure 4. A two-dimensional HDA X on $\Sigma = \{a, b\}$, see Example 4.2.

4.1. Precubical sets and HDAs

To make the above precise, let $\square$ denote the set of conclists. A *precubical set* consists of a set of cells X together with a mapping $\mathsf{ev} : X \to \square$ which to every cell assigns its list of active events. For a conclist U we write $X[U] = \{x \in X \mid \mathsf{ev}(x) = U\}$ for the cells of type U. Further, for every $U \in \square$ and subset $A \subseteq U$ there are *face maps* $\delta^0_A, \delta^1_A : X[U] \to X[U - A]$. The *upper* face maps δ^1_A terminate the events in A, whereas the *lower* face maps δ^0_A "unstart" these events: they map cells $x \in X[U]$ to cells $\delta^0_A(x) \in X[U - A]$ where the events in A are not yet active.

If $A, B \subseteq U$ are disjoint, then the order in which events in A and B are terminated or unstarted should not matter, so we require that $\delta^\nu_A \delta^\mu_B = \delta^\mu_B \delta^\nu_A$ for $\nu, \mu \in \{0, 1\}$: the *precubical identities*. A *higher-dimensional automaton* (*HDA*) is a precubical set X together with subsets $\bot_X, \top_X \subseteq X$ of *start* and *accept* cells. For a precubical set X and subsets $Y, Z \subseteq X$ we denote by X^Z_Y the HDA with precubical set X, start cells Y and accept cells Z. We do *not* generally assume that precubical sets or HDAs are finite. The *dimension* of an HDA X is $\dim(X) = \sup\{|\mathsf{ev}(x)| \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$.

Example 4.1. One-dimensional HDAs X are standard automata. Cells in $X[\emptyset]$ are states, cells in $X[a]$ for $a \in \Sigma$ are a-labelled transitions. Face maps δ^0_a and δ^1_a attach source and target states to transitions. In contrast to ordinary automata we allow start and accept *transitions* instead of merely states, so languages of such automata may contain not only words but also "words with interfaces". In any case, at most one event is active at any point in time, so the event order is unnecessary.

Example 4.2. Figure 4 shows a two-dimensional HDA X both as a combinatorial object (left) and in a more geometric realisation (right). We write isomorphism classes of conclists as lists of labels and omit the set braces in $\delta^0_{\{a\}}$ etc. X has four zero-dimensional cells, or states, displayed in grey on the left; four one-dimensional transitions, two labelled a and displayed in red and two labelled b and shown in green; and one two-dimensional cell displayed in yellow.

An *HDA-map* between HDAs X and Y is a function $f : X \to Y$ that preserves structure: types of cells ($\mathsf{ev}_Y \circ f = \mathsf{ev}_X$), face maps ($f(\delta^\nu_A(x)) = \delta^\nu_A(f(x))$) and start/accept cells ($f(\bot_X) \subseteq \bot_Y$,

$f(\top_X) \subseteq \top_Y$). Similarly, a precubical map is a function that preserves the first two of these three. HDAs and HDA-maps form a category, as do precubical sets and precubical maps.

4.2. Paths and their labels

Computations of HDAs are paths: sequences of cells connected by face maps. A *path* in X is, thus, a sequence

$$\alpha = (x_0, \varphi_1, x_1, \dots, x_{n-1}, \varphi_n, x_n), \tag{1}$$

where the x_i are cells of X and the φ_i indicate types of face maps: for every i, $(x_{i-1}, \varphi_i, x_i)$ is either

- $(\delta^0_A(x_i), \nearrow^A, x_i)$ for $A \subseteq \mathsf{ev}(x_i)$ (an *upstep*)
- or $(x_{i-1}, \searrow_B, \delta^1_B(x_{i-1}))$ for $B \subseteq \mathsf{ev}(x_{i-1})$ (a *downstep*).

Upsteps start events in A while downsteps terminate events in B. The *source* and *target* of α as in (1) are $\mathsf{src}(\alpha) = x_0$ and $\mathsf{tgt}(\alpha) = x_n$.

The set of all paths in X starting at $Y \subseteq X$ and terminating in $Z \subseteq X$ is denoted by $\mathsf{Path}(X)^Z_Y$; we write $\mathsf{Path}(X)_Y = \mathsf{Path}(X)^X_Y$, $\mathsf{Path}(X)^Z = \mathsf{Path}(X)^Z_X$, and $\mathsf{Path}(X) = \mathsf{Path}(X)^X_X$. A path α is *accepting* if $\mathsf{src}(\alpha) \in \bot_X$ and $\mathsf{tgt}(\alpha) \in \top_X$. Paths α and β may be concatenated if $\mathsf{tgt}(\alpha) = \mathsf{src}(\beta)$; their concatenation is written $\alpha * \beta$, and we omit the "$*$" in concatenations if convenient.

Path equivalence is the congruence $\simeq$ generated by $(z \nearrow^A y \nearrow^B x) \simeq (z \nearrow^{A\cup B} x)$, $(x \searrow_A y \searrow_B z) \simeq (x \searrow_{A\cup B} z)$, and $\gamma\alpha\delta \simeq \gamma\beta\delta$ whenever $\alpha \simeq \beta$. Intuitively, this relation allows to assemble subsequent upsteps or downsteps into one "bigger" step. A path is *sparse* if its upsteps and downsteps are alternating, so that no more such assembling may take place. Every equivalence class of paths contains a unique sparse path.

Example 4.3. In one-dimensional HDAs, paths are sequences of transitions connected at states. Path equivalence is a trivial relation, and all paths are sparse.

Example 4.4. The HDA X of Figure 4 admits five sparse accepting paths:

$$v \nearrow^a e \searrow_a w \nearrow^b h, \qquad v \nearrow^a e \searrow_a w \nearrow^b h \searrow_b y,$$
$$v \nearrow^{ab} q \searrow_a h, \qquad v \nearrow^{ab} q \searrow_{ab} y, \qquad v \nearrow^b g \searrow_b x \nearrow^a f \searrow_a y.$$

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

- If $\alpha = (x)$, then $\mathsf{ev}(\alpha) = \mathsf{id}_{\mathsf{ev}(x)}$.
- If $\alpha = (y \nearrow^A x)$, then $\mathsf{ev}(\alpha) = {}_A\!\uparrow\mathsf{ev}(x)$.
- If $\alpha = (x \searrow_B y)$, then $\mathsf{ev}(\alpha) = \mathsf{ev}(x)\!\downarrow_B$.
- If $\alpha = \alpha_1 * \dots * \alpha_n$ is a concatenation, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \dots * \mathsf{ev}(\alpha_n)$.

[9, Lemma 8] shows that $\alpha \simeq \beta$ implies $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$. Further, if $\alpha = \alpha_1 * \dots * \alpha_n$ is a sparse path, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\alpha_1) * \dots * \mathsf{ev}(\alpha_n)$ is a sparse step decomposition.

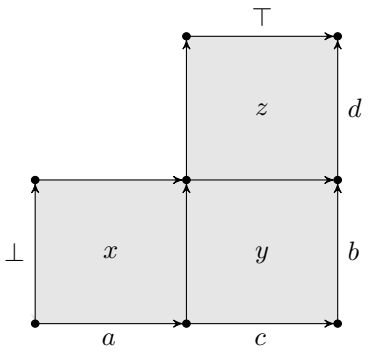


Figure 5. HDA Y consisting of three squares glued along common faces.

Example 4.5. Event ipomsets of paths in one-dimensional HDAs are words, possibly with interfaces. Sparse step decompositions of words are obtained by splitting symbols into starts and terminations, for example, $\bullet ab = \bullet a * b\bullet * \bullet b$.

Example 4.6. The event ipomsets of the five sparse accepting paths in the HDA X of Figure 4 are $ab\bullet$, ab, $\left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$, and ba. Figure 5 shows another HDA, which admits an accepting path

$$(\delta^0_a x \nearrow^a x \searrow_a \delta^1_a x \nearrow^c y \searrow_b \delta^1_b y \nearrow^d z \searrow_d \delta^1_d z).$$

Its event ipomset is precisely the ipomset of Figure 3, with the indicated sparse step decomposition arising from the sparse presentation above.

4.3. Languages of HDAs

The *language* of an HDA X is

$$\mathsf{Lang}(X) = \{\mathsf{ev}(\alpha) \mid \alpha \text{ accepting path in } X\}.$$

[9, Proposition 10] shows that languages of HDAs are sets of ipomsets which are closed under subsumption, *i.e.*, languages in the sense of Definition 2.4.

A language is *regular* if it is the language of a finite HDA.

Example 4.7. The languages of our example HDAs are

$$\mathsf{Lang}(X) = \left\{ \left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] \right\}{\downarrow} = \left\{ \left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right], ab\bullet, \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], ab, ba \right\}$$

and

$$\mathsf{Lang}(Y) = \left\{ \left[\begin{tikzcd} a \arrow[r] \arrow[dr] & c\bullet \\ \bullet b \arrow[r] & d \end{tikzcd} \right] \right\}{\downarrow}.$$

We say that a cell $x \in X$ in an HDA X is

- *accessible* if $\mathsf{Path}(X)^x_\bot \neq \emptyset$, *i.e.*, x can be reached by a path from a start cell;
- *coaccessible* if $\mathsf{Path}(X)^\top_x \neq \emptyset$, *i.e.*, there is a path from x to an accept cell;
- *essential* if it is both accessible and coaccessible.

A path is *essential* if its source and target cells are essential. This implies that all its cells are essential. Segments of accepting paths are always essential.

The set of essential cells of X is denoted by $\mathsf{ess}(X)$; this is not necessarily a sub-HDA of X given that faces of essential cells may be non-essential. For example, all bottom cells of the HDA Y in Figure 5 are inaccessible and hence non-essential.

Lemma 4.8. Let X be an HDA. There exists a smallest sub-HDA $X^{\mathsf{ess}} \subseteq X$ that contains all essential cells, and $\mathsf{Lang}(X^{\mathsf{ess}}) = \mathsf{Lang}(X)$. If $\mathsf{ess}(X)$ is finite, then X^{ess} is also finite.

Proof:
The set of all faces of essential cells

$$X^{\mathsf{ess}} = \{\delta^0_A \delta^1_B(x) \mid x \in \mathsf{ess}(X),\ A, B \subseteq \mathsf{ev}(x),\ A \cap B = \emptyset\}$$

is a sub-HDA of X. Clearly every sub-HDA of X that contains $\mathsf{ess}(X)$ must also contain X^{ess}. Since all accepting paths are essential, $\mathsf{Lang}(X^{\mathsf{ess}}) = \mathsf{Lang}(X)$. If $|\mathsf{ess}(X)| = n$ and $|\mathsf{ev}(x)| \leq d$ for all $x \in \mathsf{ess}(X)$, then $|X^{\mathsf{ess}}| \leq n \cdot 3^d$, since a cell of dimension $\leq d$ has at most 3^d faces. □

4.4. Track objects

Track objects, introduced in [8], provide a mapping from ipomsets to HDAs and are a powerful tool for reasoning about languages. Below we adapt the definition from [8, Section 5.3].

Definition 4.9. The *track object* of an ipomset P is the HDA $\square^P$ defined as follows:

- $\square^P$ is the set of all functions $x : P \to \{0, *, 1\}$ such that

$$p < q \implies (x(p), x(q)) \in \{(0,0), (*,0), (1,0), (1,*), (1,1)\}.$$

- For $x \in \square^P$, $\mathsf{ev}(x) = x^{-1}(*)$ (the condition above ensures that $x^{-1}(*)$ is discrete);
- For $x \in \square^P$, $\nu \in \{0,1\}$ and $A \subseteq \mathsf{ev}(x)$,

$$\delta^\nu_A(x)(p) = \begin{cases} \nu & \text{for } p \in A, \\ x(p) & \text{for } p \notin A; \end{cases}$$

- $\bot_{\square^P} = \{c^P_\bot\}$ and $\top_{\square^P} = \{c^\top_P\}$, where

$$c^P_\bot(p) = \begin{cases} * & \text{if } p \in S_P, \\ 0 & \text{if } p \notin S_P, \end{cases} \qquad c^\top_P(p) = \begin{cases} * & \text{if } p \in T_P, \\ 1 & \text{if } p \notin T_P; \end{cases}$$

We list some properties of track objects needed later.

Lemma 4.10. Let X be an HDA, $x, y \in X$ and $P \in \mathsf{iiPoms}$. The following conditions are equivalent:

1. There exists a path $\alpha \in \mathsf{Path}(X)_x^y$ such that $\mathsf{ev}(\alpha) = P$.
2. There is an HDA-map $f : \square^P \to X_x^y$ (*i.e.*, $f(c_\bot^P) = x$ and $f(c_P^\top) = y$).

Proof:
This is an immediate consequence of [8, Proposition 89]. □

Lemma 4.11. If $P, Q \in \mathsf{iiPoms}$ are such that $P \sqsubseteq Q$, then there exists an HDA-map $\square^P \to \square^Q$.

Proof:
This is [8, Lemma 63]. □

Lemma 4.12. Let X be an HDA, $x, y \in X$, $\beta \in \mathsf{Path}(X)_x^y$ and $P \sqsubseteq Q = \mathsf{ev}(\beta)$. Then there exists $\alpha \in \mathsf{Path}(X)_x^y$ such that $\mathsf{ev}(\alpha) = P$.

Proof:
This follows immediately from Lemmas 4.10 and 4.11. □

Lemma 4.13. Let X be an HDA, $x, y \in X$ and $\gamma \in \mathsf{Path}(X)_x^y$. Assume that $\mathsf{ev}(\gamma) = P * Q$ for ipomsets P and Q. Then there exist paths $\alpha \in \mathsf{Path}(X)_x$ and $\beta \in \mathsf{Path}(X)^y$ such that $\mathsf{ev}(\alpha) = P$, $\mathsf{ev}(\beta) = Q$ and $\mathsf{tgt}(\alpha) = \mathsf{src}(\beta)$.

Proof:
By Lemma 4.10, there is an HDA-map $f : \square^{PQ} \to X_x^y$. By [8, Lemma 65], there exist precubical maps $j_P : \square^P \to \square^{PQ}$, $j_Q : \square^Q \to \square^{PQ}$ such that $j_P(c_\bot^P) = c_\bot^{PQ}$, $j_P(c_P^\top) = j_Q(c_\bot^Q)$ and $j_Q(c_Q^\top) = c_{PQ}^\top$. Let $z = f(j_P(c_\bot^P))$, then $f \circ j_P : \square^P \to X_x^z$ and $f \circ j_Q : \square^Q \to X_z^y$ are HDA-maps, and by applying Lemma 4.10 again to j_P and j_Q we obtain α and β. □

5. Myhill-Nerode Theorem

The *prefix quotient* of a language $L \in \mathscr{L}$ by an ipomset P is the language

$$P \setminus L = \{Q \in \mathsf{iiPoms} \mid PQ \in L\}.$$

Similarly, the *suffix quotient* of L by P is $L/P = \{Q \in \mathsf{iiPoms} \mid QP \in L\}$. Denote

$$\mathsf{suff}(L) = \{P \setminus L \mid P \in \mathsf{iiPoms}\}, \qquad \mathsf{pref}(L) = \{L/P \mid P \in \mathsf{iiPoms}\}.$$

We record the following property of quotient languages.

Lemma 5.1. If L is a language and $P \sqsubseteq Q$, then $Q \setminus L \subseteq P \setminus L$.

Proof:
If $P \sqsubseteq Q$, then $PR \sqsubseteq QR$. Thus,

$$R \in Q \setminus L \iff QR \in L \implies PR \in L \iff R \in P \setminus L.$$

□

The main goal of this section is to show the following.

Theorem 5.2. For a language $L \in \mathscr{L}$ the following conditions are equivalent.

(a) L is regular.

(b) The set $\mathsf{suff}(L) \subseteq \mathscr{L}$ is finite.

(c) The set $\mathsf{pref}(L) \subseteq \mathscr{L}$ is finite.

We prove only the equivalence between (a) and (b); equivalence between (a) and (c) is symmetric. First we prove the implication (a) $\implies$ (b). Let X be an HDA with $\mathsf{Lang}(X) = L$. For $x \in X$ define languages $\mathsf{Pre}(x) = \mathsf{Lang}(X^x_\bot)$ and $\mathsf{Post}(x) = \mathsf{Lang}(X^\top_x)$.

Lemma 5.3. For every $P \in \mathsf{iiPoms}$, $P \setminus L = \bigcup\{\mathsf{Post}(x) \mid x \in X,\ P \in \mathsf{Pre}(x)\}$.

Proof:
We have

$$\begin{aligned} Q \in P \setminus L \iff PQ \in L &\overset{\text{Lem. 4.10}}{\iff} \exists\, f : \square^{PQ} \to X = X^\top_\bot \\ &\overset{\text{Lem. 4.13}}{\iff} \exists\, x \in X, g : \square^P \to X^x_\bot,\ h : \square^Q \to X^\top_x \\ &\overset{\text{Lem. 4.10}}{\iff} \exists\, x \in X : P \in \mathsf{Lang}(X^x_\bot),\ Q \in \mathsf{Lang}(X^\top_x) \\ &\iff \exists\, x \in X : P \in \mathsf{Pre}(x),\ Q \in \mathsf{Post}(x). \end{aligned}$$

The last condition says that Q belongs to the right-hand side of the equation. □

Proof of Theorem 5.2, (a) $\implies$ (b)**:**
The family of languages $\{P \setminus L \mid P \in \mathsf{iiPoms}\}$ is a subfamily of $\{\bigcup_{x \in Y} \mathsf{Post}(x) \mid Y \subseteq X\}$ which is finite. □

5.1. HDA construction

Now we show that (b) implies (a). Fix a language $L \in \mathscr{L}$, with $\mathsf{suff}(L)$ finite or infinite. We will construct an HDA $\mathsf{MN}(L)$ that recognises L and show that if $\mathsf{suff}(L)$ is finite, then the essential part $\mathsf{MN}(L)^{\mathsf{ess}}$ is finite. The cells of $\mathsf{MN}(L)$ are equivalence classes of ipomsets under a relation $\approx_L$ induced by L which we will introduce below. The relation $\approx_L$ is defined using prefix quotients, but needs to be stronger than prefix quotient equivalence. This is because events may be concurrent and because ipomsets have interfaces. We give examples just after the construction.

For an ipomset ${}_SP_T$ define its *(target) signature* to be the starter $\mathsf{fin}(P) = {}_{T-S}{\uparrow}T$. Thus $\mathsf{fin}(P)$ collects all target events of P, and its source interface contains those events that are also in the source interface of P. We also write $\mathsf{rfin}(P) = T - S \subseteq \mathsf{fin}(P)$: the set of all target events of P that are not source events. An important property is that removing elements of $\mathsf{rfin}(P)$ does not change the source interface of P. For example,

$$\mathsf{fin}\left(\begin{bmatrix}{}^{\bullet}a^{\bullet}\\ {}^{\bullet}a\\ c^{\bullet}\end{bmatrix}\right) = \begin{bmatrix}{}^{\bullet}a^{\bullet}\\ c^{\bullet}\end{bmatrix}, \quad \mathsf{fin}\left(\begin{bmatrix}{}^{\bullet}ac^{\bullet}\\ {}^{\bullet}b\,{}^{\bullet}\end{bmatrix}\right) = \begin{bmatrix}c^{\bullet}\\ {}^{\bullet}b^{\bullet}\end{bmatrix}, \quad \mathsf{fin}\left(\begin{bmatrix}ac^{\bullet}\\ b\,{}^{\bullet}\end{bmatrix}\right) = \begin{bmatrix}c^{\bullet}\\ b^{\bullet}\end{bmatrix};$$

rfin is $\{c\}$ in the first two examples and equal to $\left[\begin{smallmatrix}c\\ b\end{smallmatrix}\right]$ in the last.

We define two equivalence relations on iiPoms induced by L:

- Ipomsets P and Q are *weakly equivalent* ($P \sim_L Q$) if $\mathsf{fin}(P) \cong \mathsf{fin}(Q)$ and $P \setminus L = Q \setminus L$. Obviously, $P \sim_L Q$ implies $T_P \cong T_Q$ and $\mathsf{rfin}(P) \cong \mathsf{rfin}(Q)$.
- Ipomsets P and Q are *strongly equivalent* ($P \approx_L Q$) if $P \sim_L Q$ and for all $A \subseteq \mathsf{rfin}(P) \cong \mathsf{rfin}(Q)$ we have $(P - A) \setminus L = (Q - A) \setminus L$.

Evidently $P \approx_L Q$ implies $P \sim_L Q$, but the inverse does not always hold. We explain in Example 5.6 below why $\approx_L$, and not $\sim_L$, is the proper relation to use for constructing $\mathsf{MN}(L)$.

Lemma 5.4. If $P \approx_L Q$, then $P - A \approx_L Q - A$ for all $A \subseteq \mathsf{rfin}(P) \cong \mathsf{rfin}(Q)$.

Proof:
For every A we have $(P - A) \setminus L = (Q - A) \setminus L$, and

$$\mathsf{fin}(P - A) = \mathsf{fin}(P) - A \cong \mathsf{fin}(Q) - A = \mathsf{fin}(Q - A),$$

Thus, $P - A \sim_L Q - A$. Further, for every $B \subseteq \mathsf{rfin}(P - A) \cong \mathsf{rfin}(Q - A)$,

$$((P - A) - B) \setminus L = (P - (A \cup B)) \setminus L = (Q - (A \cup B)) \setminus L = ((Q - A) - B) \setminus L,$$

which shows that $P - A \approx_L Q - A$. □

Now define an HDA $\mathsf{MN}(L)$ as follows. For $U \in \square$, write $\mathsf{iiPoms}_U = \{P \in \mathsf{iiPoms} \mid T_P \cong U\}$ and let

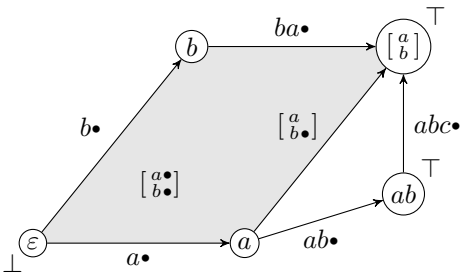
$$\mathsf{MN}(L)[U] = \mathsf{iiPoms}_U/{\approx_L} \cup \{w_U\},$$

where the w_U are new *subsidiary* cells which are introduced solely to define some lower faces. (They will not affect the language of $\mathsf{MN}(L)$).

The $\approx_L$-equivalence class of P will be denoted by $\langle P\rangle$ (but often just P in examples). Face maps are defined as follows, for $A \subseteq U \in \square$ and $P \in \mathsf{iiPoms}_U$:

$$\delta^0_A(\langle P\rangle) = \begin{cases}\langle P - A\rangle & \text{if } A \subseteq \mathsf{rfin}(P),\\ w_{U-A} & \text{otherwise,}\end{cases} \qquad \delta^1_A(\langle P\rangle) = \langle P * U{\downarrow}_A\rangle, \tag{2}$$

$$\delta^0_A(w_U) = \delta^1_A(w_U) = w_{U-A}.$$



$\mathsf{MN}(L)[\emptyset]$

P	$P \setminus L$
ε	L
a	$\{b, bc\}$
b	$\{a\}$
ab	$\{\varepsilon, c\}$
$\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$	$\{\varepsilon\}$

$\mathsf{MN}(L)[a]$

P	$P \setminus L$
$a\bullet$	$\{\left[\begin{smallmatrix} \bullet a \\ b \end{smallmatrix}\right], \bullet ab, \bullet abc\}$
$ba\bullet$	$\{\bullet a\}$

$\mathsf{MN}(L)[b]$

P	$P \setminus L$
$b\bullet$	$\{\left[\begin{smallmatrix} a \\ \bullet b \end{smallmatrix}\right], \bullet ba\}$
$ab\bullet$	$\{\bullet b, \bullet bc\}$
$\left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right]$	$\{\bullet b\}$

$\mathsf{MN}(L)[c]$

P	$P \setminus L$
$abc\bullet$	$\{\bullet c\}$

$\mathsf{MN}(L)[\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]]$

P	$P \setminus L$
$\left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right]$	$\{\left[\begin{smallmatrix} \bullet a \\ \bullet b \end{smallmatrix}\right]\}$

Figure 6. HDA $\mathsf{MN}(L)$ of Example 5.5, showing names of cells instead of labels (labels are target interfaces of names). Tables show essential cells together with prefix quotients.

In other words, if A has no source events of P, then δ^0_A removes A from P (the source interface of P is unchanged). If A contains any source event, then $\delta^0_A(P)$ is a subsidiary cell.

Finally, start and accept cells are given by

$$\bot_{\mathsf{MN}(L)} = \{\langle \mathsf{id}_U \rangle\}_{U \in \square}, \qquad \top_{\mathsf{MN}(L)} = \{\langle P \rangle \mid P \in L\}.$$

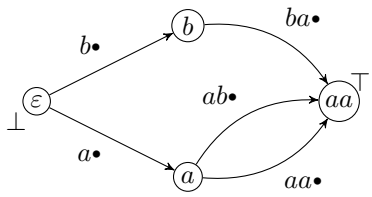
The cells $\langle P \rangle$ will be called *regular*. They are $\approx_L$-equivalence classes of ipomsets, lower face maps unstart events, and upper face maps terminate events. All faces of subsidiary cells w_U are subsidiary, and upper faces of regular cells are regular. Below we present several examples, in which we show only the essential part $\mathsf{MN}(L)^{\mathsf{ess}}$ of $\mathsf{MN}(L)$.

Example 5.5. Let $L = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], abc\}{\downarrow} = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], ab, ba, abc\}$. Figure 6 shows the HDA $\mathsf{MN}(L)^{\mathsf{ess}}$ together with a list of essential cells of $M(L)$ and their prefix quotients in L. Note that the state $\langle a \rangle$ has *two* outgoing b-labelled edges: $\langle ab\bullet \rangle$ and $\langle \left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right] \rangle$. The generating ipomsets have different prefix quotients because of $\{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], abc\} \subseteq L$, but the same lower face $\langle a \rangle$. (Note that $\langle ba\bullet \rangle = \langle \left[\begin{smallmatrix} a\bullet \\ b \end{smallmatrix}\right] \rangle$.)

Intuitively, $\mathsf{MN}(L)^{\mathsf{ess}}$ is thus *non-deterministic*; this is interesting because the standard Myhill-Nerode theorem for finite automata constructs deterministic automata. We will give a precise definition of determinism for HDAs in the next section and show in Example 6.5 that no deterministic HDA X exists with $\mathsf{Lang}(X) = L$.

Example 5.6. Here we explain why we need to use $\approx_L$-equivalence classes and not $\sim_L$-equivalence classes. The example is one-dimensional, which means that it applies to standard finite automata. The reason one does not see the problem in the standard Myhill-Nerode construction for finite automata is that this operates only on states and not on transitions.

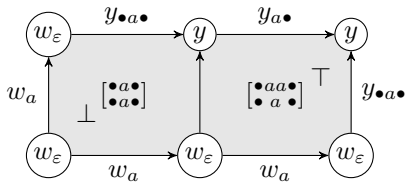
Let $L = \{aa, ab, ba\}$, then $\mathsf{MN}(L)^{\mathsf{ess}}$ is as below.



We have $aa\bullet \setminus L = ba\bullet \setminus L = \{\bullet a\}$, thus $aa\bullet \sim_L ba\bullet$. Yet $aa\bullet$ and $ba\bullet$ are not strongly equivalent, because $a \setminus L = \{a, b\} \neq \{a\} = b \setminus L$. This provides an example of weakly equivalent ipomsets whose lower faces are not weakly equivalent and shows why we cannot use $\sim_L$ to construct $\mathsf{MN}(L)$.

Remark 5.7. As the previous example indicates, if L is one-dimensional and all words in L have empty interfaces, then $\mathsf{ess}(\mathsf{MN}(L))$ is the standard Myhill-Nerode finite automaton for L.

Example 5.8. The language $L = \{[\begin{smallmatrix}\bullet & aa & \bullet\\ \bullet & a & \bullet\end{smallmatrix}]\}$ is recognised by the HDA $\mathsf{MN}(L)^{\mathsf{ess}}$ below:



Cells with the same names are identified. Here we see subsidiary cells w_ε and w_a, and regular cells that are not coaccessible (denoted by y indexed with their signature). The middle vertical edge is $\langle[\begin{smallmatrix}\bullet & a & \bullet\\ \bullet & a & \bullet\end{smallmatrix}]\rangle$, $y_{\bullet a\bullet} = \langle[\begin{smallmatrix}\bullet & a & \bullet\\ \bullet & a & \end{smallmatrix}]\rangle = \langle[\begin{smallmatrix}\bullet & aa & \\ \bullet & a & \bullet\end{smallmatrix}]\rangle$, $y_{a\bullet} = \langle[\begin{smallmatrix}\bullet & aa & \bullet\\ \bullet & a & \end{smallmatrix}]\rangle$, and $y = \langle[\begin{smallmatrix}\bullet & a & \\ \bullet & a & \end{smallmatrix}]\rangle = \langle[\begin{smallmatrix}\bullet & aa & \\ \bullet & a & \end{smallmatrix}]\rangle$.

5.2. $\mathsf{MN}(L)$ is well-defined

We need to show that $\mathsf{MN}(L)$ is well-defined, *i.e.*, that the formulas (2) do not depend on the choice of a representative in $\langle P\rangle$ and that the precubical identities are satisfied.

Lemma 5.9. Let P, Q and R be ipomsets with $T_P = T_Q = S_R$. Then

$$P \setminus L \subseteq Q \setminus L \implies (PR) \setminus L \subseteq (QR) \setminus L.$$

In particular, $P \setminus L = Q \setminus L$ implies $(PR) \setminus L = (QR) \setminus L$.

Proof:
For $N \in \mathsf{iiPoms}$ we have

$$\begin{aligned} N \in (PR) \setminus L \iff PRN \in L \iff RN \in P \setminus L \\ \implies RN \in Q \setminus L \iff QRN \in L \iff N \in (QR) \setminus L. \end{aligned}$$

□

The next lemma shows an operation to "add order" to an ipomset P. This is done by first removing some points $A \subseteq T_P$ and then adding them back in, forcing arrows from all other points in P. The result is obviously subsumed by P.

Lemma 5.10. For $P \in \mathsf{iiPoms}$ and $A \subseteq \mathsf{rfin}(P)$, $(P - A) * {}_A{\uparrow}T_P \sqsubseteq P$.

The next two lemmas, whose proofs are again obvious, state that events may be unstarted or terminated in any order.

Lemma 5.11. Let U be a conclist and $A, B \subseteq U$ disjoint subsets. Then

$$U{\downarrow}_B * (U - B){\downarrow}_A = U{\downarrow}_{A \cup B} = U{\downarrow}_A * (U - A){\downarrow}_B.$$

Lemma 5.12. Let $P \in \mathsf{iiPoms}$ and $A, B \subseteq T_P$ disjoint subsets. Then

$$(P * T_P{\downarrow}_B) - A = (P - A) * (T_P - A){\downarrow}_B.$$

Lemma 5.13. Assume that $P \approx_L Q$ for $P, Q \in \mathsf{iiPoms}_U$. Then $P * U{\downarrow}_B \approx_L Q * U{\downarrow}_B$ for every $B \subseteq U$.

Proof:
Obviously $\mathsf{fin}(P * U{\downarrow}_B) = \mathsf{fin}(P) - B \cong \mathsf{fin}(Q) - B = \mathsf{fin}(Q * U{\downarrow}_B)$. For every $A \subseteq \mathsf{rfin}(P) - B \simeq \mathsf{rfin}(Q) - B$ we have

$$((P - A) * (U - A){\downarrow}_B) \setminus L = ((Q - A) * (U - A){\downarrow}_B) \setminus L$$

by assumption and Lemma 5.9. But $(P * U{\downarrow}_B) - A = (P - A) * (U - A){\downarrow}_B$ and $(Q * U{\downarrow}_B) - A = (Q - A) * (U - A){\downarrow}_B$ by Lemma 5.12. □

Proposition 5.14. $\mathsf{MN}(L)$ is a well-defined HDA.

Proof:
The face maps are well-defined: for δ^0_A this follows from Lemma 5.4, for δ^1_B from Lemma 5.13. The precubical identities $\delta^\nu_A \delta^\mu_B = \delta^\mu_B \delta^\nu_A$ are clear for $\nu = \mu = 0$, follow from Lemma 5.11 for $\nu = \mu = 1$, and from Lemma 5.12 for $\{\nu, \mu\} = \{0, 1\}$. □

5.3. Paths and essential cells of $\mathsf{MN}(L)$

The next lemma provides paths in $\mathsf{MN}(L)$.

Lemma 5.15. For every $N, P \in \mathsf{iiPoms}$ such that $T_N \cong S_P$ there exists a path $\alpha \in \mathsf{Path}(\mathsf{MN}(L))^{\langle NP \rangle}_{\langle N \rangle}$ such that $\mathsf{ev}(\alpha) = P$.

Proof:
Choose a decomposition $P = Q_1 * \cdots * Q_n$ into starters and terminators. Denote $U_k = T_{Q_k} = S_{Q_{k+1}}$ and define

$$x_k = \langle N * Q_1 * \cdots * Q_k \rangle, \qquad \varphi_k = \begin{cases} \nearrow^A & \text{if } Q_k = {}_A\!\uparrow\!U_k, \\ \searrow_B & \text{if } Q_k = U_{k-1}\!\downarrow\!_B \end{cases}$$

for $k = 1, \dots, n$. If $\varphi_k = \nearrow^A$ and $Q_k = {}_A\!\uparrow\!U_k$, then

$$\delta^0_A(x_k) = \langle N * Q_1 * \cdots * Q_{k-1} * {}_A\!\uparrow\!U_k - A \rangle = \langle N * Q_1 * \cdots * Q_{k-1} * \mathsf{id}_{U_k - A} \rangle = x_{k-1}.$$

If $\varphi_k = \searrow_B$ and $Q_k = U_{k-1}\!\downarrow\!_B$, then

$$\delta^1_B(x_{k-1}) = \langle N * Q_1 * \cdots * Q_{k-1} * U_{k-1}\!\downarrow\!_B \rangle = x_k.$$

Thus, $\alpha = (x_0, \varphi_1, x_1, \dots, \varphi_n, x_n)$ is a path with $\mathsf{ev}(\alpha) = P$, $\mathsf{src}(\alpha) = \langle N \rangle$ and $\mathsf{tgt}(\alpha) = \langle N * P \rangle$. □

Our goal is now to describe essential cells of $\mathsf{MN}(L)$.

Lemma 5.16. All regular cells of $\mathsf{MN}(L)$ are accessible. If $P \setminus L \neq \emptyset$, then $\langle P \rangle$ is coaccessible.

Proof:
Both claims follow from Lemma 5.15. For every P there exists a path from $\langle \mathsf{id}_{S_P} \rangle$ to $\langle \mathsf{id}_{S_P} * P \rangle = \langle P \rangle$. If $Q \in P \setminus L$, then there exists a path $\alpha \in \mathsf{Path}(\mathsf{MN}(L))^{\langle PQ \rangle}_{\langle P \rangle}$, and $PQ \in L$ entails that $\langle PQ \rangle \in \top_{\mathsf{MN}(L)}$. □

Lemma 5.17. Subsidiary cells of $\mathsf{MN}(L)$ are not accessible. If $P \setminus L = \emptyset$, then the cell $\langle P \rangle$ is not coaccessible.

Proof:
If $\alpha \in \mathsf{Path}(\mathsf{MN}(L))^{w_U}_{\perp}$, then it contains a step β from a regular cell to a subsidiary cell (since all start cells are regular). Yet β can be neither an upstep (since lower faces of subsidiary cells are subsidiary) nor a downstep (since upper faces of regular cells are regular). This contradiction proves the first claim.

To prove the second part we use a similar argument. If $P \setminus L = \emptyset$, then a path $\alpha \in \mathsf{Path}(\mathsf{MN}(L))^{\top}_{\langle P \rangle}$ contains only regular cells (as shown above). Given that $R \setminus L \neq \emptyset$ for all $\langle R \rangle \in \top_{\mathsf{MN}(L)}$, α must contain a step β from $\langle Q \rangle$ to $\langle R \rangle$ such that $Q \setminus L = \emptyset$ and $R \setminus L \neq \emptyset$. If β is a downstep, *i.e.*, $\beta = (\langle Q \rangle \searrow_A \langle Q * U\!\downarrow\!_A \rangle)$, and $N \in R \setminus L = (Q * U\!\downarrow\!_A) \setminus L$, then $U\!\downarrow\!_A * N \in Q \setminus L \neq \emptyset$: a contradiction. If $\beta = (\langle R - A \rangle \nearrow^A \langle R \rangle)$ is an upstep and $N \in R \setminus L$, then, by Lemma 5.10,

$$(R - A) * {}_A\!\uparrow\!U * N \sqsubseteq R * N \in L,$$

implying that $Q \setminus L = (R - A) \setminus L \neq \emptyset$ by Lemma 5.1: another contradiction. □

Lemmas 5.16 and 5.17 together immediately imply the following.

Proposition 5.18. $\mathsf{ess}(\mathsf{MN}(L)) = \{\langle P \rangle \mid P \setminus L \neq \emptyset\}$.

5.4. $\mathsf{MN}(L)$ recognises L

We are finally ready to show that $\mathsf{Lang}(\mathsf{MN}(L)) = L$. One inclusion follows directly from Lemma 5.15:

Lemma 5.19. $L \subseteq \mathsf{Lang}(\mathsf{MN}(L))$.

Proof:
For every $P \in \mathsf{iiPoms}$ there exists a path $\alpha \in \mathsf{Path}(\mathsf{MN}(L))^{\langle P\rangle}_{\langle \mathsf{id}_{S_P}\rangle}$ such that $\mathsf{ev}(\alpha) = P$. If $P \in L$, then $\varepsilon \in P \setminus L$, *i.e.*, $\langle P\rangle$ is an accept cell. Thus α is accepting and $P = \mathsf{ev}(\alpha) \in \mathsf{Lang}(\mathsf{MN}(L))$. □

The converse inclusion requires more work. For a regular cell $\langle P\rangle$ of $\mathsf{MN}(L)$ denote $\langle P\rangle \setminus L = P \setminus L$ (this obviously does not depend on the choice of P).

Lemma 5.20. If $S \in \square$ and $\alpha \in \mathsf{Path}(\mathsf{MN}(L))_{\langle \mathsf{id}_S\rangle}$, then $\mathsf{tgt}(\alpha) \setminus L \subseteq \mathsf{ev}(\alpha) \setminus L$.

Proof:
By Lemma 5.17, all cells appearing along α are regular. We proceed by induction on the length of α. For $\alpha = (\langle \mathsf{id}_S\rangle)$ the claim is obvious. If α is non-trivial, we have two cases.

- $\alpha = \beta * (\delta^0_A(\langle P\rangle) \nearrow^A \langle P\rangle)$, where $\langle P\rangle \in \mathsf{MN}(L)[U]$ and $A \subseteq \mathsf{rfin}(P) \subseteq U \cong T_P$. By the induction hypothesis,
$$(P - A) \setminus L = \delta^0_A(\langle P\rangle) \setminus L = \mathsf{tgt}(\beta) \setminus L \subseteq \mathsf{ev}(\beta) \setminus L.$$
For $Q \in \mathsf{iiPoms}$ we have
$$\begin{aligned} Q \in P \setminus L \iff PQ \in L &\implies (P - A) * {}_A\!\uparrow U * Q \in L && \text{(Lemma 5.10)}\\ &\iff {}_A\!\uparrow U * Q \in (P - A) \setminus L \\ &\implies {}_A\!\uparrow U * Q \in \mathsf{ev}(\beta) \setminus L && \text{(induction hypothesis)}\\ &\iff \mathsf{ev}(\beta) * {}_A\!\uparrow U * Q \in L \\ &\iff \mathsf{ev}(\alpha) * Q \in L \iff Q \in \mathsf{ev}(\alpha) \setminus L. \end{aligned}$$
Thus, $\langle P\rangle \setminus L = P \setminus L \subseteq \mathsf{ev}(\alpha) \setminus L$.
- $\alpha = \beta * (\langle P\rangle \searrow_B \delta^1_B(\langle P\rangle))$, where $\langle P\rangle \in \mathsf{MN}(L)[U]$ and $B \subseteq U \cong T_P$. By inductive assumption, $P \setminus L = \mathsf{tgt}(\beta) \setminus L \subseteq \mathsf{ev}(\beta) \setminus L$. Thus,
$$\mathsf{tgt}(\alpha) \setminus L = \delta^1_B(\langle P\rangle) \setminus L = \langle P * U{\downarrow_B}\rangle \setminus L \subseteq (\mathsf{ev}(\beta) * U{\downarrow_B}) \setminus L = \mathsf{ev}(\alpha) \setminus L.$$

The inclusion above follows from Lemma 5.9. □

Proposition 5.21. $\mathsf{Lang}(\mathsf{MN}(L)) = L$.

Proof:
The inclusion $L \subseteq \mathsf{Lang}(\mathsf{MN}(L))$ is shown in Lemma 5.19. For the converse, let $S \in \square$ and $\alpha \in \mathsf{Path}(\mathsf{MN}(L))_{\langle \mathsf{id}_S\rangle}$, then Lemma 5.20 implies
$$\mathsf{tgt}(\alpha) \in \top_{\mathsf{MN}(L)} \iff \varepsilon \in \mathsf{tgt}(\alpha) \setminus L \implies \varepsilon \in \mathsf{ev}(\alpha) \setminus L \iff \mathsf{ev}(\alpha) \in L,$$
that is, if α is accepting, then $\mathsf{ev}(\alpha) \in L$. □

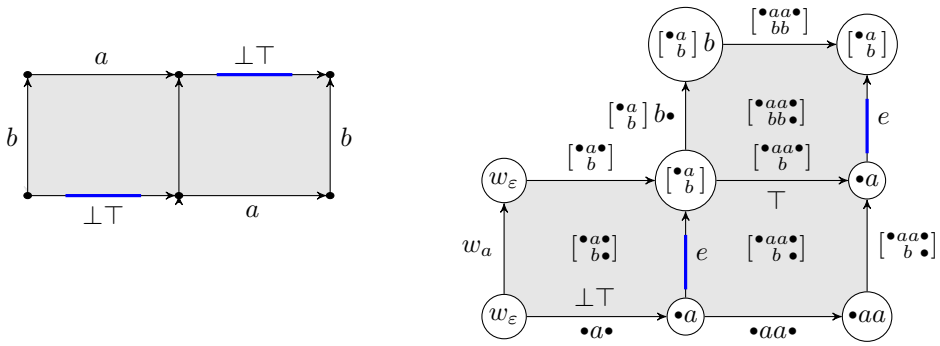


Figure 7. Two HDAs recognising the language of Example 5.23. On the left side, start/accept edges are identified; on the right, e-labelled edges are identified.

5.5. Finiteness of $\mathsf{MN}(L)$

The HDA $\mathsf{MN}(L)$ is not finite, since it contains infinitely many subsidiary cells w_U. Below we show that its essential part $\mathsf{MN}(L)^{\mathsf{ess}}$ is finite if L has finitely many prefix quotients.

Lemma 5.22. If $\mathsf{suff}(L)$ is finite, then $\mathsf{ess}(\mathsf{MN}(L))$ is finite.

Proof:
For $\langle P\rangle, \langle Q\rangle \in \mathsf{ess}(L)$, we have $\langle P\rangle = \langle Q\rangle \iff f(\langle P\rangle) = f(\langle Q\rangle)$, where

$$f(\langle P\rangle) = (P \setminus L, \mathsf{fin}(P), ((P - A) \setminus L)_{A \subseteq \mathsf{rfin}(P)}).$$

We will show that f takes only finitely many values on $\mathsf{ess}(L)$. Indeed, $P \setminus L$ belongs to the finite set $\mathsf{suff}(L)$. Further, all ipomsets in $P \setminus L$ have source interfaces equal to T_P. Since $P \setminus L$ is non-empty, $\mathsf{fin}(P)$ is a starter with T_P as underlying conclist. Yet, there are only finitely many starters on any conclist. The last coordinate also may take only finitely many values, since $\mathsf{rfin}(P)$ is finite and $(P - A) \setminus L \in \mathsf{suff}(L)$. □

Proof of Theorem 5.2, (b) $\Longrightarrow$ (a)**:**
By Lemma 5.22 and Lemma 4.8, $\mathsf{MN}(L)^{\mathsf{ess}}$ is a finite HDA. With Proposition 5.21, $\mathsf{Lang}(\mathsf{MN}(L)^{\mathsf{ess}}) = \mathsf{Lang}(\mathsf{MN}(L)) = L$. □

Example 5.23. We finish this section with another example, which shows some subtleties related to higher-dimensional loops. Let L be the language of the HDA shown to the left of Figure 7 (a looping version of the HDA of Figure 5), then

$$L = \{\bullet a\bullet\} \cup \{[{}^{\bullet aa\bullet}_{b}]^n \mid n \geq 1\}{\downarrow}.$$

Our construction yields $\mathsf{MN}(L)^{\mathsf{ess}}$ as shown on the right of the figure. Here, $e = \langle [{}^{\bullet a}_{b\bullet}]\rangle$, and the two e-labelled edges and their corresponding faces are identified. These identifications follow from the fact that $[{}^{\bullet aa}_{bb\bullet}] \approx_L [{}^{\bullet a}_{b\bullet}]$, $[{}^{\bullet aa}_{bb}] \approx_L [{}^{\bullet a}_{b}]$, and $[{}^{\bullet aa}_{b}] \approx_L \bullet a$. Note that $[{}^{\bullet a\bullet}_{b\bullet}]$ and $[{}^{\bullet aa\bullet}_{bb\bullet}]$ are not strongly equivalent, since they have different signatures: $[{}^{\bullet a\bullet}_{b\bullet}]$ and $[{}^{a\bullet}_{b\bullet}]$, respectively.

6. Determinism

We now make precise our notion of determinism and show that not all HDAs may be determinised. Recall that we do not assume finiteness.

Definition 6.1. An HDA X is *deterministic* if

1. for every $U \in \square$ there is at most one initial cell in $X[U]$, and
2. for all $V \in \square$, $A \subseteq V$ and any essential cell $x \in X[V - A]$ there exists at most one essential cell $y \in X[V]$ such that $x = \delta^0_A(y)$.

That is, in any essential cell x in a deterministic HDA X and for any set A of events, there is at most one way to start A in x and remain in the essential part of X (recall that termination of events is always deterministic). We allow multiple initial cells because ipomsets in $\mathsf{Lang}(X)$ may have different source interfaces; for each source interface in $\mathsf{Lang}(X)$, there can be at most one matching start cell in X. Note that we must restrict our definition to essential cells as inessential cells may not always be removed (in contrast to the case of standard automata).

A language is deterministic if it is recognised by a deterministic HDA. We develop a language-internal criterion for being deterministic.

Definition 6.2. A language L is *swap-invariant* if it holds for all $P, Q, P', Q' \in \mathsf{iiPoms}$ that $PP' \in L$, $QQ' \in L$ and $P \sqsubseteq Q$ imply $QP' \in L$.

That is, if the P prefix of $PP' \in L$ is subsumed by Q (which is, thus, "more concurrent" than P), and if Q itself may be extended to an ipomset in L, then P may be swapped for Q in the ipomset PP' to yield $QP' \in L$.

Lemma 6.3. L is swap-invariant if and only if $P \sqsubseteq Q$ implies $P \setminus L = Q \setminus L$ for all $P, Q \in \mathsf{iiPoms}$, unless $Q \setminus L = \emptyset$.

Proof:
Assume that L is swap-invariant and let $P \sqsubseteq Q$. The inclusion $Q \setminus L \subseteq P \setminus L$ follows from Lemma 5.1, and

$$R \in Q \setminus L,\ R' \in P \setminus L \iff QR, PR' \in L \implies QR' \in L \iff R' \in Q \setminus L$$

implies that $P \setminus L \subseteq Q \setminus L$. The calculation

$$PP', QQ' \in L,\ P \sqsubseteq Q \iff P' \in P \setminus L,\ Q' \in Q \setminus L,\ P \sqsubseteq Q \implies P' \in Q \setminus L \iff QP' \in L$$

shows the converse. □

Our main goal is to show the following criterion, which will be implied by Propositions 6.10 and 6.12 below.

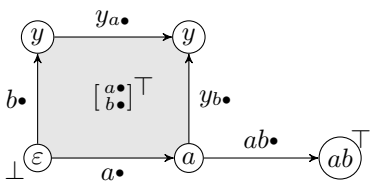
Theorem 6.4. A language L is deterministic if and only if it is swap-invariant.

Example 6.5. The regular language $L = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}], ab, ba, abc\}$ from Example 5.5 is not swap-invariant: using Lemma 6.3, $ab\bullet \sqsubseteq [\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}]$, but $\{ab\bullet\} \setminus L = \{\bullet b, \bullet bc\} \neq \{\bullet b\} = \{[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}]\} \setminus L$. Hence L is not deterministic.

The next examples explain why we need to restrict to essential cells in the definition of deterministic HDAs.

Example 6.6. The HDA in Example 5.8 is deterministic. There are two different a-labelled edges starting at w_ε (w_a and $\langle[\begin{smallmatrix} \bullet a \\ \bullet a\bullet \end{smallmatrix}]\rangle$), yet it does not disturb determinism since w_ε is not accessible.

Example 6.7. Let $L = \{ab, [\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}]\}$. Then $\mathsf{MN}(L)^{\mathsf{ess}}$ is as follows:



It is deterministic: there are two b-labelled edges leaving a, namely $y_{b\bullet}$ and $ab\bullet$, but only the latter is coaccessible.

The next lemma shows that up to path equivalence, paths on deterministic HDAs are determined by their labels. That is, up to path equivalence, deterministic HDAs are *unambiguous*. Note that [1] shows that non-deterministic HDAs may exhibit unbounded ambiguity.

Lemma 6.8. Let X be a deterministic HDA and $\alpha, \beta \in \mathsf{Path}(X)_\bot$ with $\mathsf{tgt}(\alpha), \mathsf{tgt}(\beta) \in \mathsf{ess}(X)$. If $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$, then $\alpha \simeq \beta$.

Proof:
We can assume that $\alpha = \alpha_1 * \cdots * \alpha_n$ and $\beta = \beta_1 * \cdots * \beta_m$ are sparse; note that all of these cells are essential. We show that $\alpha_k = \beta_k$ for all k which implies the claim.

Denote $P = \mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$, then

$$P = \mathsf{ev}(\alpha_1) * \cdots * \mathsf{ev}(\alpha_n)$$

is a sparse step decomposition of P. Similarly, $P = \mathsf{ev}(\beta_1) * \cdots * \mathsf{ev}(\beta_m)$ is a sparse step decomposition. Yet sparse step decompositions are unique by Proposition 3.5; hence, $m = n$ and $\mathsf{ev}(\alpha_k) = \mathsf{ev}(\beta_k)$ for every k.

We show by induction that $\alpha_k = \beta_k$. First, $\mathsf{ev}(\alpha_0) = \mathsf{ev}(\beta_0)$ implies $\alpha_0 = \beta_0$ by determinism. Now assume that $\alpha_{k-1} = \beta_{k-1}$. Let $x = \mathsf{src}(\alpha_k) = \mathsf{tgt}(\alpha_{k-1}) = \mathsf{tgt}(\beta_{k-1}) = \mathsf{src}(\beta_k)$. If $P_k = \mathsf{ev}(\alpha_k) = \mathsf{ev}(\beta_k)$ is a terminator $U{\downarrow_B}$, then $\alpha_k = \delta^1_B(x) = \beta_k$. If P_k is a starter ${}_A{\uparrow}U$, then there are $y, z \in X$ such that $\delta^0_A(y) = \delta^0_A(z) = x$. As y and z are essential and X is deterministic, this implies $y = z$ and $\alpha_k = \beta_k$. $\square$

Lemma 6.9. Let α and β be essential paths on a deterministic HDA X. Assume that $\mathsf{src}(\alpha) = \mathsf{src}(\beta)$ and $\mathsf{ev}(\alpha) \sqsubseteq \mathsf{ev}(\beta)$. Then $\mathsf{tgt}(\alpha) = \mathsf{tgt}(\beta)$.

Proof:
By Lemma 4.12, there exists a path $\gamma \in \mathsf{Path}(X)^{\mathsf{tgt}(\beta)}_{\mathsf{src}(\beta)}$ such that $\mathsf{ev}(\gamma) = \mathsf{ev}(\alpha)$. Lemma 6.8 implies that $\gamma \simeq \alpha$ and then $\mathsf{tgt}(\alpha) = \mathsf{tgt}(\gamma) = \mathsf{tgt}(\beta)$. □

Proposition 6.10. If L is deterministic, then L is swap-invariant.

Proof:
Let X be a deterministic HDA that recognises L and fix ipomsets $P \sqsubseteq Q$. From Lemma 5.1 follows that $Q \setminus L \subseteq P \setminus L$. It remains to prove that if $Q \setminus L \neq \emptyset$, then $P \setminus L \subseteq Q \setminus L$. Denote $U \cong S_P \cong S_Q$.

Let $R \in Q \setminus L$ and let $\omega \in \mathsf{Path}(X)^{\top}_{\langle \mathsf{id}_U \rangle}$ be an accepting path that recognises QR. By Lemma 4.13, there exists a path $\beta \in \mathsf{Path}(X)_{\langle \mathsf{id}_U \rangle}$ such that $\mathsf{ev}(\beta) = Q$.

Now assume that $R' \in P \setminus L$, and let $\omega' \in \mathsf{Path}(X)^{\top}_{\langle \mathsf{id}_U \rangle}$ be a path such that $\mathsf{ev}(\omega') = PR'$. By Lemma 4.13, there exist paths $\alpha \in \mathsf{Path}(X)_{\langle \mathsf{id}_U \rangle}$ and $\gamma \in \mathsf{Path}(X)^{\mathsf{tgt}(\omega')}$ such that $\mathsf{tgt}(\alpha) = \mathsf{src}(\gamma)$, $\mathsf{ev}(\alpha) = P$ and $\mathsf{ev}(\gamma) = R'$. From Lemma 6.9 and $P \sqsubseteq Q$ follows that $\mathsf{tgt}(\alpha) = \mathsf{tgt}(\beta)$. Thus, β and γ may be concatenated to an accepting path $\beta * \gamma$. By $\mathsf{ev}(\beta * \gamma) = QR'$ we have $QR' \in L$, *i.e.*, $R' \in Q \setminus L$. □

Lemma 6.11. If $\langle P \rangle \in \mathsf{ess}(\mathsf{MN}(L))$ and $A \subseteq \mathsf{rfin}(P)$, then $\langle P - A \rangle \in \mathsf{ess}(\mathsf{MN}(L))$.

Proof:
By Lemma 5.16, $\langle P - A \rangle$ is accessible. By assumption, $\langle P \rangle$ is coaccessible and $(\langle P - A \rangle \nearrow^{A} \langle P \rangle)$ is a path, so $\langle P - A \rangle$ is also coaccessible. □

Proposition 6.12. If L is swap-invariant, then $\mathsf{MN}(L)$ and $\mathsf{MN}(L)^{\mathsf{ess}}$ are deterministic.

Proof:
Since $\mathsf{MN}(L)^{\mathsf{ess}}$ is a sub-HDA of $\mathsf{MN}(L)$, it suffices to prove that $\mathsf{MN}(L)$ is deterministic. $\mathsf{MN}(L)$ contains only one start cell $\langle \mathsf{id}_U \rangle$ for every $U \in \square$.

Fix $U \in \square$, $P, Q \in \mathsf{iiPoms}_U$ and $A \subseteq U$. Assume that $\delta^0_A(\langle P \rangle) = \delta^0_A(\langle Q \rangle)$, *i.e.*, $\langle P - A \rangle = \langle Q - A \rangle$, and $\langle P \rangle, \langle Q \rangle, \langle P - A \rangle \in \mathsf{ess}(\mathsf{MN}(L))$. We will prove that $\langle P \rangle = \langle Q \rangle$, or equivalently, $P \approx_L Q$.

We have $\mathsf{fin}(P - A) = \mathsf{fin}(Q - A) =: {}_S\!\uparrow\!(U - A)$. First, notice that A, regarded as a subset of P (or Q), contains no start events: else, we would have $\delta^0_A(\langle P \rangle) = w_{U-A}$ (or $\delta^0_A(\langle Q \rangle) = w_{U-A}$). As a consequence, $\mathsf{fin}(P) = \mathsf{fin}(Q) = {}_S\!\uparrow\!U$.

For every $B \subseteq \mathsf{rfin}(P) = \mathsf{rfin}(Q)$ we have

$$\begin{aligned} P - A \approx_L Q - A &\implies (P - (A \cup B)) \setminus L = (Q - (A \cup B)) \setminus L \\ &\implies ((P - (A \cup B)) * {}_{(A-B)}\!\uparrow\!U) \setminus L = ((Q - (A \cup B)) * {}_{(A-B)}\!\uparrow\!U) \setminus L. \end{aligned}$$

The first implication follows from the definition, and the second from Lemma 5.9. From Lemma 5.10 follows that

$$(P - (A \cup B)) *_{(A-B)} \uparrow U \sqsubseteq P - B, \quad (Q - (A \cup B)) *_{(A-B)} \uparrow U \sqsubseteq Q - B.$$

Thus, by swap-invariance we have $(P - B) \setminus L = (Q - B) \setminus L$; note that Lemma 6.11 guarantees that neither of these languages is empty. □

7. Higher-Dimensional Automata with Interfaces

Higher-dimensional automata with interfaces (iHDAs) were introduced in [9] as a tool that allowed to prove a Kleene theorem for HDAs. Both HDAs and iHDAs recognise the same class of languages, yet, compared to iHDAs, HDAs have a flaw: they enforce introducing non-essential cells that serve solely as faces of other cells. We will show below that essential parts of iHDAs are again iHDAs, a fact which allows us to give a Myhill-Nerode construction using iHDAs which proceeds along different lines and, we believe, is more simple and principled.

We will also provide a notion of deterministic iHDAs which, again, is simpler in that it does not have to restrict to essential cells, and show that the notions of deterministic languages of HDAs and iHDAs agree.

7.1. Iprecubical sets and iHDAs

The main difference between HDAs and iHDAs is that events in iHDAs may be marked as *source events* or *target events*. Accepting runs may never terminate target events and, similarly, source events must have been present from the very beginning of an accepting run.

A *concurrency list with interfaces* (*iconclist*) $(U, \dashrightarrow, S, T, \lambda)$ is a conclist $(U, \dashrightarrow, \lambda)$ together with subsets $S, T \subseteq U$. Equivalently, iconclists are iposets with empty precedence relation; conclists are iconclists with empty interfaces. We write ${}_SU_T$ for an iconclist as above.

Let $\mathrm{I}\square$ denote the set of iconclists. An *iprecubical set* consists of a set of cells X together with a mapping $\mathsf{iev} : X \to \mathrm{I}\square$. For an iconclist ${}_SU_T$ we write $X[{}_SU_T] = \{x \in X \mid \mathsf{iev}(x) = {}_SU_T\}$. Face maps in iprecubical sets cannot unstart events in source interfaces neither terminate events in target interfaces. That is, for every iconclist ${}_SU_T$ and subsets $A, B \subseteq U$ such that $A \cap S = B \cap T = \emptyset$ there are *face maps*

$$\delta^0_A : X[{}_SU_T] \to X[{}_S(U - A)_{(T-A)}], \qquad \delta^1_B : X[{}_SU_T] \to X[{}_{(S-B)}(U - B)_T].$$

Further, for $A, B \subseteq U$ with $A \cap B = \emptyset$ and $\nu, \mu \in \{0, 1\}$, $\delta^\nu_A \delta^\mu_B = \delta^\mu_B \delta^\nu_A$ whenever these are defined.

A *higher-dimensional automaton with interfaces* (*iHDA*) is an iprecubical set X together with subsets $\bot_X, \top_X \subseteq X$ of *start* and *accept* cells such that for all $x \in \bot_X$ with $\mathsf{iev}(x) = {}_SU_T$, $S = U$ and for all $x \in \top_X$ with $\mathsf{iev}(x) = {}_SU_T$, $T = U$. That is, events in start cells are source events and cannot be unstarted, and events in accept cells are target events and cannot be terminated.

Remark 7.1. Every precubical set X may be regarded as an iprecubical set X' such that $X'[{}_\emptyset U_\emptyset] = X[U]$ and $X'[{}_SU_T] = \emptyset$ whenever $S \neq \emptyset$ or $T \neq \emptyset$. If X is an HDA and all its start and accept cells

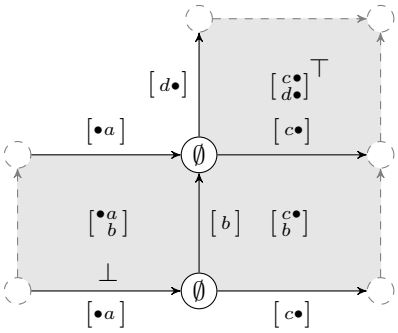


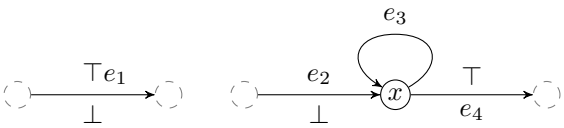
Figure 8. An example of an iHDA. Cells are marked with their event iconclists.

are vertices (elements of $X[\emptyset]$), then X' may be regarded as an iHDA as well. This fails in presence of higher-dimensional start or accept cells due to the condition on event iconclists of such cells.

Example 7.2. Let X be the iHDA defined by $X = \{x, e_1, e_2, e_3, e_4\}$, $\mathsf{ev}(x) = \emptyset$,

$$\mathsf{ev}(e_1) = {}_a a_a, \quad \mathsf{ev}(e_2) = {}_a a_\emptyset, \quad \mathsf{ev}(e_3) = {}_\emptyset a_\emptyset, \quad \mathsf{ev}(e_4) = {}_\emptyset a_a,$$

$\delta^1_a(e_2) = \delta^0_a(e_3) = \delta^1_a(e_3) = \delta^0_a(e_4) = x$, and $\bot_X = \{e_1, e_2\}$, $\top_X = \{e_1, e_4\}$. Note that e_1 has neither an upper nor a lower face since its only event a is in both interfaces. For the opposite reason, the edge e_3 can be neither start nor accept cell.



Example 7.3. Figure 8 shows an example of a two-dimensional iHDA. The initial cell has event iconclist ${}_a a_\emptyset$ and hence no lower face. This lack of lower face propagates to the left two-dimensional cell, with event iconclist $\left[\begin{smallmatrix} \bullet a \\ b \end{smallmatrix}\right]$. Hence iHDAs are *partial* HDAs in the sense of [7, 12], but the notion of partiality is more restricted here, given that it is on the level of events.

7.2. Paths and languages

Paths on iHDAs are defined as for HDAs. Namely, a path is a sequence $\alpha = (x_0, \varphi_1, x_1, \dots, x_n)$ such that each $(x_{i-1}, \varphi_i, x_i)$ is either

- an *upstep* $(\delta^0_A(x_i), \nearrow^A, x_i)$ for $x_i \in X[{}_S U_T]$, $A \subseteq U - S$, or
- a *downstep* $(x_{i-1}, \searrow_B, \delta^1_B(x_{i-1}))$ for $x_{i-1} \in X[{}_S U_T]$, $B \subseteq U - T$.

A path α is accepting if $\mathsf{src}(\alpha) = x_0$ is a start cell and $\mathsf{tgt}(\alpha) = x_n$ is an accept cell.

For a cell $x \in X$ of an iHDA X we denote by $\mathsf{ev}(x)$ the underlying conclist of $\mathsf{iev}(x)$; note that

$$\mathsf{iev}(x) = (S_{\mathsf{iev}(x)}, \mathsf{ev}(x), T_{\mathsf{iev}(x)}).$$

The *event ipomset* of a path α is defined inductively as before: $\mathsf{ev}((x)) = \mathsf{id}_{\mathsf{ev}(x)}$, $\mathsf{ev}(y \nearrow^A x) = {}_A\!\uparrow\mathsf{ev}(x)$, $\mathsf{ev}(x \searrow_B y) = \mathsf{ev}(x)\!\downarrow_B$, and $\mathsf{ev}(\alpha * \beta) = \mathsf{ev}(\alpha) * \mathsf{ev}(\beta)$. The *language* of an iHDA X is

$$\mathsf{Lang}(X) = \{\mathsf{ev}(\alpha) \mid \alpha \text{ accepting path in } X\}.$$

Example 7.4. The language of the iHDA from Example 7.2 is $\{\bullet a \bullet\} \cup \{\bullet a a^n a \bullet \mid n \geq 0\}$. The language of the iHDA from Example 7.3 is

$$\left\{ \left[\begin{array}{ccc} \bullet a & \longrightarrow & c \bullet \\ & \searrow & \\ b & \longrightarrow & d \bullet \end{array} \right] \right\} \downarrow.$$

Because of the requirement that events in start cells may not be unstarted and those in accept cells may not be terminated, an event in an iHDA carries information whether it will be eventually terminated, and whether it has been present from the beginning. This is expressed by the following lemma which shows that iconclists of cells may be recovered from ipomsets of accepting paths:

Lemma 7.5. Let X be an iHDA, α be a path in X and $P = \mathsf{ev}(\alpha)$.

1. If $\mathsf{src}(\alpha) \in \bot_X$, then $\mathsf{iev}(\mathsf{tgt}(\alpha)) \cong (S_P \cap T_P, T_P, Z)$ for some $Z \subseteq T_P$.
2. If $\mathsf{tgt}(\alpha) \in \top_X$, then $\mathsf{iev}(\mathsf{src}(\alpha)) \cong (Y, S_P, S_P \cap T_P)$ for some $Y \subseteq S_P$.
3. If α is accepting, then $\mathsf{iev}(\mathsf{src}(\alpha) \cong (S_P, S_P, S_P \cap T_P)$ and $\mathsf{iev}(\mathsf{tgt}(\alpha) \cong (S_P \cap T_P, T_P, T_P)$.

Proof:
It is sufficient to prove 1., 2. is then obtained by reversal and 3. follows from 1. and 2. Induction with respect to the length of α. If $\alpha = (x)$, then $P = \mathsf{id}_U = (U, U, U)$ and $\mathsf{iev}(x) = (U, U, Z)$.

If $\alpha = \beta * (\delta^0_A(x) \nearrow^A x)$, $\mathsf{iev}(x) = (S, U, T)$, and $A \subseteq U - S$, then $\mathsf{ev}(\beta) = P - A$ and

$$\mathsf{iev}(\delta^0_A(x)) = (S, U - A, T - A) = (S_{P-A} \cap T_{P-A}, T_{P-A}, Z) = (S_P \cap T_P, T_P - A, Z)$$

by the inductive hypothesis for β. Thus, $(S, U, T) = (S_P \cap T_P, T_P, Z)$.

Finally, let $\alpha = \beta * (y \searrow_B \delta^1_B(y))$, $\mathsf{iev}(y) = (S, U, T)$, and $B \subseteq U - T$. Denote $\mathsf{ev}(\beta) = Q$, then we have $P = Q * T_Q\!\downarrow_B$, $S_P = S_Q$ and $T_P = T_Q - B$. Therefore,

$$\begin{aligned} \mathsf{iev}(\mathsf{tgt}(\alpha)) &= (S - B, U - B, T) = (S, U, T) - B = \mathsf{iev}(\mathsf{tgt}(\beta)) - B \overset{\text{ind.}}{=} \\ &(S_Q \cap T_Q, T_Q, Z) - B = (S_Q \cap (T_Q - B), T_Q - B, Z - B) = (S_P \cap T_P, T_P, Z - B). \end{aligned}$$

The proof is complete. □

7.3. HDAs vs. iHDAs

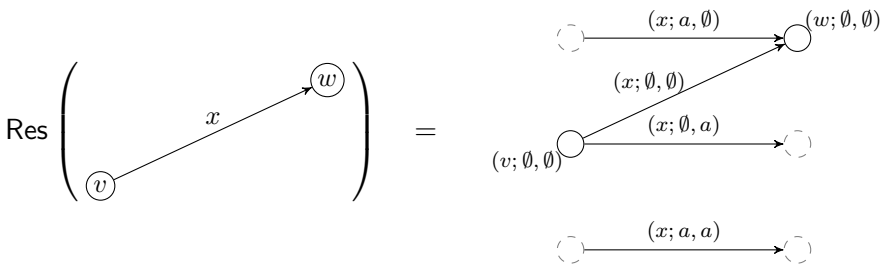
HDAs and iHDAs are related via a pair of adjoint functors: resolution which maps an HDA X to an iHDA $\mathsf{Res}(X)$ by adjoining all possible assignments of interfaces, and its left adjoint closure, which maps an iHDA X to an HDA $\mathsf{Cl}(X)$ by filling in missing faces. These are introduced in [10] and have the important property that they preserve languages. We define them below and develop some lemmas.

The *resolution* of an HDA X is the iHDA $\mathsf{Res}(X)$ defined as follows. For ${}_SU_T \in \mathbb{I}\square$, $A \subseteq U - S$ and $B \subseteq U - T$ we put

$$\mathsf{Res}(X)[{}_SU_T] = \{(x; S, T) \mid x \in X[U]\},$$
$$\delta^0_A((x; S, T)) = (\delta^0_A(x); S, T - A), \qquad \delta^1_B((x; S, T)) = (\delta^1_B(x); S - B, T).$$

A cell $(x; S, T) \in \mathsf{Res}(X)[{}_SU_T]$ is a start cell if $x \in X_\perp$ and $S = U$, and an accept cell if $x \in X^\top$ and $T = U$. Every cell $x \in X[U]$ thus produces $4^{|U|}$ cells in $\mathsf{Res}(X)$, hence if X is finite, then so is $\mathsf{Res}(X)$.

Example 7.6. For the precubical set X with $X[a] = \{x\}$ and $X[\emptyset] = \{v, w\}$ we have



If $((x_0; S_0, T_0), \varphi_1, (x_1; S_1, T_1), \varphi_2, \dots, \varphi_n, (x_n; S_n, T_n))$ is an accepting path in $\mathsf{Res}(X)$, then $(x_0, \varphi_1, x_1, \varphi_2 \dots, \varphi_n, x_n)$ is an accepting path in X with the same event ipomset. Conversely, for every accepting path $\alpha = (x_0, \varphi_1, \dots, x_n)$ in X there exists unique subsets $S_k, T_k \subseteq \mathsf{ev}(x_k)$ such that $((x_0; S_0, T_0), \varphi_1, \dots, (x_n; S_n, T_n))$ is an accepting path in $\mathsf{Res}(X)$. (Indeed, $S_0 = \mathsf{ev}(x_0)$, $T_n = \mathsf{ev}(x_n)$, S_k and φ_k determine S_{k+1}, T_{k+1} and φ_k determine T_k). As a consequence we obtain:

Lemma 7.7. Let X be an HDA. If $(x; S, T) \in \mathsf{Res}(X)$ is essential, then $x \in X$ is essential.

Lemma 7.8. ([10, Prop. 11.2])
For any HDA X, $\mathsf{Lang}(\mathsf{Res}(X)) = \mathsf{Lang}(X)$.

The *closure* of an iHDA X is the HDA $\mathsf{Cl}(X)$ defined, for all $U \in \square$, by

- $\mathsf{Cl}(X) = \{[x; A, B] \mid x \in X,\ A \subseteq S_{\mathsf{iev}(x)},\ B \subseteq T_{\mathsf{iev}(x)},\ A \cap B = \emptyset\}$;
- $\mathsf{ev}([x; A, B]) = \mathsf{ev}(x) - (A \cup B)$ for $[x; A, B] \in \mathsf{Cl}(X)$;
- $\delta^0_C([x; A, B]) = [\delta^0_{C - S_{\mathsf{iev}(x)}}(x); A \cup (C \cap S_{\mathsf{iev}(x)}), B]$ for $C \subseteq \mathsf{ev}([x; A, B])$;
- $\delta^1_D([x; A, B]) = [\delta^1_{D - T_{\mathsf{iev}(x)}}(x); A, B \cup (D \cap T_{\mathsf{iev}(x)})]$ for $D \subseteq \mathsf{ev}([x; A, B])$;

- $\top_{\mathsf{Cl}(X)} = \{[x;\emptyset,\emptyset] \mid x \in \bot_X\}$, $\top_{\mathsf{Cl}(X)} = \{[x;\emptyset,\emptyset] \mid x \in \top_X\}$.

Intuitively, closure fills in the missing cells of the iHDA X. Lower face maps δ^0_C of $\mathsf{Cl}(X)$ take as much of the face map of X as possible, while the remaining events are added to the set A; similarly for upper faces.

Lemma 7.9. Let X be an iHDA and $x, y \in X$. The function $\Phi : \mathsf{Path}(X)^y_x \to \mathsf{Path}(\mathsf{Cl}(X))^{[y;\emptyset,\emptyset]}_{[x;\emptyset,\emptyset]}$,

$$\Phi(x_0, \varphi_1, x_1, \dots, x_n) = ([x_0;\emptyset,\emptyset], \varphi_1, [x_1;\emptyset,\emptyset], \dots, [x_n;\emptyset,\emptyset])$$

is a bijection. Moreover, $\mathsf{ev}(\alpha) = \mathsf{ev}(\Phi(\alpha))$ for all α.

Proof:
Injectivity of Φ is clear. Let $\alpha = ([x_0, A_0, B_0], \varphi_1, \dots, \varphi_n, [x_n, A_n, B_n]) \in \mathsf{Path}(\mathsf{Cl}(X))^{[y;\emptyset,\emptyset]}_{[x;\emptyset,\emptyset]}$. For every step $([x_k; A_k, B_k], \varphi_k, [x_{k+1}; A_{k+1}, B_{k+1}])$ it follows from the definition of face maps that $|A_k| \geq |A_{k+1}|$ and $|B_k| \leq |B_{k+1}|$. Thus, $A_k \subseteq A_0 = \emptyset$ and $B_k \subseteq B_n = \emptyset$ for all k and then $\alpha = \Phi(x_0, \varphi_1, x_1, \dots, x_n)$. The second claim is obvious. □

Lemma 7.10. ([10, Prop. 11.4])
For any iHDA X, $\mathsf{Lang}(\mathsf{Cl}(X)) = \mathsf{Lang}(X)$.

The following two lemmas are analogues to Lemmas 4.12 and 4.13 for iHDAs.

Lemma 7.11. Let X be an iHDA, $x, y \in X$, $\alpha \in \mathsf{Path}(X)^y_x$ and $P \sqsubseteq Q = \mathsf{ev}(\alpha)$. Then there exists $\beta \in \mathsf{Path}(X)^y_x$ such that $\mathsf{ev}(\beta) = P$.

Proof:
This follows from Lemma 4.12 applied to $\mathsf{Cl}(X)$ and Lemma 7.9. □

Lemma 7.12. Let X be an iHDA, $x, y \in X$ and $\gamma \in \mathsf{Path}(X)^y_x$. Assume that $\mathsf{ev}(\gamma) = P * Q$ for ipomsets P and Q. Then there exist paths $\alpha \in \mathsf{Path}(X)_x$ and $\beta \in \mathsf{Path}(X)^y$ such that $\mathsf{ev}(\alpha) = P$, $\mathsf{ev}(\beta) = Q$ and $\mathsf{tgt}(\alpha) = \mathsf{src}(\beta)$.

Proof:
We apply Lemma 4.13 to the path $\Phi(\gamma)$ and obtain that there are paths α' and β' in $\mathsf{Cl}(X)$ such that $\mathsf{ev}(\alpha') = P$, $\mathsf{ev}(\beta') = Q$ and $\mathsf{tgt}(\alpha') = \mathsf{src}(\beta')$. By Lemma 7.9, $\alpha = \Phi^{-1}(\alpha')$ and $\beta = \Phi^{-1}(\beta')$ satisfy the required conditions. □

7.4. Essential iHDAs

As for HDAs, we say that a cell $x \in X$ of an iHDA X is *essential* if it accessible and coaccessible. Let $\mathsf{ess}(X) \subseteq X$ be the set of essential cells. We show below that, contrary to the situation for HDAs, $\mathsf{ess}(X)$ is itself an iHDA.

Let $\mathsf{dist}(x, y)$ be the minimal length of a path from x to y. A cell y is accessible if $\mathsf{dist}(x, y) < \infty$ for some $x \in \bot_X$ and is coaccessible if $\mathsf{dist}(y, z) < \infty$ for some $z \in \top_X$. The following follows directly.

Lemma 7.13. Let $y \in X[_S U_T]$, $A \subseteq U - S$ and $B \subseteq U - T$.

- For any $x \in X$, $\mathsf{dist}(x, y) \leq \mathsf{dist}(x, \delta^0_A(y)) + 1$ and $\mathsf{dist}(x, \delta^1_B(y)) \leq \mathsf{dist}(x, y) + 1$.
- For any $z \in X$, $\mathsf{dist}(y, z) \leq \mathsf{dist}(\delta^1_B(y), z) + 1$ and $\mathsf{dist}(\delta^0_A(y), z) \leq \mathsf{dist}(y, z) + 1$.

The next lemma only holds because of the special properties of start and accept cells in iHDAs.

Lemma 7.14. Let $y \in X[_S U_T]$, $A \subseteq U - S$ and $B \subseteq U - T$.

- For every $x \in \bot_X$, $\mathsf{dist}(x, \delta^0_A(y)) \leq \mathsf{dist}(x, y)$.
- For every $z \in \top_X$, $\mathsf{dist}(y, z) \geq \mathsf{dist}(\delta^1_B(y), z)$.

Proof:
We only show the first inequality; the second is symmetric. We fix x and proceed by induction on cells y with respect to $\mathsf{dist}(x, y)$. If $\mathsf{dist}(x, y) = 0$, then $y = x$ is a start cell. Thus, $S = U$, $A = \emptyset$ and $\delta^0_A(y) = y$.

Now let $n = \mathsf{dist}(x, y) > 0$. Without loss of generality we may assume that $A = \{a\}$. Let α be a path from x to y of length n, and let $\alpha = \beta * \gamma$ be a decomposition with γ having length 1. Clearly, $\mathsf{dist}(x, \mathsf{src}(\gamma)) = n - 1$. Consider three cases:

- $\gamma = (\delta^0_B(y) \nearrow^B y)$ and $a \in B$. Then $\beta * (\delta^0_B(y) \nearrow^{B-a} \delta^0_a(y))$ has length n.
- $\gamma = (\delta^0_B(y) \nearrow^B y)$ and $a \notin B$. Then $\mathsf{dist}(x, \delta^0_a(\delta^0_B(y))) = \mathsf{dist}(x, \delta^0_B(\delta^0_a(y))) \leq n - 1$ by induction, and then by Lemma 7.13, $\mathsf{dist}(x, \delta^0_a(y)) \leq \mathsf{dist}(x, \delta^0_B(\delta^0_a(y))) + 1 \leq n$.
- $\gamma = (z \searrow_B y)$. Then $y = \delta^1_B(z)$, and $\mathsf{dist}(x, \delta^0_a(z)) \leq \mathsf{dist}(x, z) = n - 1$ by induction. By Lemma 7.13, $\mathsf{dist}(x, \delta^0_a(y)) = \mathsf{dist}(x, \delta^1_B(\delta^0_a(z))) \leq \mathsf{dist}(x, \delta^0_a(z)) + 1 \leq n$.

□

Proposition 7.15. For every iHDA X, $\mathsf{ess}(X) \subseteq X$ is an iHDA.

Proof:
Let $y \in X[_S U_T]$ be essential. We show that all faces of y are also essential. There exist $x \in \bot_X$ and $z \in \top_X$ such that $\mathsf{dist}(x, y), \mathsf{dist}(y, z) < \infty$. By Lemmas 7.13 and 7.14, $\mathsf{dist}(x, \delta^0_A(y)), \mathsf{dist}(x, \delta^1_B(y)) < \infty$ and $\mathsf{dist}(\delta^0_A(y), z), \mathsf{dist}(\delta^1_B(y), z) < \infty$ for all $A \subseteq U - S$, $B \subseteq U - T$ as well. Thus, all faces of y are essential, which concludes the proof. □

8. Myhill-Nerode Construction for iHDAs

We now develop a Myhill-Nerode construction which for a given regular language L constructs an iHDA $\mathsf{iMN}(L)$. Our construction proceeds in several steps. First we construct a universal iHDA iFree which recognises all ipomsets, then we restrict iFree depending on the given language, and finally we quotient this $\mathsf{iFree}(L)$ by an equivalence relation which preserves its language and ensures that the quotient is finite if L is regular.

8.1. iFree

The universal iHDA iFree is defined as follows:

$$\begin{gathered}
\mathsf{iFree} = \{(P,Z) \mid P \in \mathsf{iiPoms},\ Z \subseteq T_P\}, \qquad \mathsf{iev}(P,Z) = (S_P \cap T_P, T_P, Z),\\
\delta^0_A(P,Z) = (P - A, Z - A) \text{ for } (P,Z) \in \mathsf{iFree}[{}_S U_T], A \subseteq U - S \cong T_P - S_P,\\
\delta^1_B(P,Z) = (P * T_P{\downarrow}_B, Z) \text{ for } (P,Z) \in \mathsf{iFree}[{}_S U_T], B \subseteq U - T \cong T_P - Z,\\
\bot_{\mathsf{iFree}} = \{(\mathsf{id}_U, T) \mid T \subseteq U \in \square\}, \qquad \top_{\mathsf{iFree}} = \{(P, T_P) \mid P \in \mathsf{iiPoms}\}.
\end{gathered}$$

It is clear that iFree is well-defined; the precubical identities follow easily from Lemmas 5.11 and 5.12. We need some lemmas about existence and uniqueness (up to subsumption) of paths in iFree.

Lemma 8.1. Let P and Q be ipomsets such that $T_Q \cong S_P$, let $Z \subseteq T_P$, and $Y = S_P \cap Z \cong T_Q \cap Z \subseteq T_Q$. There exists a path $\alpha \in \mathsf{Path}(\mathsf{iFree})^{(QP,Z)}_{(Q,Y)}$ with $\mathsf{ev}(\alpha) = P$.

Proof:
We use induction on a step decomposition of P. If $P = \mathsf{id}_{T_Q}$, then $Y = Z$, and $\alpha = ((Q,Y))$ satisfies the required conditions. If $P = P' * {}_A{\uparrow}U$, then $P' = P - A$ and by induction there exists $\beta \in \mathsf{Path}(\mathsf{iFree})^{(QP',Z-A)}_{(PQ,Y)}$ such that $\mathsf{ev}(\beta) = P'$. Thus, $\mathsf{ev}(\beta * ((QP', Z - A) \nearrow^A (QP, Z)) = P$. Finally, if $P = P' * T_{P'}{\downarrow}_B$, $T_P \cong T_{P'} - B$, then $\mathsf{ev}(\beta * ((QP', Z) \searrow_B (QP, Z)) = P$. □

Lemma 8.2. For every $(P,Z) \in \mathsf{iFree}$ and $Y \subseteq S_P$ we have

$$\left\{\mathsf{ev}(\alpha) \;\middle|\; \alpha \in \mathsf{Path}(\mathsf{iFree})^{(P,Z)}_{(\mathsf{id}_{S_P},Y)}\right\} = \begin{cases} \{P\}{\downarrow} & \text{if } Y = S_P \cap Z,\\ \emptyset & \text{otherwise.}\end{cases}$$

Proof:
($\subseteq$). It is enough to show that for every path $\alpha \in \mathsf{Path}(\mathsf{iFree})^{(P,Z)}_{(\mathsf{id}_V,Y)}$ we have $V \cong S_P$, $\mathsf{ev}(\alpha) \sqsubseteq P$ and $Y = S_P \cap Z$. The first statement is clear; the rest we prove by induction on the length of α. If α is constant, then $P = \mathsf{id}_{S_P}$, $Y = Z$, and thus $\mathsf{ev}(\alpha) = \mathsf{id}_{S_P} = P$. If $\alpha = \beta * ((P - A, Z - A) \nearrow^A (P, Z))$ is a concatenation with an upstep, then

$$\mathsf{ev}(\alpha) = \mathsf{ev}(\beta) * {}_A{\uparrow}T_P \overset{\text{ind.}}{\sqsubseteq} (P - A) * {}_A{\uparrow}T_P \overset{\text{L. 5.10}}{\sqsubseteq} P$$

and $Y \overset{\text{ind.}}{=} S_{(P-A)} \cap (Z - A) = S_P \cap Z$, since $A \cap S_P = \emptyset$. If $\alpha = \beta * ((Q,Z) \searrow_B (P,Z))$ and $P \cong Q * T_Q{\downarrow}_B$, then

$$\mathsf{ev}(\alpha) = \mathsf{ev}(\beta) * T_Q{\downarrow}_B \overset{\text{ind.}}{\sqsubseteq} Q * T_Q{\downarrow}_B = P$$

and $Y = S_Q \cap Z = S_P \cap Z$.
($\supseteq$). This follows from Lemma 8.1 for $Q = \mathsf{id}_{S_P}$ and Lemma 7.11. □

Corollary 8.3. $\mathsf{Lang}(\mathsf{iFree}) = \mathsf{iiPoms}$.

Proof:
For every ipomset P there is a path $\alpha \in \mathsf{Path}(\mathsf{iFree})^{(P,T_P)}_{(\mathsf{id}_{S_P},S_P\cap T_P)}$ such that $\mathsf{ev}(\alpha) = P$. □

8.2. $\mathsf{iFree}(L)$

Fix a language L; we will restrict iFree to an iHDA that recognises L. Let $\top_L = \{(P, T_P) \mid P \in L\}$ and define

$$\mathsf{iFree}(L) = \mathsf{ess}((\mathsf{iFree}, \bot_{\mathsf{iFree}}, \top_L)).$$

That is, we restrict accept cells of iFree to the ones that accept ipomsets in L and then reduce to the essential part.

Lemma 8.4. $\mathsf{Lang}(\mathsf{iFree}(L)) = L$.

Proof:
This follows from Lemma 8.2:

$$\mathsf{Lang}(\mathsf{iFree}(L)) = \bigcup_{P \in L} \left\{\mathsf{ev}(\alpha) \mid \alpha \in \mathsf{Path}(\mathsf{iFree})^{(P,T_P)}_{(\mathsf{id}_{S_P}, S_P \cap T_P)}\right\} = \bigcup_{P \in L} \{P\}{\downarrow} = L.$$

□

We provide a description of $\mathsf{iFree}(L)$ in terms of quotient languages. For an ipomset P and $Z \subseteq T_P$ define the partial quotient language by

$$P \setminus^Z L = \{Q \in \mathsf{iiPoms} \mid PQ \in L, S_Q \cap T_Q = Z\} = \{Q \in P \setminus L \mid S_Q \cap T_Q = Z\}.$$

In other words, $P \setminus^Z L$ consists of all "continuations" of P that do not terminate events of Z (and terminate all other target events of P). Obviously, $P \setminus L = \bigsqcup_{Z \subseteq T_P} P \setminus^Z L$.

Example 8.5. Let $L = \{[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}]\}{\downarrow} \cup \{ab\} = \{[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}], ab\bullet, ab\}$. Then $P \setminus^\emptyset L = P \setminus L$ whenever $T_P = \emptyset$, and

$$\begin{aligned} a\bullet \setminus^\emptyset L &= \{[\begin{smallmatrix} \bullet a \\ b\bullet \end{smallmatrix}]\}{\downarrow} \cup \{\bullet ab\}, & a\bullet \setminus^a L &= \emptyset, \\ b\bullet \setminus^\emptyset L &= \emptyset, & b\bullet \setminus^b L &= \{[\begin{smallmatrix} a \\ \bullet b\bullet \end{smallmatrix}]\}, \\ ab\bullet \setminus^\emptyset L &= \{\bullet b\}, & ab\bullet \setminus^b L &= \{\bullet b\bullet\}, \end{aligned}$$

$[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}] \setminus^b L = \{[\begin{smallmatrix} \bullet a \\ \bullet b\bullet \end{smallmatrix}]\}$, and $[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}] \setminus^\emptyset L = [\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}] \setminus^a L = [\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}] \setminus^{a,b} L = \emptyset$.

Lemma 8.6. A cell $(P, Z) \in \mathsf{iFree}$ belongs to $\mathsf{iFree}(L)$ if and only if $P \setminus^Z L \neq \emptyset$.

Proof:
By construction, $(P, Z) \in \mathsf{iFree}$ is accessible. We show that (P, Z) is coaccessible if and only if $P \setminus^Z L \neq \emptyset$. If $P \setminus^Z L \neq \emptyset$, then there is Q such that $PQ \in L$ and $S_Q \cap T_Q = Z$. By Lemma 8.1, there exists a path from (P, Z) to (PQ, T_Q), showing that (P, Z) is coaccessible.

If (P, Z) is coaccessible, then there is a path β in iFree with $\mathsf{src}(\beta) = (P, Z)$ and $\mathsf{tgt}(\beta) \in \top_L$. Let $Q = \mathsf{ev}(\beta)$. We also have a path α from $\bot_{\mathsf{iFree}}$ to (P, Z). The concatenation $\alpha * \beta$ is a path in $\mathsf{iFree}(L)$ with $\mathsf{ev}(\alpha * \beta) = PQ$. Hence $PQ \in L$. Further, $T_Q = \mathsf{ev}(\mathsf{tgt}(\alpha * \beta)) = Z$, since $\mathsf{tgt}(\alpha * \beta)$ is an accept cell. That is, $Q \in P \setminus^Z L$. □

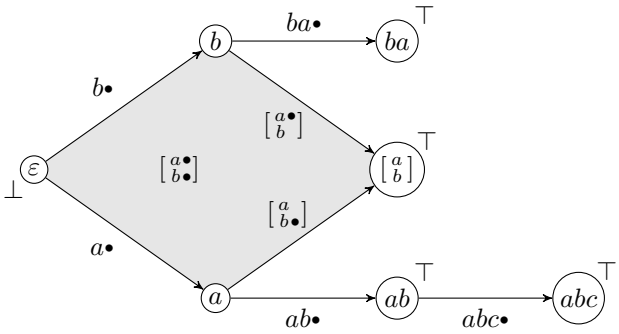


Figure 9. $\mathsf{iFree}(L)$ for $L = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], abc\}{\downarrow}$, see Example 8.7.

Example 8.7. Let $L = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], abc\}{\downarrow} = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], ab, ba, abc\}$ be the language of Example 5.5. We construct $\mathsf{iFree}(L)$. First, note that $\mathsf{iFree}(L)[{}_S U_T] = \emptyset$ if $S \neq \emptyset$, given that $P \setminus L = \emptyset$ if $S_P \neq \emptyset$. Similarly, $\mathsf{iFree}(L)[{}_S U_T] = \emptyset$ if $T \neq \emptyset$, as all ipomsets in L have empty terminating interface.

That is, $\mathsf{iFree}(L)[U]$ is only non-empty for conclists U (without interfaces). For these,

$$\begin{aligned}
\mathsf{iFree}(L)[\emptyset] &= \{(\varepsilon, \emptyset), (a, \emptyset), (b, \emptyset), (ab, \emptyset), (ba, \emptyset), (\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], \emptyset), (abc, \emptyset)\}, \\
\mathsf{iFree}(L)[a] &= \{(a\bullet, \emptyset), (ba\bullet, \emptyset), (\left[\begin{smallmatrix} a\bullet \\ b \end{smallmatrix}\right], \emptyset)\}, \\
\mathsf{iFree}(L)[b] &= \{(b\bullet, \emptyset), (ab\bullet, \emptyset), (\left[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}\right], \emptyset)\}, \\
\mathsf{iFree}(L)[c] &= \{(abc\bullet, \emptyset)\}, \\
\mathsf{iFree}(L)[\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]] &= \{(\left[\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}\right], \emptyset)\}.
\end{aligned}$$

(Compare these with the cells of $\mathsf{MN}(L)$ in Figure 6.) Geometrically, $\mathsf{iFree}(L)$ looks as in Figure 9; note that it is an HDA in the sense of Remark 7.1.

8.3. $\mathsf{iMN}(L)$

The iHDA $\mathsf{iFree}(L)$ is infinite as soon as L is. (It contains at least one accept cell for every element of L.) Analogously to the construction in Section 5.1, we introduce an equivalence relation depending on L. Now however, the relation is not defined on ipomsets but directly on $\mathsf{iFree}(L)$. In order for the quotient iHDA to be well-defined, we will need our equivalence to be a *congruence* in the sense that faces of equivalent cells are again equivalent.

We say that $(P, Z), (Q, Y) \in \mathsf{iFree}(L)[{}_S U_T]$ are *weakly equivalent*, and write $(P, Z) \sim_L (Q, Y)$, if $P \setminus^Z L = Q \setminus^Y L$. (This is the analogue of the relation $\sim_L$ of Section 5.1.) This relation is not necessarily a congruence, for example, if $L = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right], aa\}{\downarrow}$ (*cf.* Example 5.6), then $(aa\bullet, \emptyset) \sim_L (ba\bullet, \emptyset)$ but

$$\delta_a^0(aa\bullet, \emptyset) = (a, \emptyset) \not\sim_L (b, \emptyset) = \delta_a^0(ba\bullet, \emptyset).$$

Thus we introduce the maximal congruence contained in $\sim_L$. Say that $(P, Z), (Q, Y) \in \mathsf{iFree}(L)[{}_S U_T]$ are *strongly equivalent*, denoted $(P, Z) \approx_L (Q, Y)$, if

- $\delta_A^0(P, Z) \sim_L \delta_A^0(Q, Y)$ for all $A \subseteq U - S$, and

- $\delta^1_B(P,Z) \sim_L \delta^1_B(Q,Y)$ for all $B \subseteq U - T$.

The first is equivalent to the condition $(P - A) \backslash^{Z-A} L = (Q - A) \backslash^{Y-A} L$ for every $A \subseteq U - S$ (*cf.* the conditions for $\approx_L$ in Section 5.1); and the latter is always satisfied.

It is obvious that every congruence contained in $\sim_L$ must be contained in $\approx_L$ as well. Below we show that $\approx_L$ is indeed a congruence, hence the biggest congruence contained in $\sim_L$, and describe its quotient iHDA.

Lemma 8.8. Let $(P,Z),(Q,Y) \in \mathsf{iFree}(L)[_S U_T]$. If $(P,Z) \approx_L (Q,Y)$, then

1. $\delta^0_A(P,Z) \approx_L \delta^0_A(Q,Y)$ for $A \subseteq U - S$;
2. $\delta^1_B(P,Z) \approx_L \delta^1_B(Q,Y)$ for $B \subseteq U - T$;
3. $(P,Z) \in \bot_{\mathsf{iFree}(L)} \implies (P,Z) = (Q,Y)$,
4. $(P,Z) \in \top_{\mathsf{iFree}(L)} \iff (Q,Y) \in \top_{\mathsf{iFree}(L)}$.

Proof:
1. We have $\delta^0_A(P,Z) = (P - A, Z - A), \delta^0_A(Q,Y) = (Q - A, Y - A) \in \mathsf{iFree}[(S, U - A, T - A)]$. For every $C \subseteq U - (S \cup A)$,

$$\begin{aligned}(P - A) - C \backslash^{(Z-A)-C} L &= P - (A \cup C) \backslash^{Z-(A\cup C)} L \\ &\approx_L Q - (A \cup C) \backslash^{Y-(A\cup C)} L = (Q - A) - C \backslash^{(Y-A)-C} L.\end{aligned}$$

2. We have $\delta^1_B(P,Z) = (P * U{\downarrow_B}, Z), \delta^1_B(Q,Y) = (Q * U{\downarrow_B}, Y) \in \mathsf{iFree}[(S - B, U - B, T)]$. For every $C \subseteq U - (S \cup B)$,

$$\begin{aligned}(P,Z) \approx_L (Q,Z) &\implies (P - C) \backslash^{Z-C} L = (Q - C) \backslash^{Y-C} L \\ &\overset{\text{L. 5.9}}{\implies} (P - C) * (U - C){\downarrow_B} \backslash^{Z-C} L = (Q - C) * (U - C){\downarrow_B} \backslash^{Y-C} L \\ &\overset{\text{L. 5.11}}{\iff} (P * U{\downarrow_B} - C) \backslash^{Z-C} L = (Q * U{\downarrow_B} - C) \backslash^{Y-C} L.\end{aligned}$$

3. If $(P,Z) \in \bot_{\mathsf{iFree}(L)}$, then $P = \mathsf{id}_U$ and $Z = U = T$: the only start cell in $\mathsf{iFree}(L)[_S U_T]$.
4. If $(P,Z) \in \top_{\mathsf{iFree}(L)}$, then $Z = T_P = T$ and $\mathsf{id}_T \in P \backslash^Z L$. Since $(P,Z) \approx_L (Q,Y)$, we have $\mathsf{id}_{T_Q} \cong \mathsf{id}_T \in Q \backslash^Z L$, and $Z = Y = T_Q$. Thus, $(Y,Z) \in \top_{\mathsf{iFree}(L)}$. □

We may thus define the iHDA $\mathsf{iMN}(L)$ as the quotient of $\mathsf{iFree}(L)$ by $\approx_L$:

$$\mathsf{iMN}(L)[_S U_T] = \mathsf{iFree}(L)[_S U_T]/{\approx_L},$$
$$\bot_{\mathsf{iMN}(L)} = \{\langle (P,Z)\rangle \mid (P,Z) \in \bot_{\mathsf{iFree}(L)}\}, \qquad \top_{\mathsf{iMN}(L)} = \{\langle (P,Z)\rangle \mid (P,Z) \in \top_{\mathsf{iFree}(L)}\}.$$

Remark 8.9. If all ipomsets in L have empty interfaces, then $\mathsf{iMN}(L)[_S U_T] = \emptyset$ unless $S = T = \emptyset$ (*cf.* Example 8.7). Further, $\mathsf{iMN}(L)[_\emptyset U_\emptyset] = \mathsf{ess}(\mathsf{MN}(L))[U]$, so both constructions effectively coincide. We will see below that this is not the case if L contains ipomsets with non-empty interfaces.

8.4. Examples

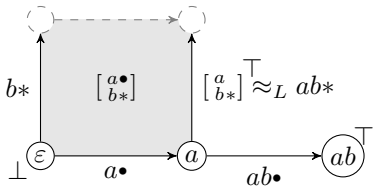
Cells of iHDAs $\mathsf{iMN}(L)$ correspond to equivalence classes of pairs (P, Z) for $Z \subseteq P$. For greater clarity, in the examples below we label a cell $\langle (P, Z) \rangle$ only by the ipomset P but mark target events belonging to Z by asterisks instead of bullets: for example $(a\bullet, \{a\})$ is written as $a*$ and $(a\bullet, \emptyset)$ as $a\bullet$.

Example 8.10. Let $L = \{\bullet a\bullet\} \cup \{\bullet a a^n a\bullet \mid n \geq 0\}$. Then $\mathsf{iMN}(L)$ is the iHDA from Example 7.2, and

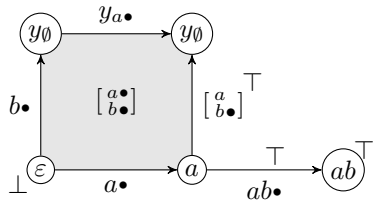
$$e_1 = \bullet a* = \langle (\bullet a\bullet, a) \rangle, \qquad e_2 = \bullet a\bullet = \langle (\bullet a\bullet, \emptyset) \rangle, \qquad e_3 = \{(a^n a\bullet, \emptyset) \mid n \geq 0\},$$
$$e_4 = \{(a^n a\bullet, a) \mid n \geq 0\}, \qquad x = \{(\bullet a a^n, \emptyset) \mid n \geq 0\}.$$

Example 8.11. For the language $L = \{[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}]\}{\downarrow} \cup \{ab\} = \{[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}], ab\bullet, ab\}$ of Example 8.5, $\mathsf{iMN}(L)$ and $\mathsf{MN}(L)$ are as follows:

$\mathsf{iMN}(L)$:

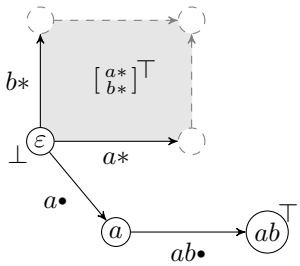


$\mathsf{MN}(L)$:



Note that $[\begin{smallmatrix} a \\ b* \end{smallmatrix}] = ([\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}], \{b\}) \approx_L (ab\bullet, \{b\}) = ab*$, but $[\begin{smallmatrix} a \\ b\bullet \end{smallmatrix}] \not\approx_L ab\bullet$: the relation $\approx_L$ on cells of $\mathsf{iFree}(L)$ is finer than the strong equivalence $\approx_L$ on ipomsets used in the construction of $\mathsf{MN}(L)$ in Section 5.1.

Example 8.12. Let $L = \{ab, [\begin{smallmatrix} a\bullet \\ b\bullet \end{smallmatrix}]\}$. Then $\mathsf{iMN}(L)$ is the same as $\mathsf{iFree}(L)$ and looks as follows:



Example 8.13. Let $L = \{\bullet a\bullet\} \cup \{[\begin{smallmatrix} \bullet aa\bullet \\ b \end{smallmatrix}]^n \mid n \geq 1\}{\downarrow}$ be the language of Example 5.23, then $\mathsf{iMN}(L)$ is displayed in Figure 10. Blue arrows marked e are identified, as well as their corresponding endpoints. We have $e = \langle [\begin{smallmatrix} \bullet a \\ b\bullet \end{smallmatrix}] \rangle = \langle ab\bullet \rangle$ and $\mathsf{ev}(e) = {}_\emptyset b_\emptyset$.

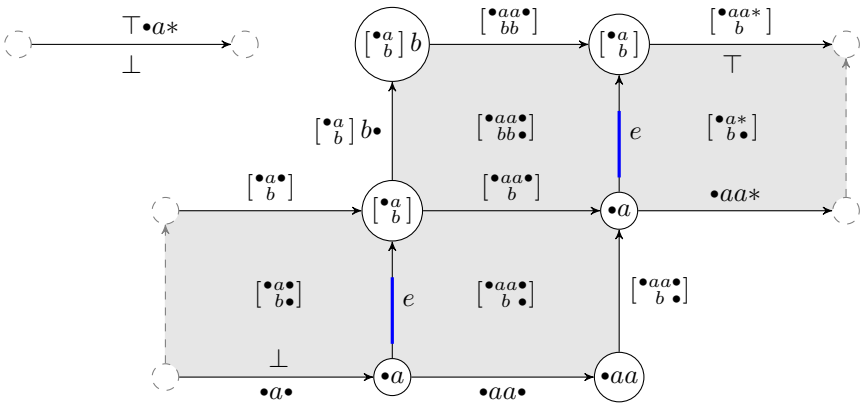


Figure 10. $\mathsf{iMN}(L)$ for $L = \{\bullet a \bullet\} \cup \{[{}^{\bullet aa\bullet}_{b}]^n \mid n \geq 1\}{\downarrow}$, see Example 8.13.

8.5. $\mathsf{iMN}(L)$ recognises L

For a cell $x \in \mathsf{iMN}(L)$ denote $x \setminus L := P \setminus^Z L$ for any $(P, Z) \in x$. This clearly does not depend on the choice of a representative.

Lemma 8.14. Assume that $x \in \mathsf{iMN}(L)[{}_S U_T]$, $A \subseteq U - S$, $B \subseteq U - T$. Then

1. $Q \in x \setminus L \implies {}_A{\uparrow}U * Q \in \delta^0_A(x) \setminus L$,
2. $U{\downarrow}_B * Q \in x \setminus L \iff Q \in \delta^1_B(x) \setminus L$.

Proof:
Fix $(P, Z) \in x$. Recall that $(S_Q, S_Q, S_Q \cap T_Q) \cong {}_U U_Z$ for every $Q \in x \setminus L$. For the first part,

$$\begin{aligned} Q \in x \setminus L \iff Q \in P \setminus^Z L &\iff PQ \in L \\ &\implies (P - A) * {}_A{\uparrow}U * Q \in L \qquad \text{(Lem. 5.1 \& 5.10)} \\ &\iff {}_A{\uparrow}U * Q \in (P - A) \setminus^{Z-A} L \\ &\iff {}_A{\uparrow}U * Q \in \delta^0_A(x) \setminus L. \end{aligned}$$

For the second part of the lemma,

$$\begin{aligned} Q \in \delta^1_B(x) \setminus L \iff Q \in (P * U{\downarrow}_B) \setminus^Z L &\iff P * U{\downarrow}_B * Q \in L \\ &\iff U{\downarrow}_B * Q \in P \setminus^Z L \iff U{\downarrow}_B * Q \in x \setminus L. \end{aligned}$$

□

Lemma 8.15. If $\alpha \in \mathsf{Path}(\mathsf{iMN}(L))_\bot$ and $\mathsf{tgt}(\alpha) \in \mathsf{iMN}(L)[{}_S U_T]$, then $\mathsf{tgt}(\alpha) \setminus L \subseteq \mathsf{ev}(\alpha) \setminus^T L$.

Proof:
Induction on the length of the path α. If $\alpha = (x)$ for $x = \langle(\mathsf{id}_U, T)\rangle$, then

$$x \setminus L = (\mathsf{id}_U) \setminus^T L = \mathsf{ev}(\alpha) \setminus^T L.$$

If $\alpha = \beta * (\delta^0_A(x) \nearrow^A x)$ for $x \in \mathsf{iMN}(L)[_S U_T]$ and $A \subseteq U - S$, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta) *_A\!\uparrow U$ and

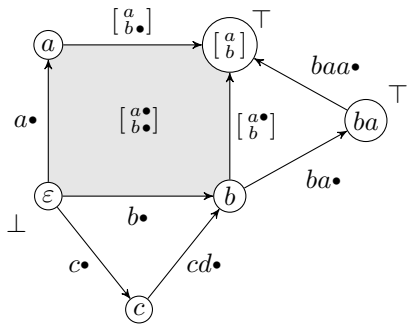
$$Q \in x \setminus L \overset{\text{Lem. 8.14}}{\Longrightarrow} {}_A\!\uparrow U * Q \in \delta^0_A(x) \setminus L \overset{\text{ind.}}{\Longrightarrow} {}_A\!\uparrow U * Q \in \mathsf{ev}(\beta) \setminus^{T-A} L \iff Q \in \mathsf{ev}(\alpha) \setminus^T L.$$

If $\alpha = \beta * (x \searrow_B \delta^1_B(x))$, then $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta) * U\!\downarrow_B$ and

$$Q \in \delta^1_B(x) \setminus L \overset{\text{Lem. 8.14}}{\iff} U\!\downarrow_B * Q \in x \setminus L \overset{\text{ind.}}{\Longrightarrow} U\!\downarrow_B * Q \in \mathsf{ev}(\beta) \setminus^T L \iff Q \in \mathsf{ev}(\alpha) \setminus^T L.$$

□

Example 8.16. Let $L = \{[\begin{smallmatrix} a \\ b \end{smallmatrix}]\}\!\downarrow \cup \{baa, cda, cdaa\}$, then $\mathsf{iMN}(L)$ is as follows:



Note that there are two paths recognising cda. One of them ends at $[\begin{smallmatrix} a \\ b \end{smallmatrix}]$, yet there is no P such that $cda \sqsubseteq P$ and $P \approx_L [\begin{smallmatrix} a \\ b \end{smallmatrix}]$. This explains why Lemma 8.15 cannot be strengthened.

Lemma 8.17. Let $x \in \mathsf{iMN}(L)[_S U_T]$. Then $x \in \top_{\mathsf{iMN}(L)}$ if and only if $\mathsf{id}_T \in x \setminus L$.

Proof:
We have $x \in \top_{\mathsf{iMN}(L)}$ if and only if there exists an ipomset $P \in L$ such that $x = \langle (P, T_P) \rangle$ and $T_P \cong U = T$. But $P \in L$ if and only if $\mathsf{id}_T \in P \setminus^T L = x \setminus L$. □

Proposition 8.18. $\mathsf{Lang}(\mathsf{iMN}(L)) = L$.

Proof:
The quotient map $\mathsf{iFree}(L) \to \mathsf{iMN}(L)$ is an iHDA-map, hence it induces an inclusion of languages. By Lemma 8.4, $L = \mathsf{Lang}(\mathsf{iFree}(L)) \subseteq \mathsf{iMN}(L)$.

For the other direction, let α be an accepting path in $\mathsf{iMN}(L)$. Since $\mathsf{tgt}(\alpha) \in \mathsf{iMN}(L)[_S U_U]$ is an accept cell, there exists $Q \in L$ such that $\mathsf{tgt}(\alpha) = \langle (Q, T_Q) \rangle$ with $T_Q \cong U$. Thus, by Lemma 8.17, $\mathsf{id}_U \in Q \setminus^U L = \mathsf{tgt}(\alpha) \setminus L$. By Lemma 8.15, $\mathsf{id}_U \in \mathsf{tgt}(\alpha) \setminus L \subseteq \mathsf{ev}(\alpha) \setminus^U L$, which implies $\mathsf{ev}(\alpha) \in L$. This proves $\mathsf{Lang}(\mathsf{iMN}(L)) \subseteq L$. □

9. Determinism in iHDAs

The notion of determinism for iHDAs is different from the one for HDAs, given that we do not have to restrict to essential cells. Yet we will show that languages recognised by deterministic HDAs and deterministic iHDAs are the same.

Definition 9.1. An iHDA X is *deterministic* if

1. for every ${}_UU_T \in \mathbb{I}\square$ there is at most one start cell in $X[{}_UU_T]$, and
2. for every ${}_SU_T \in \mathbb{I}\square$, $A \subseteq U - S$ and $x \in X[(S, U - A, T - A)]$, there is at most one cell $y \in X[{}_SU_T]$ such that $x = \delta^0_A(y)$.

Compared to deterministic *HDAs*, we now allow one start cell for every pair $U \supseteq T$ of source interface and target interface. That is because the information of which events may be terminated in an accepting path is already contained in the target interface of its source cell, *cf.* Lemma 7.5.

Lemma 9.2. If HDA X is deterministic, then the iHDA $\mathsf{ess}(\mathsf{Res}(X))$ is also deterministic.

Proof:
The first condition is clear. To prove the second, fix ${}_SU_T \in \mathbb{I}\square$, $A \subseteq U - S$ and $(x; S, T - A) \in \mathsf{ess}(\mathsf{Res}(X))[(S, U - A, T - A)]$. There is at most one essential $y \in X[U]$ such that $\delta^0_A(y) = x$.

Let $(z; S, T) \in \mathsf{ess}(\mathsf{Res}(X))[{}_SU_T]$ such that $\delta^0_A(z; S, T) = (x, S, T - A)$. By definition, $\delta^0_A(z) = x$, and by Lemma 7.7 we obtain that z is essential; as a consequence, $y = z$. $\square$

From Lemmas 7.8 and 9.2 we conclude:

Corollary 9.3. If L is recognised by a deterministic HDA, then it is recognised by a deterministic iHDA.

The following two lemmas provide analogues to the unambiguity Lemma 6.8 for deterministic HDAs.

Lemma 9.4. Let X be a deterministic iHDA and $\alpha, \beta \in \mathsf{Path}(X)^\top_\bot$. If $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$, then $\alpha \simeq \beta$.

Proof:
Denote $P = \mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$. Without loss of generality we may assume that $\alpha = \alpha_1 * \cdots * \alpha_n$ and $\beta = \beta_1 * \cdots * \beta_m$ are sparse. We show that $n = m$ and $\alpha_k = \beta_k$ for all k which implies the claim.

Both $P = \mathsf{ev}(\alpha_1) * \cdots * \mathsf{ev}(\alpha_n)$ and $P = \mathsf{ev}(\beta_1) * \cdots * \mathsf{ev}(\beta_m)$ are sparse step decomposition of P. Proposition 3.5 implies that $m = n$ and $\mathsf{ev}(\alpha_k) = \mathsf{ev}(\beta_k)$ for every k.

Denote $x_0 = \mathsf{src}(\alpha) = \mathsf{src}(\alpha_1)$, $x_k = \mathsf{tgt}(\alpha_k) = \mathsf{src}(\alpha_{k+1})$, $x_n = \mathsf{tgt}(\alpha) = \mathsf{tgt}(\alpha_n)$ and $y_0 = \mathsf{src}(\beta) = \mathsf{src}(\beta_1)$, $y_k = \mathsf{tgt}(\beta_k) = \mathsf{src}(\beta_{k+1})$, $y_n = \mathsf{tgt}(\beta) = \mathsf{tgt}(\beta_n)$. Fix k and denote $\mathsf{iev}(x_k) = (S, U, T)$. By Lemma 7.5.1, (S, U) is determined by $Q = \mathsf{ev}(\alpha_1 * \cdots * \alpha_k)$, and by Lemma 7.5.2, (U, T) is determined by $R = \mathsf{ev}(\alpha_{k+1} * \cdots * \alpha_n)$. Similarly, $\mathsf{iev}(y_k)$ is determined by $\mathsf{ev}(\beta_1 * \cdots * \beta_k)$ and $\mathsf{ev}(\beta_{k+1} * \cdots * \beta_n)$. As a consequence, $\mathsf{iev}(x_k) = \mathsf{iev}(y_k)$ for all k.

We show by induction that $x_k = y_k$. Since X is deterministic, $\mathsf{iev}(x_0) = \mathsf{iev}(y_0)$ implies $x_0 = y_0$. For $k > 0$ assume that $x_{k-1} = y_{k-1}$. If $\mathsf{ev}(\alpha_k) = \mathsf{ev}(\beta_k) = {}_A{\uparrow}U$ is a starter, then conditions $\mathsf{iev}(x_k) = \mathsf{iev}(y_k)$ and $\delta^0_A(x_k) = \delta^0_A(y_k)$ imply $x_k = y_k$ by determinism of X. If $\mathsf{ev}(\alpha_k) = \mathsf{ev}(\beta_k) = U{\downarrow}_B$ is a terminator, then $x_k = \delta^1_B(x_{k-1}) = \delta^1_B(y_{k-1}) = y_k$. □

Lemma 9.5. Let X be a deterministic iHDA and $\alpha, \beta \in \mathsf{Path}(X)_\perp$. If $\mathsf{ev}(\alpha) = \mathsf{ev}(\beta)$ and $\mathsf{iev}(\mathsf{tgt}(\alpha)) = \mathsf{iev}(\mathsf{tgt}(\beta))$, then $\alpha \simeq \beta$.

Proof:
Again, we assume that α and β are sparse. Denote $x = \mathsf{tgt}(\alpha)$, $y = \mathsf{tgt}(\beta)$, ${}_SU_T = \mathsf{iev}(x) = \mathsf{iev}(y)$. Modify X by adding accept cells

$$x' = \delta^1_{U-T}(x), y' = \delta^1_{U-T}(y) \in X[{}_SU_U].$$

The paths $\alpha' = \alpha * (x \searrow_{U-T} x')$ and $\beta' = \beta * (y \searrow_{U-T} y')$ are accepting, and $\mathsf{ev}(\alpha') = \mathsf{ev}(\beta') = \mathsf{ev}(\alpha) * U{\downarrow}_{U-T}$. Let α'' and β'' be sparse paths that are equivalent to α' and β', respectively. By Lemma 9.4, $\alpha'' = \beta''$ and thus both α' and β' are refinements of $\gamma := \alpha'' = \beta''$. If $U - T = \emptyset$, then $\alpha = \beta = \gamma$. Otherwise, decompose $\gamma = \gamma' * \omega$, where ω is the last step of γ. Then ω is a downstep $(z \searrow_B x')$ such that $U - T \subseteq B$ and $\alpha = \beta = \gamma' * (z \searrow_{B-T} \delta^1_{B-T}(z))$. □

Lemma 9.6. Let X be a deterministic iHDA and $\alpha, \beta \in \mathsf{Path}(X)_\perp$. If $\mathsf{ev}(\alpha) \sqsubseteq \mathsf{ev}(\beta)$ and $\mathsf{iev}(\mathsf{tgt}(\alpha)) = \mathsf{iev}(\mathsf{tgt}(\beta))$, then $\mathsf{tgt}(\alpha) = \mathsf{tgt}(\beta)$.

Proof:
By Lemma 7.11 there exists $\alpha' \in \mathsf{Path}(X)^{\mathsf{tgt}(\beta)}_{\mathsf{src}(\beta)}$ such that $\mathsf{ev}(\alpha') = \mathsf{ev}(\alpha)$. From Lemma 9.5 follows that $\mathsf{tgt}(\alpha) = \mathsf{tgt}(\alpha') = \mathsf{tgt}(\beta)$. □

Proposition 9.7. If L is recognised by a deterministic iHDA, then L is swap-invariant.

Proof:
Like the proof of Proposition 6.10, swapping out the applications of Lemma 4.13 with Lemma 7.12 and the one of Lemma 6.9 with Lemma 9.6. □

Together with Theorem 6.4, Corollary 9.3 and Proposition 9.7 now imply that a language is recognised by a deterministic iHDA if and only if it is swap-invariant.

10. Conclusion and Further Work

We have proven a Myhill-Nerode type theorem for higher-dimensional automata (HDAs), stating that a language is regular if and only if it has finite prefix quotient. We have also introduced deterministic HDAs and shown that not all finite HDAs are determinizable. Lastly, we have seen that both notions are somewhat simpler when using higher-dimensional automata with interfaces (iHDAs), given that no restrictions to essential parts are necessary.

HDAs are arguably simpler than iHDAs, and also somewhat more standard as a model for concurrent computations. On the other hand, we have seen in [9, 10] and now also here that because of the structural axioms of HDAs, certain concepts are easier to state and prove for iHDAs than for HDAs. This same observation has led to the introduction of partial HDAs in [7, 12], of which iHDAs are a more restricted event-based version. In particular, it appears that the trees of Dubut's [7] are related to some of our iHDA constructions developed here.

Our Myhill-Nerode theorem provides a language-internal criterion for whether a language is regular, and we have developed a similar one to distinguish deterministic languages. Another important aspect is the *decidability* of these questions, together with other standard problems such as membership or language inclusion. Together with coauthors A. Amrane and H. Bazille, we show in [1] that these are decidable.

Given that we have shown that not all regular languages are deterministic, one might ask for the approximation of deterministic languages by other, less restrictive notions. It is shown in [1] that non-deterministic HDAs may exhibit unbounded ambiguity, but other approaches such as for example history-determinism [4] or residuality [6] remain to be explored. It appears that our Myhill-Nerode HDAs may be residual in some sense, which would open connections to for example automata learning [2, 5, 31].

Acknowledgement. We are indebted to members and associates of the (i)Po(m)set Project[1] for numerous discussions regarding the subjects of this paper; any errors, however, are exclusively ours.

References

[1] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. In Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa, editors, *ICTAC*, volume 14446 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2023. `https://arxiv.org/abs/2305.02873`.

[2] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[3] Marek A. Bednarczyk. *Categories of Asynchronous Systems*. PhD thesis, University of Sussex, UK, 1987.

[4] Udi Boker and Karoliina Lehtinen. When a little nondeterminism goes a long way: An introduction to history-determinism. *ACM SIGLOG News*, 10(1):24–51, 2023.

[5] Benedikt Bollig, Peter Habermehl, Carsten Kern, and Martin Leucker. Angluin-style learning of NFA. In Craig Boutilier, editor, *IJCAI*, pages 1004–1009, 2009.

[6] François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. In Afonso Ferreira and Horst Reichel, editors, *STACS*, volume 2010 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2001.

[7] Jérémy Dubut. Trees in partial higher dimensional automata. In Mikołaj Bojańczyk and Alex Simpson, editors, *FOSSACS*, volume 11425 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2019.

[1] `https://ulifahrenberg.github.io/pomsetproject/`

[8] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021. `https://arxiv.org/abs/2103.07557`.

[9] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. A Kleene theorem for higher-dimensional automata. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *CONCUR*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[10] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Kleene theorem for higher-dimensional automata. *CoRR*, abs/2202.03791, 2022. `https://arxiv.org/abs/2202.03791`. Long version of [9].

[11] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285(B):104914, 2022. `https://arxiv.org/abs/2106.10895`.

[12] Uli Fahrenberg and Axel Legay. Partial higher-dimensional automata. In Lawrence S. Moss and Pawel Sobocinski, editors, *CALCO*, volume 35 of *Leibniz International Proceedings in Informatics*, pages 101–115. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[13] Uli Fahrenberg and Krzysztof Ziemiański. A Myhill-Nerode theorem for higher-dimensional automata. In Luís Gomes and Robert Lorenz, editors, *PETRI NETS*, volume 13929 of *Lecture Notes in Computer Science*, pages 167–188. Springer, 2023.

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

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

[16] Ryszard Janicki and Maciej Koutny. Structure of concurrency. *Theoretical Computer Science*, 112(1):5–52, 1993.

[17] Ryszard Janicki and Maciej Koutny. Operational semantics, interval orders and sequences of antichains. *Fundamentae Informatica*, 169(1-2):31–55, 2019.

[18] Christian Johansen. ST-structures. *Journal of Logic and Algebraic Methods in Programming*, 85(6):1201–1233, 2015. `https://arxiv.org/abs/1406.0641`.

[19] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13:85–108, 1981.

[20] Carl A. Petri. *Kommunikation mit Automaten*. Number 2 in Schriften des IIM. Institut für Instrumentelle Mathematik, Bonn, 1962.

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

[22] Vaughan R. Pratt. Chu spaces and their interpretation as concurrent objects. In *Computer Science Today: Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 392–405. Springer, 1995.

[23] Vaughan R. Pratt. Transition and cancellation in concurrency and branching time. *Mathematical Structures in Computer Science*, 13(4):485–529, 2003.

[24] Mike W. Shields. Concurrent machines. *Comput. J.*, 28(5):449–465, 1985.

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

[26] Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. See also [27].

[27] Rob J. van Glabbeek. Erratum to "On the expressiveness of higher dimensional automata". *Theoretical Computer Science*, 368(1-2):168–194, 2006.

[28] Rob J. van Glabbeek and Ursula Goltz. Refinement of actions and equivalence notions for concurrent systems. *Acta Informatica*, 37(4/5):229–327, 2001.

[29] Rob J. van Glabbeek and Gordon D. Plotkin. Configuration structures. In *LICS*, pages 199–209. IEEE Computer Society, 1995.

[30] Rob J. van Glabbeek and Gordon D. Plotkin. Configuration structures, event structures and Petri nets. *Theoretical Computer Science*, 410(41):4111–4159, 2009.

[31] Gerco van Heerdt, Tobias Kappé, Jurriaan Rot, and Alexandra Silva. Learning pomset automata. In Stefan Kiefer and Christine Tasson, editors, *FOSSACS*, volume 12650 of *Lecture Notes in Computer Science*, pages 510–530. Springer, 2021.