Elements of Higher-Dimensional Automata Theory

Uli Fahrenberg

LMF, Université Paris-Saclay, France

LMF Seminar, October 2025

Language theory of higher-dimensional automata

- Languages of Higher-Dimensional Automata [MSCS 2021]
- Kleene Theorem for Higher-Dimensional Automata [LMCS 2024]
- Myhill-Nerode Theorem for Higher-Dimensional Automata [FI 2024]
- Decision and Closure Properties for Higher-Dimensional Automata [TCS 2025]

- Logic and Languages of Higher-Dimensional Automata [DLT 2024]
- Presenting Interval Pomsets with Interfaces [RAMiCS 2024]
- Bisimulations and Logics for Higher-Dimensional Automata. [ICTAC 2024]
- Higher-Dimensional Automata: Extension to Infinite Tracks. [FSCD 2025]
- Kamp Theorem for Higher Dimensional Automata. [CSL 2026]

Today:

1. What are HDAs?
2. What are languages of HDAs?
3. What can I do with languages of HDAs?

Nice people

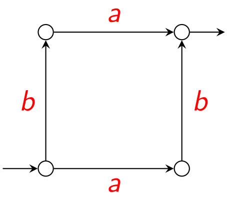
- Eric Goubault, Paris
- Christian Johansen, Oslo
- Georg Struth, Sheffield
- Krzysztof Ziemiański, Warsaw

- Amazigh Amrane (LRE), Hugo Bazille (LRE), Paul Brunet (LACL), Emily Clement (LIPN), Jérémy Dubut (LIX), Marie Fortin (IRIF), Jérémy Ledent (IRIF), ...

- See also `https://ulifahrenberg.github.io/pomsetproject/`

Higher-dimensional automata

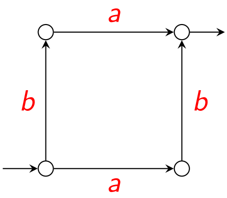
semantics of "*a* parallel *b*":



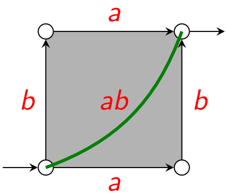
$a.b + b.a$

Higher-dimensional automata

semantics of "a parallel b":



$a.b + b.a$



a and b are independent

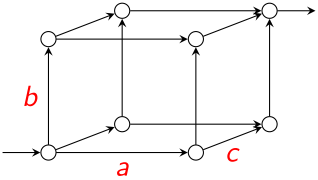
Higher-dimensional automata & concurrency

HDAs as a model for concurrency:

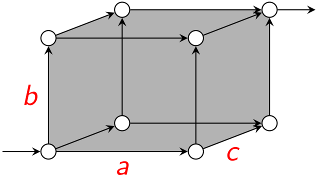
- points: states
- edges: transitions
- squares, cubes etc.: independency relations / concurrently executing events
- two-dimensional automata $\cong$ asynchronous transition systems [Bednarczyk]

[van Glabbeek 2006, TCS]: Up to history-preserving bisimilarity, HDAs “generalize the main models of concurrency proposed in the literature” (notably, event structures and Petri nets)

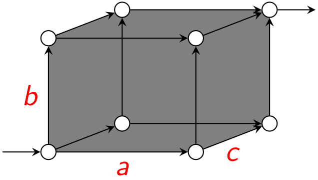
Examples



no concurrency



two out of three



full concurrency

Higher-dimensional automata

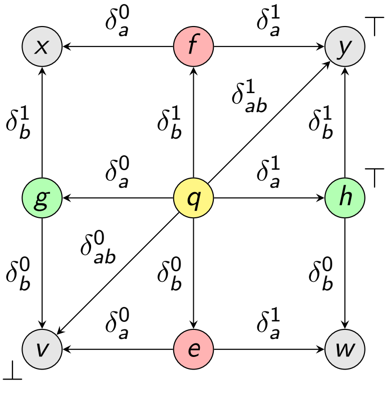
A conclist is a finite, ordered and Σ-labelled set. (a list of events)

A precubical set X consists of:

- A set of cells X (cubes)
- Every cell $x \in X$ has a conclist $\mathrm{ev}(x)$ (list of events active in x)
- We write $X[U] = \{x \in X \mid \mathrm{ev}(x) = U\}$ for a conclist U (cells of type U)
- For every conclist U and $A \subseteq U$ there are:
  upper face map $\delta_A^1 : X[U] \to X[U \setminus A]$ (terminating events A)
  lower face map $\delta_A^0 : X[U] \to X[U \setminus A]$ ("unstarting" events A)
- Precube identities: $\delta_A^\mu \delta_B^\nu = \delta_B^\nu \delta_A^\mu$ for $A \cap B = \emptyset$ and $\mu, \nu \in \{0, 1\}$

A higher dimensional automaton (HDA) is a precubical set X with start cells $\perp \subseteq X$ and accept cells $\top \subseteq X$ (not necessarily vertices)

Example



$X[\emptyset] = \{v, w, x, y\}$

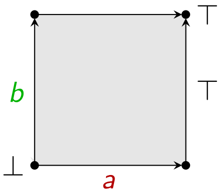
$X[a] = \{e, f\}$

$X[b] = \{g, h\}$

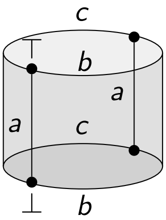
$X[ab] = \{q\}$

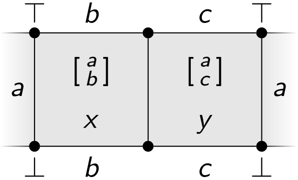
$\perp_X = \{v\}$

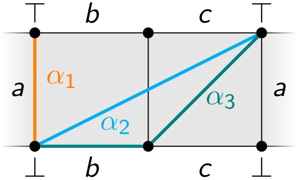
$\top_X = \{h, y\}$



Another one







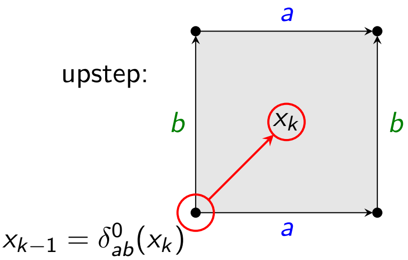
$$a \parallel (bc)^*$$

Computations of HDAs

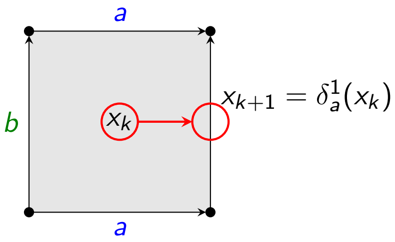
A path on an HDA X is a sequence $(x_0, \varphi_1, x_1, \dots, x_{n-1}, \varphi_n, x_n)$ such that for every k, $(x_{k-1}, \varphi_k, x_k)$ is either

- $(\delta_A^0(x_k), \nearrow^A, x_k)$ for $A \subseteq \text{ev}(x_k)$ or (upstep: start A)
- $(x_{k-1}, \searrow_B, \delta_B^1(x_{k-1}))$ for $B \subseteq \text{ev}(x_{k-1})$ (downstep: terminate B)

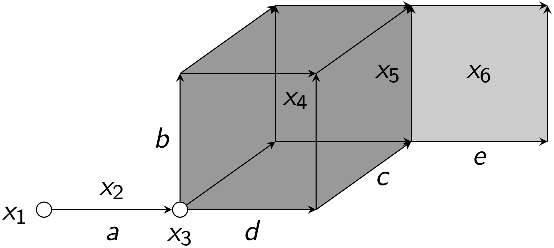
upstep:



downstep:

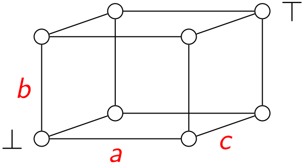


Example

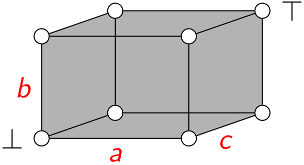


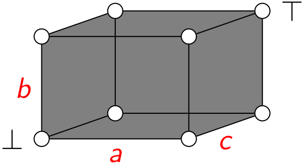
$$\left(x_1 \nearrow^{a} x_2 \searrow_{a} x_3 \nearrow^{\{b,c,d\}} x_4 \searrow_{\{c,d\}} x_5 \nearrow^{e} x_6\right)$$

Languages of HDAs: Examples



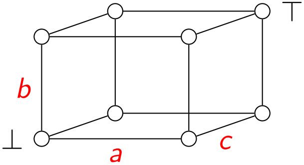
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$



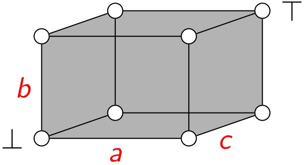


$$L_3 = \left\{ \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \dots \right\}$$

Languages of HDAs: Examples

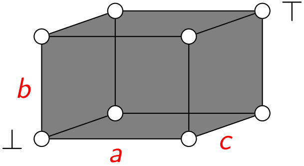


$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$



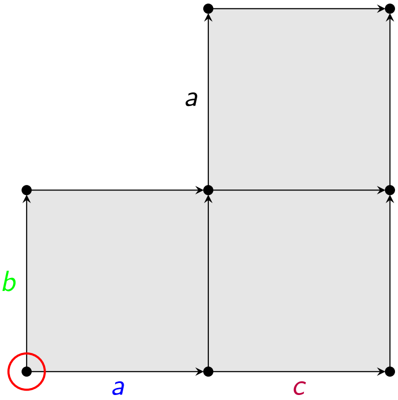
$$L_2 = \left\{ \begin{bmatrix} a \\ b \to c \end{bmatrix}, \begin{bmatrix} a \\ c \to b \end{bmatrix}, \begin{bmatrix} b \\ a \to c \end{bmatrix}, \begin{bmatrix} b \\ c \to a \end{bmatrix}, \begin{bmatrix} c \\ a \to b \end{bmatrix}, \begin{bmatrix} c \\ b \to a \end{bmatrix} \right\} \cup L_1 \cup \dots$$

sets of pomsets

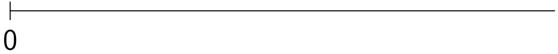


$$L_3 = \left\{ \begin{bmatrix} a \\ b \\ c \end{bmatrix} \right\} \cup L_2$$

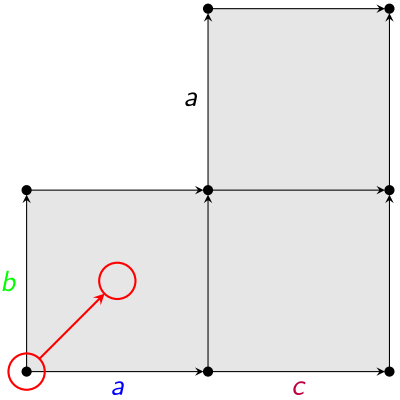
Event ipomset of a path



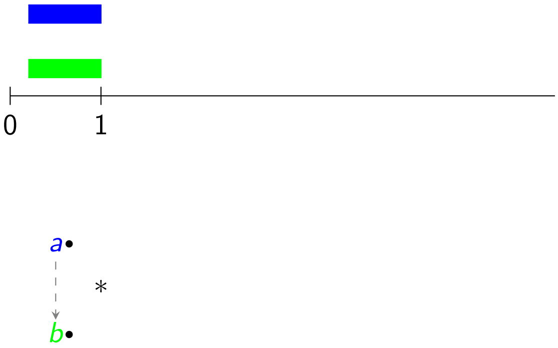
Lifetimes of events



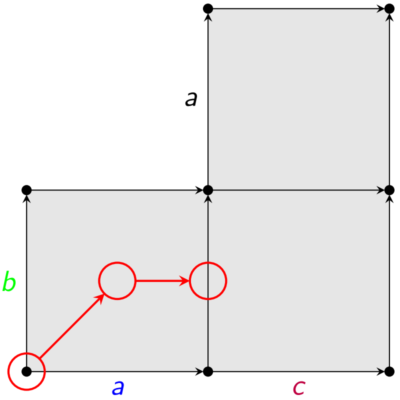
Event ipomset of a path

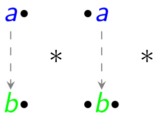


Lifetimes of events

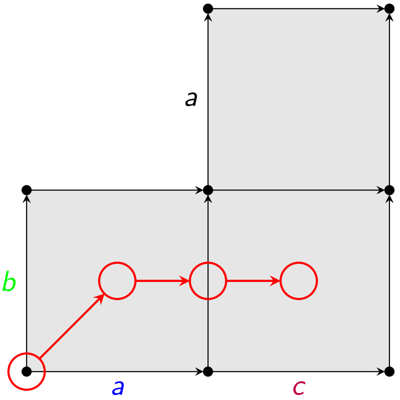


Event ipomset of a path

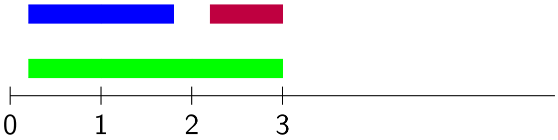


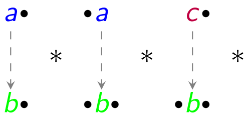


Event ipomset of a path

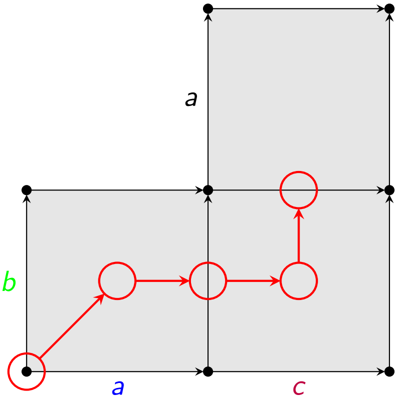


Lifetimes of events

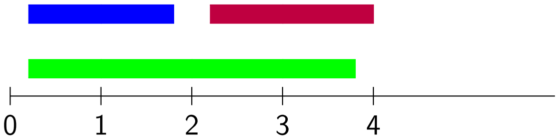


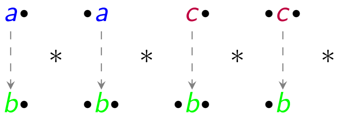


Event ipomset of a path

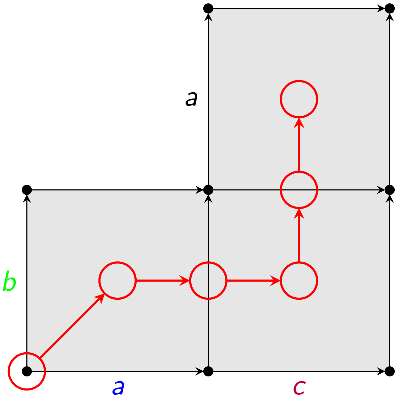


Lifetimes of events

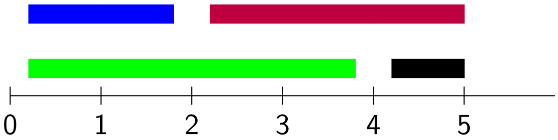


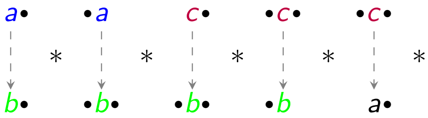


Event ipomset of a path

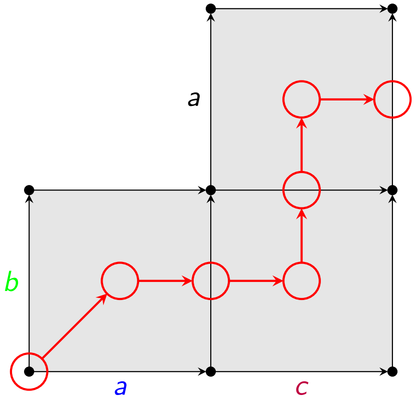


Lifetimes of events

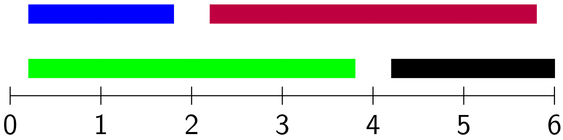




Event ipomset of a path

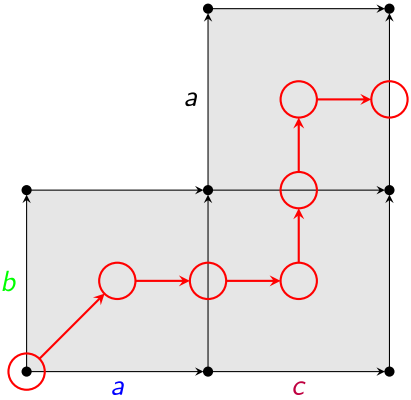


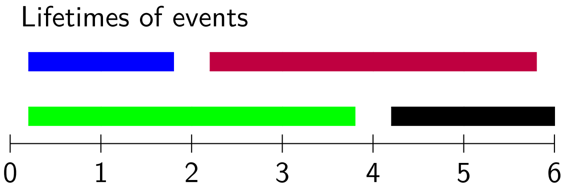
Lifetimes of events

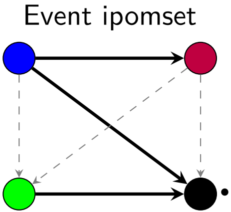


$$\begin{matrix} a\bullet \\ b\bullet \end{matrix} * \begin{matrix} \bullet a \\ \bullet b\bullet \end{matrix} * \begin{matrix} c\bullet \\ \bullet b\bullet \end{matrix} * \begin{matrix} \bullet c\bullet \\ \bullet b \end{matrix} * \begin{matrix} \bullet c\bullet \\ a\bullet \end{matrix} * \begin{matrix} \bullet c \\ \bullet a\bullet \end{matrix}$$

Event ipomset of a path







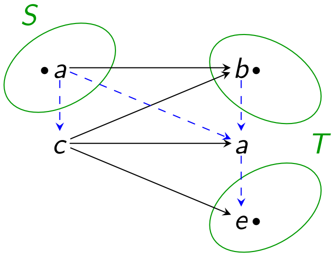
(not series-parallel!)

Pomsets with interfaces

Definition

A pomset with interfaces (ipomset): $(P, <, \dashrightarrow, S, T, \lambda)$:

- finite set P;
- two partial orders $<$ (precedence order), $\dashrightarrow$ (event order)
  - s.t. $< \cup \dashrightarrow$ is a *total relation*;
- $S, T \subseteq P$ source and target interfaces
  - s.t. S is $<$-minimal and T is $<$-maximal.

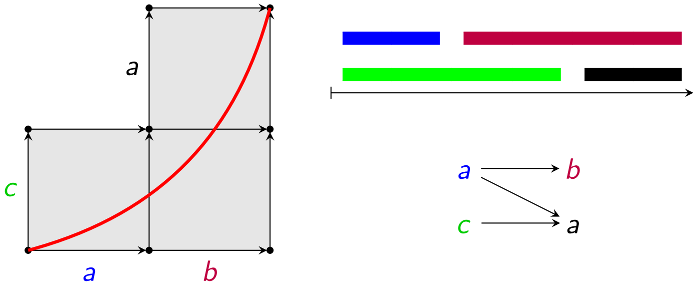


Interval orders

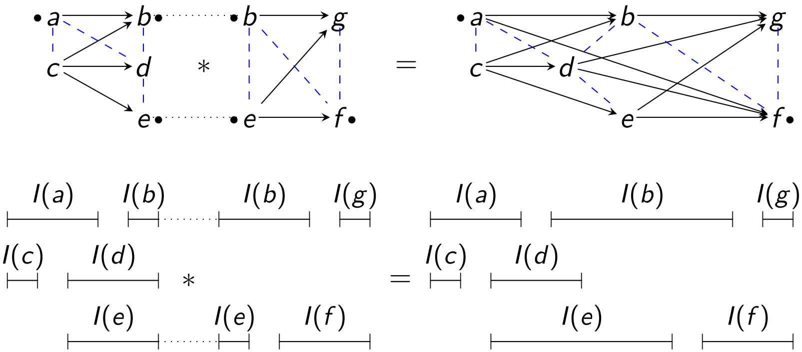
Definition

An ipomset $(P, <_P, \dashrightarrow, S, T, \lambda)$ is interval if $(P, <_P)$ has an interval representation: functions $b, e : P \to \mathbb{R}$ s.t.

- $\forall x \in P : b(x) \leq_{\mathbb{R}} e(x)$;
- $\forall x, y \in P : e(x) <_{\mathbb{R}} b(y) \iff x <_P y$

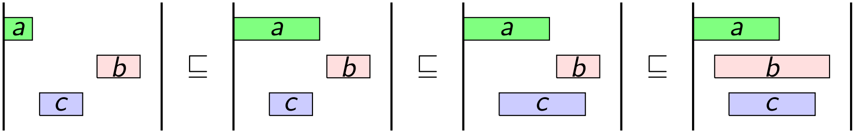


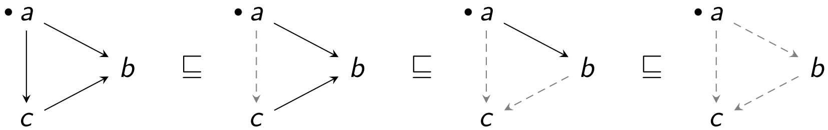
Gluing composition



- Gluing $P * Q$: P before Q, except for interfaces (which are identified)
- (also have parallel composition $P \parallel Q$: disjoint union)

Subsumption





P refines Q / Q subsumes P / $P \sqsubseteq Q$ iff

- P and Q have same interfaces
- P has more $<$ than Q
- Q has more $\dashrightarrow$ than P

Languages of HDAs

Definition

The language of an HDA X is the set of event ipomsets of all accepting paths:

$$L(X) = \{\mathrm{ev}(\pi) \mid \pi \in \mathrm{Paths}(X), \mathrm{src}(\pi) \in \bot_X, \mathrm{tgt}(\pi) \in \top_X\}$$

- $L(X)$ contains only interval ipomsets,
- is closed under subsumption,
- and has finite width

Definition

A language $L \subseteq \mathrm{iiPoms}$ is regular if there is an HDA X with $L = L(X)$.

Theorems

Definition (Rational Languages over Σ)

- Generated by $\emptyset$, $\{\epsilon\}$, and all $\{[a]\}$, $\{[\bullet\, a]\}$, $\{[a\,\bullet]\}$, $\{[\bullet\, a\,\bullet]\}$ for $a \in \Sigma$
- under operations $\cup$, $*$, $\|$ and (Kleene plus) $^+$
- (these need to take subsumption closure into account)

Definition (Monadic Second-Order Logics over Ipomsets)

$$\psi ::= a(x) \mid \mathsf{s}(x) \mid \mathsf{t}(x) \mid x < y \mid x \dashrightarrow y \mid x \in X \mid$$
$$\exists x.\, \psi \mid \forall x.\, \psi \mid \exists X.\, \psi \mid \forall X.\, \psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \neg\psi$$

Theorem (à la Kleene [LMCS 2024])

A language is rational iff it is regular.

Theorem (à la Büchi-Elgot-Trakhtenbrot [DLT 2024])

A language is rational iff it is MSO-definable, of finite width, and subsumption-closed.

More theorems

Theorem (à la Myhill-Nerode [FI 2024])

A language is rational iff it has finite prefix quotient.

Proof: $\Longrightarrow$ as usual

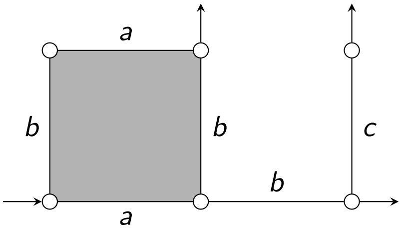
$\Longleftarrow$: usual MN congruence, but taking terminating interfaces into account

Theorem (Closure properties [TCS 2025])

Rational languages are closed under intersection
– but not under complement

- because $L = L{\downarrow}$ implies $\overline{L} = \overline{L}{\uparrow}$; also $\overline{L}$ has infinite width
- bounded-width pseudo-complement $\overline{L}^k = \{P \in \overline{L} \mid \text{wid}(P) \leq k\}$ is useful

Determinizability & ambiguity



Proposition

Not all HDAs are determinizable.

Proposition

There is a rational language which is inherently infinitely ambiguous.

$$\left(\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right] cd + ab \left[\begin{smallmatrix}c\\d\end{smallmatrix}\right]\right)^+$$

Kleene theorem: Easy parts

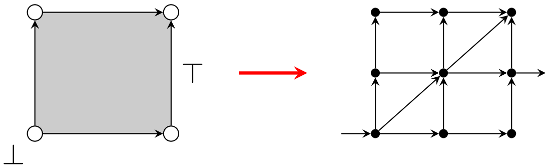
Theorem

A language is regular iff it is rational:

- *generated by* $\emptyset$, $\{\epsilon\}$, *and all* $\{[a]\}$, $\{[\bullet\, a]\}$, $\{[a\,\bullet]\}$, $\{[\bullet\, a\,\bullet]\}$ *for* $a \in \Sigma$
- *under operations* $\cup$, $*$, $\|$ *and (Kleene plus)* $^+$
- *(taking subsumption closure into account)*

Proof:

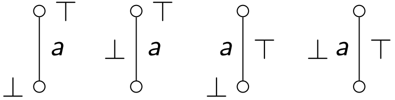
- regular $\implies$ rational: by reduction to ST-automata



Kleene theorem: Easy parts

- rational $\Longrightarrow$ regular: generators:

$L(X)$	$\emptyset$	$\{\epsilon\}$	$\{[a]\}$	$\{[\bullet\, a]\}$	$\{[a\, \bullet]\}$	$\{[\bullet\, a\, \bullet]\}$
X	$\emptyset$	$\bot \circ \top$				



- rational $\Longrightarrow$ regular: $\cup$ and $\parallel$

$$L(X) \cup L(Y) = L(X \sqcup Y) \qquad\qquad L(X) \parallel L(Y) = L(X \otimes Y)$$

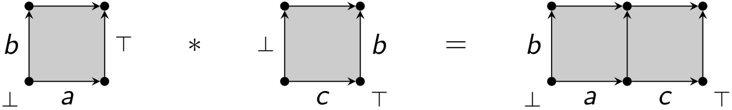
Kleene theorem: Difficult parts

- miss to see: gluings and iterations of regular languages are regular:

$$L(X) * L(Y) = L(X * Y) \qquad L(X)^+ = L(X^+)$$

- much more difficult: higher-dimensional gluings identify too much

- for example:



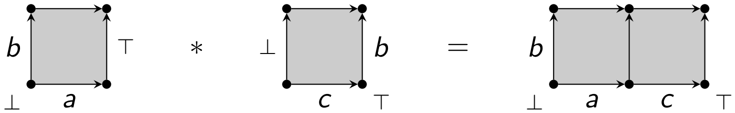
$L_2 = \emptyset$ $\qquad$ $L_1 * L_2 \ni ac$

Kleene theorem: Difficult parts

- miss to see: gluings and iterations of regular languages are regular:

$$L(X) * L(Y) = L(X * Y) \qquad L(X)^+ = L(X^+)$$

- much more difficult: higher-dimensional gluings identify too much
- for example:



$L_2 = \emptyset$ $\qquad$ $L_1 * L_2 \ni ac$

- use HDAs with interfaces and cylinder objects

HDAs with interfaces

A conclist with interfaces (iconclist) is a conclist U with subsets $S \subseteq U \supseteq T$, denoted ${}_S U_T$

(events in T cannot be terminated; events in S cannot be "unstarted")

A precubical set with interfaces (ipc-set) X consists of a set of cells X such that:

- Every cell $x \in X$ has an iconclist $\mathrm{ev}(x)$
- We write $X[{}_S U_T] = \{x \in X \mid \mathrm{ev}(x) = {}_S U_T\}$.
- For every $A \subseteq U - S$ there is a lower face map $\delta_A^0 : X[U] \to X[{}_S U_T - A]$.
- For every $B \subseteq U - T$ there is an upper face map $\delta_B^1 : X[U] \to X[{}_S U_T - b]$.
- Precubical identities: $\delta_A^\mu \delta_B^\nu = \delta_B^\nu \delta_A^\mu$ for $A \cap B = \emptyset$ and $\mu, \nu \in \{0, 1\}$
- (presheaves over a category $\mathrm{I}\square$)

An HDA with interfaces (iHDA) is a finite ipc-set with start and accept cells.

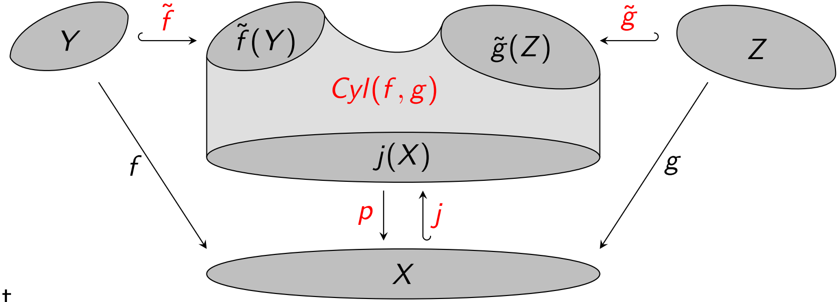
Extra conditions:

If $x \in X[{}_S U_T]$ is a start cell, then $S = U$.

If $x \in X[{}_S U_T]$ is an accept cell, then $T = U$.

Cylinders

Let X, Y, Z be ipc-sets and $f : Y \to X$, $g : Z \to X$ ipc-maps with $f(Y) \cap g(Z) = \emptyset$
There is a diagram of ipc-sets



such that

- $\tilde{f}$ is an initial inclusion;
- $\tilde{g}$ is a final inclusion;
- all paths in X from $f(Y)$ to $g(Z)$ lift to paths in $Cyl(f, g)$.

Cylinders: construction

X, Y, Z: ipc-sets, $f : Y \to X$, $g : Z \to X$: ipc-maps with $f(Y) \cap g(Z) = \emptyset$.
For ${}_S U_T \in \mathrm{I}\square$ let

$$Cyl(f, g)[{}_S U_T] = \{(x, K, L, \varphi, \psi)\}$$

such that

- $x \in X[{}_S U_T]$;
- $K \subseteq \mathrm{I}\square^U$ is an initial subset;
- $L \subseteq \mathrm{I}\square^U$ is a final subset;
- $\varphi : K \to Y$, $\psi : L \to Z$ are ipc-maps satisfying $f \circ \varphi = \iota_x|_K$ and $g \circ \psi = \iota_x|_L$:

$$\begin{tikzcd} K \arrow[r, hook] \arrow[d, "\varphi"'] & \mathrm{I}\square^U \arrow[d, "\iota_x"] & L \arrow[l, hook'] \arrow[d, "\psi"] \\ Y \arrow[r, "f"] & X & Z \arrow[l, "g"'] \end{tikzcd}$$

Gluing composition of regular languages is regular

Proposition

Gluing composition of regular languages is regular.

Proof sketch: Let L and M be regular languages.

1. We may ssume that L, M are simple, i.e., $L = L(X)$, $M = L(Y)$ for iHDAs X, Y having one initial and one accepting cell each.
2. Now replace X by $X' = Cyl(X \leftarrow \top_X : j)$ and Y by $Y' = Cyl(i : \bot_Y \to Y)$, then $L(X') = L(X)$ and $L(Y') = L(Y)$.
3. Go back to HDA and glue:

$$L(\mathbf{Cl}(X') * \mathbf{Cl}(Y')) = L(X') * L(Y') = L * M.$$

(closure $\mathbf{Cl}$: iHDA $\to$ HDA "adds missing cells")

4. So $L * M$ is recognized by a finite HDA, hence regular.

Conclusion

- Higher-Dimensional Automata: an automaton-like model for non-interleaving concurrency
- With a nice language theory!
- The trifecta Kleene–Myhill-Nerode–Büchi-Elgot-Trakhtenbrot is now complete for HDAs [LMCS 2024]–[FI 2024]–[DLT 2024]
- ST-automata provide a good combinatorial abstraction
- and geometry & topology provide intuition

Ongoing & Future work:

- Linear-time logic for HDAs — PhD Enzo Erlich
- Branching-time logics for HDAs — PhD Safa Zouari
- Higher-dimensional timed automata — [Petri Nets 2024]
- from Petri nets to HDAs — [Petri Nets 2025]
- HDAs over infinite ipomsets — PhD Jakub Gajdas