Languages of Higher-Dimensional Automata

Uli Fahrenberg

EPITA Rennes, France

RAMiCS 2021

Friends

- Christian Johansen, Gjøvik, Norway
- Georg Struth, Sheffield, UK
- Krzysztof Ziemiański, Warsaw, Poland

- Cameron Calk, Paris, France
- James Cranch, Sheffield, UK
- Eric Goubault, Paris, France

(i)Po(m)set Project

https://ulifahrenberg.github.io/poms

Search

The (i)Po(m)set Project

- What
- Who
- When
- How
- What
- Contact

What

The **(i)Po(m)set Project** is a research project at the crossroads of concurrency theory, algebra, and geometry. It aims to understand the basics of concurrency theory and develop its foundations.

Who

Members:

- Uli Fahrenberg, EPITA Rennes, France
- Christian Johansen, Norwegian University of Science and Technology, Gjøvik, Norway
- Georg Struth, University of Sheffield, UK
- Krzysztof Ziemiański, University of Warsaw, Poland

Associates:

- Cameron Calk, École polytechnique, Paris, France
- James Cranch, University of Sheffield, UK
- Eric Goubault, École polytechnique, Paris, France

Former members or associates:

How

Members and possibly associates meet about once a week on zoom to discuss research and papers and otherwise banter over big and small things.

What

Research published by the (i)Po(m)set Project:

1. Uli Fahrenberg, Christian Johansen, Christopher Trotter, Krzysztof Ziemiański: Sculptures in Concurrency. Logical Methods in Computer Science 17(2) (2021)
2. Uli Fahrenberg, Christian Johansen, Georg Struth, Ratan Bahadur Thapa: Generating Posets Beyond N. RAMiCS 2020: 82-99
3. Uli Fahrenberg, Christian Johansen, Georg Struth, Krzysztof Ziemiański: Domain Semirings United. CoRR abs/2011.04704 (2020)
4. Uli Fahrenberg, Christian Johansen, Georg Struth, Krzysztof Ziemiański: Languages of Higher-Dimensional Automata. Mathematical Structures in Computer Science (2021)
5. Uli Fahrenberg, Christian Johansen, Georg Struth, Krzysztof Ziemiański: lr-Multisemigroups and Modal Convolution Algebras. CoRR abs/2105.00188 (2021)
6. Cameron Calk, Uli Fahrenberg, Christian Johansen, Georg Struth, Krzysztof Ziemiański: lr-Multisemigroups, Modal Quantales and the Origin of Locality. RAMiCS 2021
7. Uli Fahrenberg, Christian Johansen, Georg Struth, Krzysztof Ziemiański: Posets with Interfaces for Concurrent Kleene Algebra. CoRR abs/2106.10895 (2021)

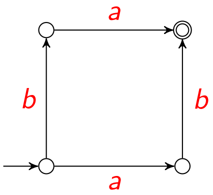
Software and data published by the (i)Po(m)set Project:

- Python code related to [2]
- Julia code related to [6]
- Iposets and gp-iposets on up to 8 points, and forbidden substructures on up to 10 points


Uli Fahrenberg
Krzysztof Ziemiański
Georg Struth
Christian Johansen

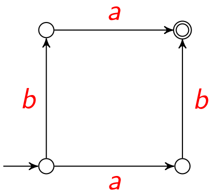
Higher-dimensional automata

$a \| b$

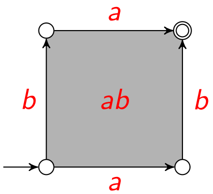


Higher-dimensional automata

$a|b$



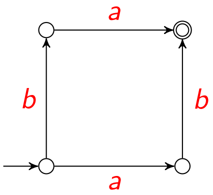
$a.b + b.a$

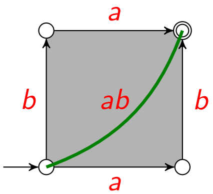


a and b are independent

Higher-dimensional automata

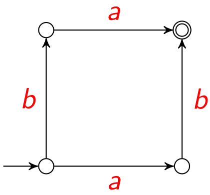
$a \| b$



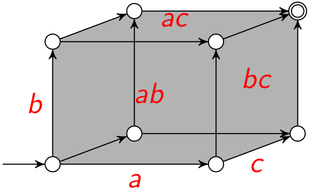


Higher-dimensional automata

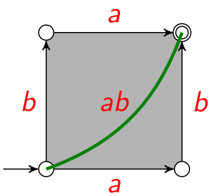
$a|b$

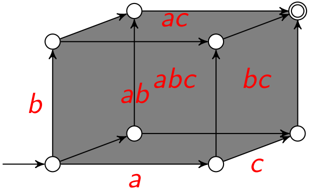


$a|b|c$



pairwise independent





$\{a, b, c\}$ independent

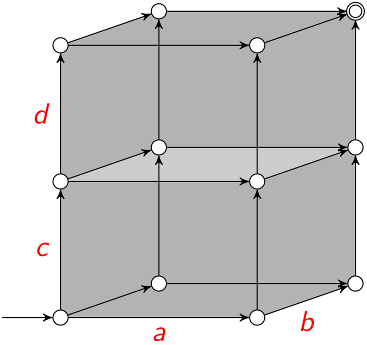
Higher-dimensional automata & concurrency

HDA 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]

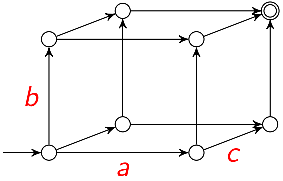
[van Glabbeek 2006, TCS]: Up to history-preserving bisimilarity, HDA "generalize the main models of concurrency proposed in the literature"

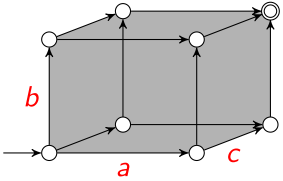
Another example

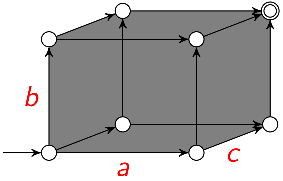


- no cubes, all faces except middle horizontal
- *a* and *b* independent; *c* introduces conflict; *d* releases conflict

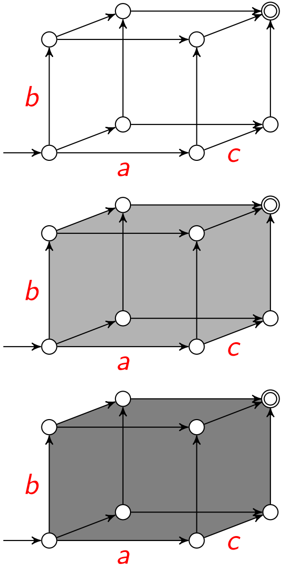
Languages of HDA





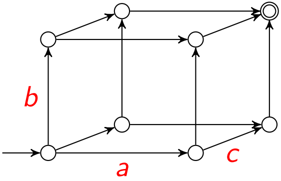


Languages of HDA

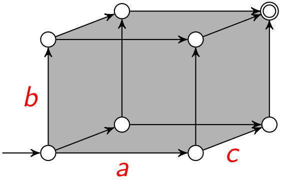


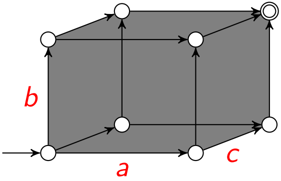
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$

Languages of HDA



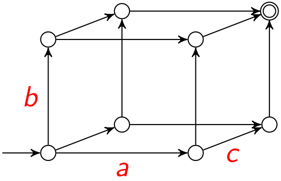
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$



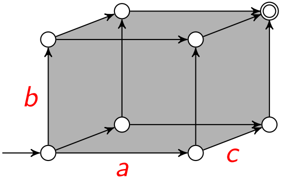


$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \dots \right\}$$

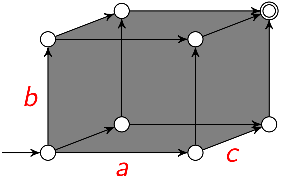
Languages of HDA



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

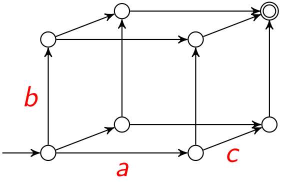


$$L_2 = \left\{ \begin{pmatrix} a \\ b \to c \end{pmatrix}, \begin{pmatrix} a \\ c \to b \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ c \to a \end{pmatrix}, \begin{pmatrix} c \\ a \to b \end{pmatrix}, \begin{pmatrix} c \\ b \to a \end{pmatrix}, \dots \right\}$$

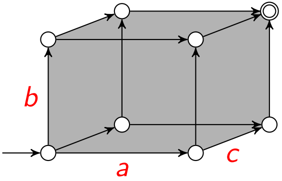


$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \dots \right\}$$

Languages of HDA

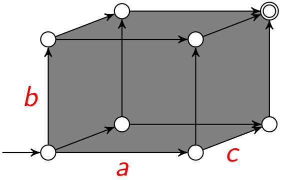


$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$



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

sets of pomsets



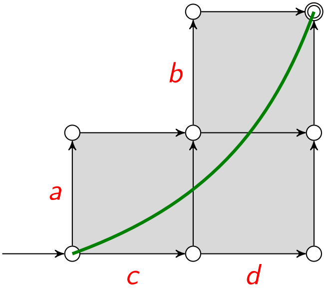
$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right\} \cup L_2$$

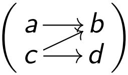
Pomsets

A (finite) pomset ("partially ordered multiset") $(P, \leq, \ell)$:

- a finite partially ordered set $(P, \leq)$
- with labeling $\ell : P \to \Sigma$
- (AKA labeled partial order)
- (up to isomorphism: don't care about identity of points)
- [Winkowski '77], [Lamport '78], etc.

Another example





- not series-parallel!

Are all pomsets generated by HDA?

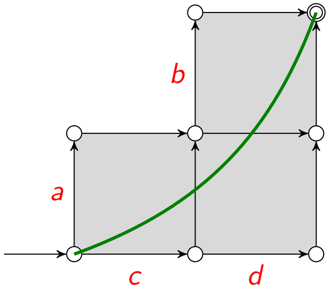
No, only (labeled) interval orders

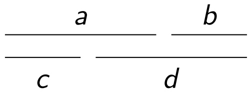
- Poset $(P, \leq)$ is an interval order iff it does not contain ($\rightrightarrows$)
  - (iff it is "2+2-free")
- iff it has an interval representation:
  - a set $I = \{[l_i, r_i]\}$ of real intervals
  - with order $[l_i, r_i] \preceq [l_j, r_j]$ iff $r_i \leq l_j$
  - and an order isomorphism $(P, \leq) \leftrightarrow (I, \preceq)$
- [Fishburn 1970]

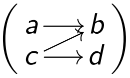
Are all pomsets generated by HDA?

No, only (labeled) interval orders

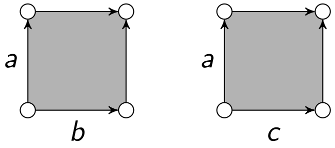
- Poset $(P, \leq)$ is an interval order iff it does not contain $(\rightrightarrows)$
  - (iff it is "2+2-free")
- iff it has an interval representation:
  - a set $I = \{[l_i, r_i]\}$ of real intervals
  - with order $[l_i, r_i] \preceq [l_j, r_j]$ iff $r_i \leq l_j$
  - and an order isomorphism $(P, \leq) \leftrightarrow (I, \preceq)$







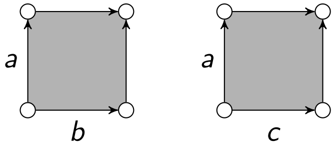
Concatenation of HDA



$\begin{pmatrix} a \\ b \end{pmatrix}$ $\begin{pmatrix} a \\ c \end{pmatrix}$

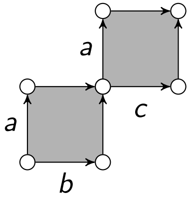
Two possible compositions:

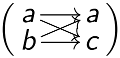
Concatenation of HDA

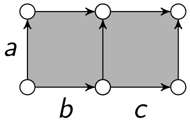


$\begin{pmatrix} a \\ b \end{pmatrix}$ $\begin{pmatrix} a \\ c \end{pmatrix}$

Two possible compositions:

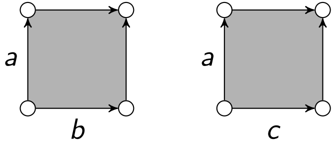






$\begin{pmatrix} a & \\ b \longrightarrow & c \end{pmatrix}$

Concatenation of HDA

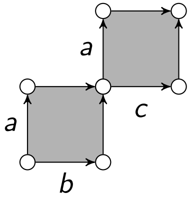


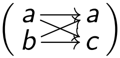
$\begin{pmatrix} a \\ b \end{pmatrix}$ $\begin{pmatrix} a \\ c \end{pmatrix}$

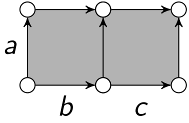
Two possible compositions:

- not clear whether the two *a* are the same event
- idea: let the objects specify how they may be composed

⇒ pomsets with interfaces







$\begin{pmatrix} a & \\ b \longrightarrow & c \end{pmatrix}$

Posets with interfaces

Definition

A poset with interfaces (iposet) is a poset P plus two injections

$$[n] \xrightarrow{s} P \xleftarrow{t} [m]$$

such that $s[n]$ is minimal and $t[m]$ is maximal in P.

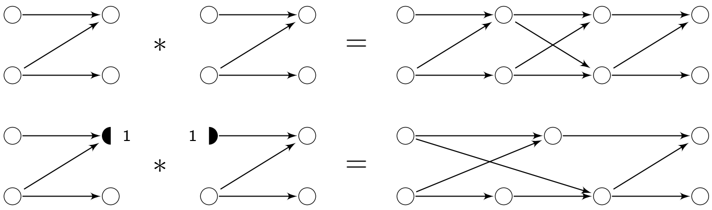
- $([n] = \{1, \dots, n\})$
- s: starting interface ; t: terminating interface
- events in $t[m]$ are *unfinished* ; events in $s[n]$ are "*unstarted*"

Gluing composition

Definition

The gluing composition of iposets $s_1 : [n] \to (P_1, \leq_1) \leftarrow [m] : t_1$ and $s_2 : [m] \to (P_2, \leq_2) \leftarrow [k] : t_2$:

$$P_1 * P_2 = \begin{cases} (P_1 \sqcup P_2)/t_1(i) = s_2(i) \\ \leq_1 \cup \leq_2 \cup (P_1 \setminus t_1[m]) \times (P_2 \setminus s_2[m]) \end{cases}$$



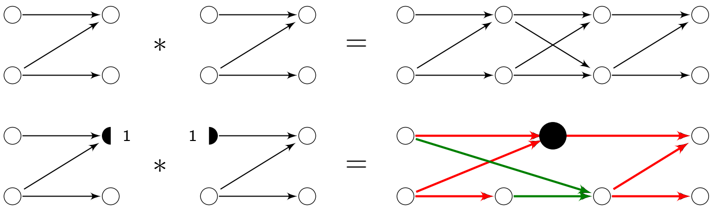
- only defined if terminating int. of P_1 is equal to starting int. of P_2

Gluing composition

Definition

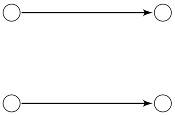
The gluing composition of iposets $s_1 : [n] \to (P_1, \le_1) \leftarrow [m] : t_1$ and $s_2 : [m] \to (P_2, \le_2) \leftarrow [k] : t_2$:

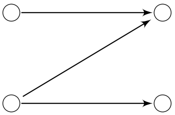
$$P_1 * P_2 = \begin{cases} (P_1 \sqcup P_2)/t_1(i) = s_2(i) \\ \le_1 \cup \le_2 \cup (P_1 \setminus t_1[m]) \times (P_2 \setminus s_2[m]) \end{cases}$$



- only defined if terminating int. of P_1 is equal to starting int. of P_2

Interval Orders vs Series-Parallel Posets





- interval orders are used in Petri net theory and distributed computing
- but have no algebraic representation (so far)
- sp-posets are used in concurrency theory & have nice algebraic theory
- Concurrent Kleene algebra
- int. orders are 2+2-free; sp-posets are **N**-free
- incomparable: 2+2 is sp; **N** is interval

⇒ [FJST RAMiCS'20]

Languages of HDA

- For an HDA A, $L(A)$ is
  - a set of (labeled) interval orders
  - closed under subsumption
- For any interval order P, $\exists$ HDA $\square^P$ for which $L(\square^P) = \{P\}{\downarrow}$
- and then for any HDA A, $P \in L(A)$ iff $\exists f : \square^P \to A$
  - very useful criterion

- Any finite language is recognized by an HDA
- $L(A \cup B) = L(A) \cup L(B)$
- $L(A \| B) = L(A) \| L(B)$ (more precisely, $(L(A) \| L(B)){\downarrow} \cap IO$)
- [FJSZ MSCS'21]

Conclusion

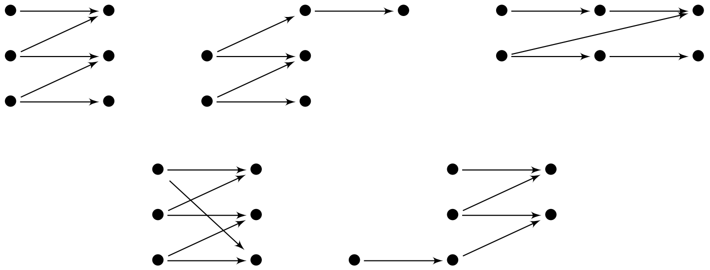
- Higher-dimensional automata: nice model for concurrency
- Languages of HDA: sets of labeled interval orders
- Po(m)sets with interfaces for compositionality / algebra

Open / coming up:

- Higher-dimensional regular languages
- 2-categories with lax tensors: algebraic setting for iposets
- Combinatorial characterization of gluing-parallel iposets
- ...

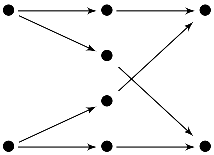
Forbidden substructures

- sp-posets $\triangleq$ **N**-free
- interval orders $\triangleq$ 2+2-free
- gluing-parallel posets $\Rightarrow$ free of



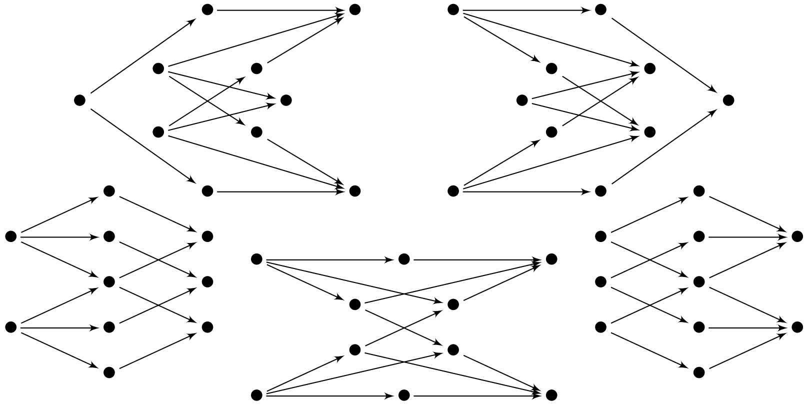
Forbidden substructures

- sp-posets $\hat{=}$ **N**-free
- interval orders $\hat{=}$ 2+2-free
- gluing-parallel posets $\Rightarrow$ free of



Forbidden substructures

- sp-posets ≙ **N**-free
- interval orders ≙ 2+2-free
- gluing-parallel posets ⇒ free of



Some numbers

n	P(n)	SP(n)	IO(n)	GP(n)	IP(n)	GPI(n)
0	1	1	1	1	1	1
1	1	1	1	1	4	4
2	2	2	2	2	17	16
3	5	5	5	5	86	74
4	16	15	15	16	532	419
5	63	48	53	63	4068	2980
6	318	167	217	313	38.933	26.566
7	2045	602	1014	1903	474.822	289.279
8	16.999	2256	5335	13.943	7.558.620	3.726.311
9	183.231	8660	31.240	120.442		
10	2.567.284	33.958	201.608	1.206.459		
11	46.749.427	135.292	1.422.074			
EIS	112	3430	22493	345673	331158	331159