Generating Posets beyond **N**

Uli Fahrenberg[1] Christian Johansen[2]
Georg Struth[3] Ratan Bahadur Thapa[2]

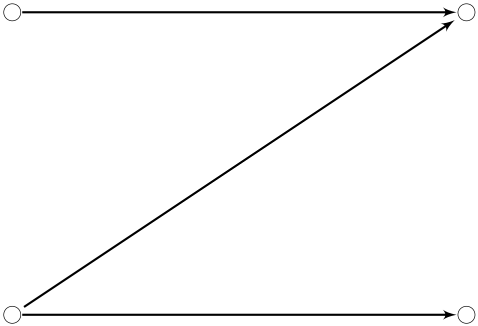
École Polytechnique, Palaiseau, France

University of Oslo, Norway

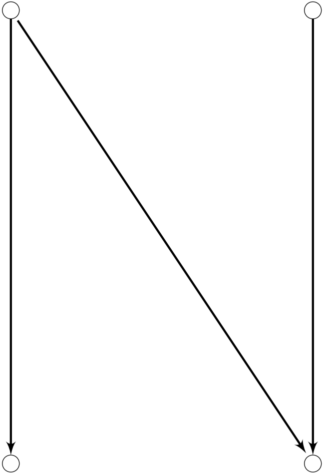
University of Sheffield, UK

Cosynus / P&A, 2021-02

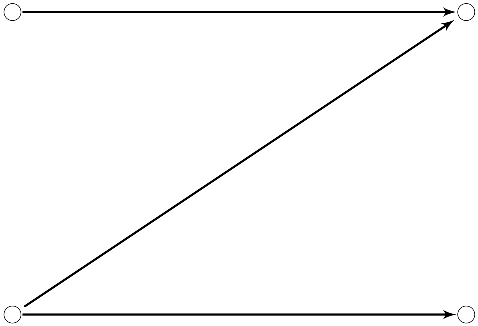
Motivation



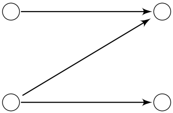
Motivation



Motivation

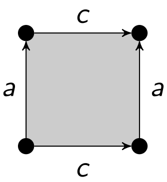


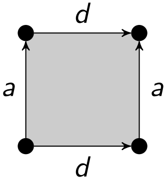
Motivation



- Kleene algebra is nice and useful
  - also its extensions: semimodules, tests, domain, . . .
- Concurrent Kleene algebra: extension of KA for concurrency
  - [Hoare, Möller, O'Hearn, Struth, van Staden, Villard, Wehrman, Zhu '09, '11, '16]
- Kleene algebra plus parallel composition
- the free CKA (minus some details): sets of series-parallel pomsets
  - labeled posets with concatenation & parallel composition
- *Something's amiss in concurrent Kleene algebra*

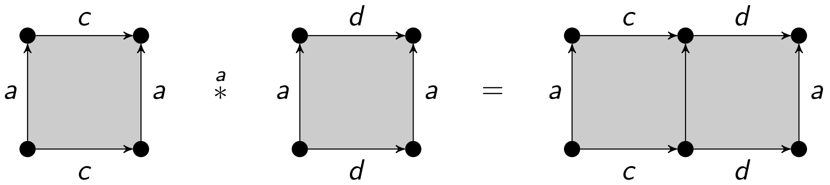
Example, Using Higher-Dimensional Automata





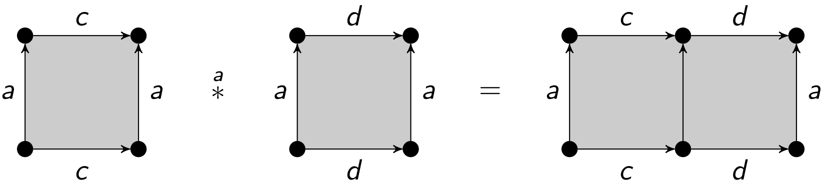
$\begin{pmatrix} a \\ c \end{pmatrix}$ $\qquad$ $\begin{pmatrix} a \\ d \end{pmatrix}$

Example, Using Higher-Dimensional Automata



$\begin{pmatrix} a \\ c \end{pmatrix}$ $\qquad$ $\begin{pmatrix} a \\ d \end{pmatrix}$ $\qquad$ $\begin{pmatrix} a \\ c \longrightarrow d \end{pmatrix}$

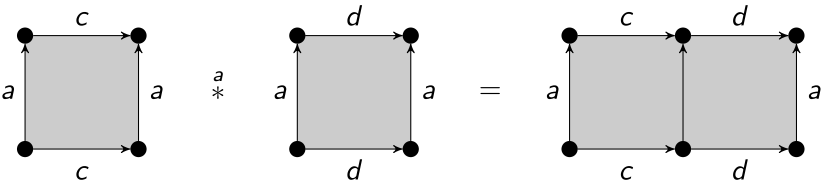
Example, Using Higher-Dimensional Automata



$\begin{pmatrix} a \\ c \end{pmatrix}$ $\qquad$ $\begin{pmatrix} a \\ d \end{pmatrix}$ $\qquad$ $\left(c \xrightarrow{\;a\;} d \right)$

$$\begin{pmatrix} a \\ c \end{pmatrix} \parallel \begin{pmatrix} a \\ d \end{pmatrix} = \begin{pmatrix} a \\ c \\ a \\ d \end{pmatrix}$$??

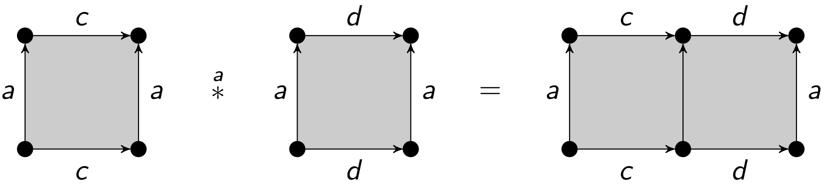
Example, Using Higher-Dimensional Automata



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

$$\begin{pmatrix} a \\ c \end{pmatrix} * \begin{pmatrix} a \\ d \end{pmatrix} = \left(\begin{tikzcd} a \arrow[r] \arrow[dr] & a \\ c \arrow[r] \arrow[ur] & d \end{tikzcd}\right)$$??

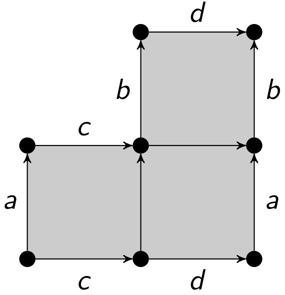
Example, Using Higher-Dimensional Automata



$$\begin{pmatrix} a \\ c \end{pmatrix} \quad \overset{a}{*} \quad \begin{pmatrix} a \\ d \end{pmatrix} \quad = \quad \left(c \xrightarrow{\;a\;} d \right)$$

- new gluing operation on pomsets, to *continue events across compositions*

Another Example



$$\begin{pmatrix} a \\ c \end{pmatrix} \overset{a}{*} \begin{pmatrix} a \\ d \end{pmatrix} \overset{d}{*} \begin{pmatrix} b \\ d \end{pmatrix} = \left(\begin{tikzcd} a \arrow[r] & b \\ c \arrow[r] \arrow[ur] & d \end{tikzcd}\right)$$

- this is the **N** pomset, which is not series-parallel
- hence our title, Generating Posets beyond **N**

Series-Parallel Posets

- a poset: *finite* set P plus partial order $\leq$: reflexive, transitive, antisymmetric
- parallel composition of posets $(P_1, \leq_1)$, $(P_2, \leq_2)$:

$$P_1 \otimes P_2 = (P_1 \sqcup P_2, \leq_1 \cup \leq_2)$$

$\upuparrows$ disjoint union

- serial composition:

$$P_1 * P_2 = (P_1 \sqcup P_2, \leq_1 \cup \leq_2 \cup P_1 \times P_2)$$

$\upuparrows$ P_1 before P_2

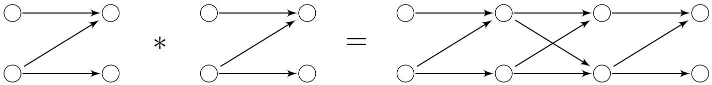
Series-Parallel Posets

- a poset: *finite* set P plus partial order $\leq$: reflexive, transitive, antisymmetric
- parallel composition of posets $(P_1, \leq_1)$, $(P_2, \leq_2)$:

$$P_1 \otimes P_2 = (P_1 \sqcup P_2, \leq_1 \cup \leq_2)$$

- serial composition:

$$P_1 * P_2 = (P_1 \sqcup P_2, \leq_1 \cup \leq_2 \cup P_1 \times P_2)$$



Series-Parallel Posets

Definition (Winkowski '77, Grabowski '81)

A poset is series-parallel (sp) if it is empty or can be obtained from the singleton poset by a finite number of serial and parallel compositions.

Theorem (Grabowski '81)

A poset is sp iff it does not contain $\boldsymbol{N}$ *as an induced subposet.*

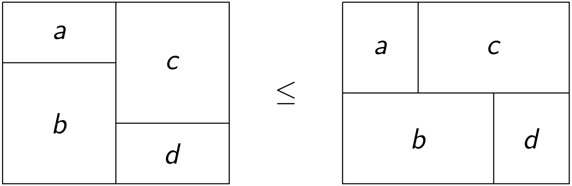
The equational theory of sp-posets is well-understood: [Gischer '88], [Bloom-Esik '96]

Concurrent Monoids

Definition (Gischer '88, Hoare *et al.* '11)

A concurrent monoid is an ordered bimonoid $(S, \leq, *, \|, 1)$ with shared $*$-$\|$-unit 1 which satisfies weak interchange:

$$(a \| b) * (c \| d) \leq (a * c) \| (b * d)$$



- subsumption on posets: $P \preceq Q$ if P "has more order" than Q

Theorem (Gischer '88, Bloom-Esik '96)

The set of sp-posets under subsumption is the free concurrent monoid.

Background: Concurrent Kleene Algebra

- concurrent monoids: basic algebraic structure for concurrent Kleene algebra
- Kleene algebra is useful in language theory (compilers!), verification, etc. etc.
- concurrent Kleene algebra: quest to extend that success to parallel programming
- distributed systems; weak memory; etc. etc.
- Tony Hoare, Bernhard Möller, Peter O'Hearn, Georg Struth, Huibiao Zhu 2009++
- process algebra with $+$ (non-determinism), $\cdot$ (concatenation), $\|$ (parallelism), * (iteration), and † (parallel iteration)
- (in this talk, no iterations!)
- problem: no **N**s!

Problem & Solution

- we like the **N** poset, but it's not series-parallel
- in fact, **N**'s are everywhere: for example, *producer-consumer:*

$$\begin{tikzcd} p_1 \arrow[r] \arrow[d] & p_2 \arrow[r] \arrow[d] & p_3 \arrow[r] \arrow[d] & p_4 \arrow[r] \arrow[d] & \cdots \\ c_1 \arrow[r] & c_2 \arrow[r] & c_3 \arrow[r] & c_4 \arrow[r] & \cdots \end{tikzcd}$$

Problem & Solution

- we like the **N** poset, but it's not series-parallel
- in fact, **N**'s are everywhere: for example, *producer-consumer:*

$$\begin{tikzcd} p_1 \arrow[r] \arrow[d] & p_2 \arrow[r, red] \arrow[d, red] & p_3 \arrow[r] \arrow[d] & p_4 \arrow[r, red] \arrow[d, red] & \cdots \\ c_1 \arrow[r, red] & c_2 \arrow[r] & c_3 \arrow[r, red] & c_4 \arrow[r] & \cdots \end{tikzcd}$$

Problem & Solution

- we like the **N** poset, but it's not series-parallel
- in fact, **N**'s are everywhere: for example, *producer-consumer:*

$$\begin{tikzcd}
p_1 \arrow[r] \arrow[d] & p_2 \arrow[r] \arrow[d] & p_3 \arrow[r] \arrow[d] & p_4 \arrow[r] \arrow[d] & \cdots \\
c_1 \arrow[r] & c_2 \arrow[r] & c_3 \arrow[r] & c_4 \arrow[r] & \cdots
\end{tikzcd}$$

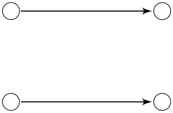
Problem

Find a class of posets which includes ***N*** *(and sp-posets) and which has good algebraic properties.*

Our Proposal

Posets with interfaces with parallel and gluing composition.

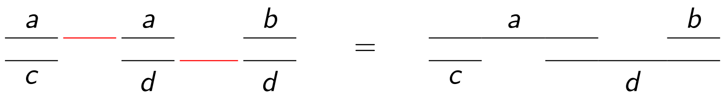
Interlude: Interval Orders



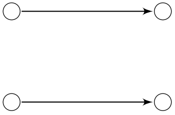
- posets which are good for concurrency?
- already in [Wiener 1914], then [Winkowski '77], [Lamport '86], [van Glabbeek '90], [Vogler '91], [Janicky '93], etc.
- interval orders: posets which have representation as (real) intervals, ordered by $\max_1 \le \min_2$
- Lemma (Fishburn '70): A poset is interval iff it does not contain $\mathbf{II} = \left(\begin{smallmatrix} \cdot \longrightarrow \cdot \\ \cdot \longrightarrow \cdot \end{smallmatrix} \right)$ as induced subposet.
- intuitively: if $a \longrightarrow b$ and $c \longrightarrow d$, then also $a \longrightarrow d$ or $c \longrightarrow b$

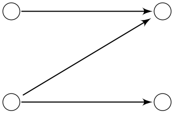
Gluing of Interval Orders

$$\begin{pmatrix} a \\ c \end{pmatrix} \overset{a}{*} \begin{pmatrix} a \\ d \end{pmatrix} \overset{d}{*} \begin{pmatrix} b \\ d \end{pmatrix} = \left(\begin{tikzcd}[ampersand replacement=\&] a \arrow[r] \& b \\ c \arrow[r] \arrow[ur] \& d \end{tikzcd}\right)$$



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
- but applicativity is doubtful!
- int. orders are **II**-free; sp-posets are **N**-free
- incomparable: **II** is sp; **N** is interval
- goal: marriage

Posets with Interfaces

Definition

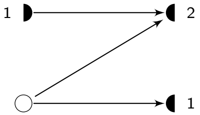
A poset with interfaces (iposet) is a poset P plus two injections

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

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

- $([n] = \{1, \dots, n\}$; $S \subseteq P$ *minimal* if $p \not< s$ for all $p \in P$, $s \in S)$
- (there are 25 non-isomorphic iposets with underlying **N**)

Interfaces



Def.: Iposet $s : [n] \to P \leftarrow [m] : t$; $s[n] \subseteq P_{\min}$, $t[m] \subseteq P_{\max}$.

- s: starting interface ; t: terminating interface
- events in $t[m]$ are *unfinished* ; events in $s[n]$ are "*unstarted*"

Definition

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

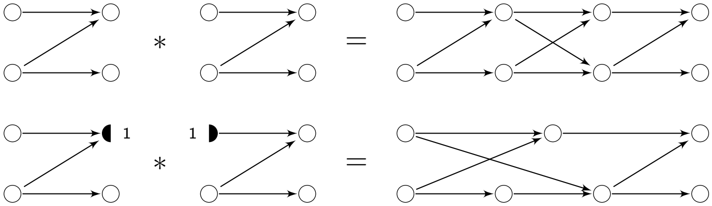
$$P_1 * P_2 = \begin{cases} (P_1 \sqcup P_2)/t_1(i) = s_2(i) \\ \leq_1 \cup \leq_2 \cup (P_1 \setminus t_1[m]) \times (P_2 \setminus s_2[m]) \end{cases}$$

Gluing Composition

Definition

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

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

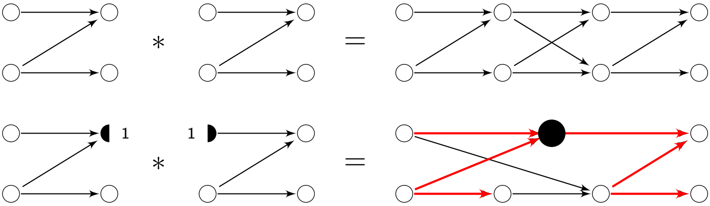


Gluing Composition

Definition

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

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

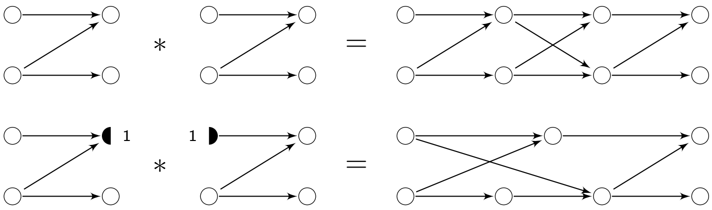


Gluing Composition

Definition

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

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



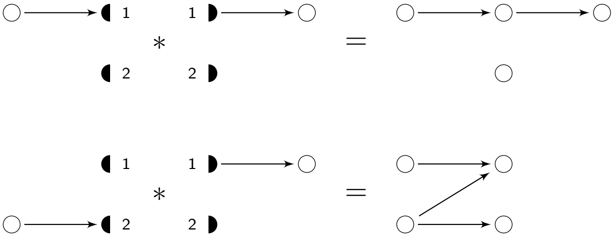
- only defined if terminating int. of P_1 is equal to starting int. of P_2
- iposets form category (with gluing as composition)

Parallel Composition

- parallel composition of iposets: put posets in parallel and renumber interfaces
- for $[n_1] \to P_1 \leftarrow [m_1]$ and $[n_2] \to P_2 \leftarrow [m_2]$, have $[n_1 + n_2] \to P_1 \otimes P_2 \leftarrow [m_1 + m_2]$
- not commutative ; only "lax tensor" ; not a PROP

Parallel Composition

- parallel composition of iposets: put posets in parallel and renumber interfaces
- for $[n_1] \to P_1 \leftarrow [m_1]$ and $[n_2] \to P_2 \leftarrow [m_2]$, have $[n_1 + n_2] \to P_1 \otimes P_2 \leftarrow [m_1 + m_2]$
- not commutative ; only "lax tensor" ; not a PROP

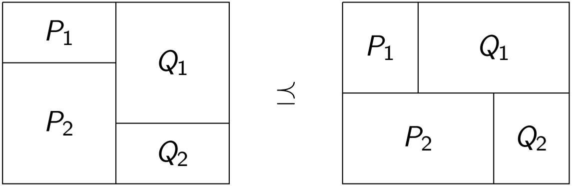


Parallel Composition

- parallel composition of iposets: put posets in parallel and renumber interfaces
- for $[n_1] \to P_1 \leftarrow [m_1]$ and $[n_2] \to P_2 \leftarrow [m_2]$, have $[n_1 + n_2] \to P_1 \otimes P_2 \leftarrow [m_1 + m_2]$
- not commutative ; only "lax tensor" ; not a PROP

$\Uparrow$

$$(P_1 \otimes P_2) * (Q_1 \otimes Q_2) \preceq (P_1 * Q_1) \otimes (P_2 * Q_2)$$



Gluing-Parallel Iposets

- recall *sp-posets*: freely generated from $\bigcirc$ using $*$ and $\otimes$
- the four singleton iposets:

$$\bigcirc \qquad {}_1\text{◗} \qquad \text{◖}_1 \qquad {}_1\text{⧓}_1$$

- gp-iposets: generated from $\bigcirc$, ${}_1\text{◗}$, ◖_1, ${}_1\text{⧓}_1$ using $*$ and $\otimes$

Proposition

Gp-iposets are freely generated, except for the relations

$$\begin{pmatrix} \text{◖}_1 \\ P \end{pmatrix} * \begin{pmatrix} {}_1\text{◗} \\ Q \end{pmatrix} = \begin{pmatrix} \bigcirc \\ P * Q \end{pmatrix} \qquad \begin{pmatrix} \text{◖}_1 \\ P \end{pmatrix} * \begin{pmatrix} {}_1\text{⧓}_1 \\ Q \end{pmatrix} = \begin{pmatrix} \text{◖}_1 \\ P * Q \end{pmatrix}$$

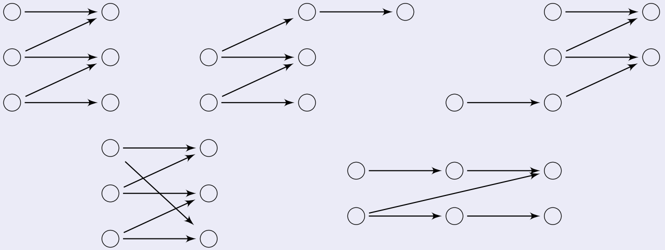
$$\begin{pmatrix} {}_1\text{⧓}_1 \\ P \end{pmatrix} * \begin{pmatrix} {}_1\text{◗} \\ Q \end{pmatrix} = \begin{pmatrix} {}_1\text{◗} \\ P * Q \end{pmatrix} \qquad \begin{pmatrix} {}_1\text{⧓}_1 \\ P \end{pmatrix} * \begin{pmatrix} {}_1\text{⧓}_1 \\ Q \end{pmatrix} = \begin{pmatrix} {}_1\text{⧓}_1 \\ P * Q \end{pmatrix}$$

Forbidden Substructures

- recall: P is sp iff P is **N**-free

Proposition

If P is gp, then it does not contain any of the following as induced subposets:



- unlike for *sp*-posets, that's not an iff (we don't know)
- but these five are the only posets on ≤ 6 points which are not gp

Some Counting, up to Isomorphism

n	P(n)	SP(n)	GP(n)	IP(n)	GPI(n)
0	1	1	1	1	1
1	1	1	1	4	4
2	2	2	2	17	16
3	5	5	5	86	74
4	16	15	16	532	419
5	63	48	63	???	2980
6	318	167	313	???	26566
7	2045	602	???	???	???
OEIS	A000112	A003430	n.a.	n.a.	n.a.

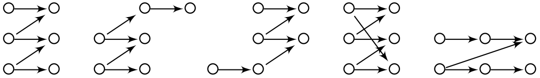
Some Counting, up to Isomorphism

n	P(n)	SP(n)	GP(n)	IP(n)	GPI(n)
0	1	1	1	1	1
1	1	1	1	4	4
2	2	2	2	17	16
3	5	5	5	86	74
4	16	15	16	532	419
5	63	48	63	???	2980
6	318	167	313	???	26566
7	2045	602	???	???	???
OEIS	A000112	A003430	n.a.	n.a.	n.a.

- slow Python implementation
- bottleneck is isomorphism checking
- new Julia implementation coming up!

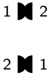
Some Counting, up to Isomorphism

n	P(n)	SP(n)	GP(n)	IP(n)	GPI(n)
0	1	1	1	1	1
1	1	1	1	4	4
2	2	2	2	17	16
3	5	5	5	86	74
4	16	15	16	532	419
5	63	48	63	???	2980
6	318	167	313	???	26566
7	2045	602	???	???	???
OEIS	A000112	A003430	n.a.	n.a.	n.a.

- P(6) – GP(6) = 5: 

Some Counting, up to Isomorphism

n	P(n)	SP(n)	GP(n)	IP(n)	GPI(n)
0	1	1	1	1	1
1	1	1	1	4	4
2	2	2	2	17	16
3	5	5	5	86	74
4	16	15	16	532	419
5	63	48	63	???	2980
6	318	167	313	???	26566
7	2045	602	???	???	???
OEIS	A000112	A003430	n.a.	n.a.	n.a.

- the only iposet on 2 points which is not gp: 

Generating Posets Up to Isomorphism

- with Olavi Äikäs, 3rd year BSc student in internship
- convert slow Python code to fast Julia code
- be clever about isomorphisms:

Generating Posets Up to Isomorphism

- with Olavi Äikäs, 3rd year BSc student in internship
- convert slow Python code to fast Julia code
- be clever about isomorphisms:
  - canonical labeling:
    $P \cong Q \iff f(P) = f(Q)$
  - isomorphism invariant:
    $P \cong Q \implies f(P) = f(Q)$
  - number of out-edges; number of in-edges
  - same numbers, but in Hasse diagram
  - filtration level
  - ...?

n	P(n)	SP(n)	GP(n)
6	318	167	313
7	2045	602	???
8	16999	2256	???

Generating Posets Up to Isomorphism

- with Olavi Äikäs, 3rd year BSc student in internship
- convert slow Python code to fast Julia code
- be clever about isomorphisms:
  - canonical labeling:
    $P \cong Q \iff f(P) = f(Q)$
  - isomorphism invariant:
    $P \cong Q \implies f(P) = f(Q)$
  - number of out-edges; number of in-edges
  - same numbers, but in Hasse diagram
  - filtration level
  - ...?

n	$\mathrm{P}(n)$	$\mathrm{SP}(n)$	$\mathrm{GP}(n)$
6	318	167	313
7	2045	602	???
8	16999	2256	???

- $\mathrm{P}(n)$: known up to $n = 16$: [Brinkmann-McKay, Order 2002]
- $\mathrm{SP}(n)$: generating formula; Ex. I.46 in [Flajolet-Sedgewick 2009]
- $\mathrm{GP}(n)$: ???

Generating Posets Up to Isomorphism

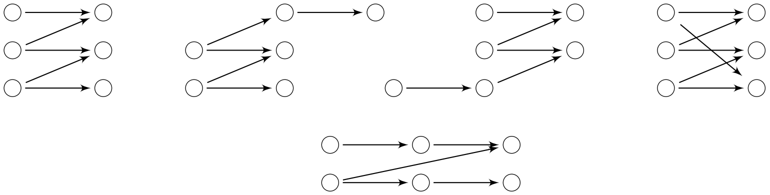
- with Olavi Äikäs, 3rd year BSc student in internship
- convert slow Python code to fast Julia code
- be clever about isomorphisms:
  - canonical labeling:
    $P \cong Q \iff f(P) = f(Q)$
  - isomorphism invariant:
    $P \cong Q \implies f(P) = f(Q)$
  - number of out-edges; number of in-edges
  - same numbers, but in Hasse diagram
  - filtration level
  - …?

n	P(n)	SP(n)	GP(n)
6	318	167	313
7	2045	602	1903
8	16999	2256	???

- P(n): known up to $n = 16$: [Brinkmann-McKay, Order 2002]
- SP(n): generating formula; Ex. I.46 in [Flajolet-Sedgewick 2009]
- GP(n): ???

New Results

- with Olavi Äikäs, 3rd year BSc student in internship
- recall Proposition: If P is gp, then it does not contain any of the following as induced subposets:



- New: there are 1903 gp-posets on 7 points
- hence 142 posets on 7 points which are not gp
- no new forbidden substructures! : for $|P| \leq 7$, P is gp iff it has no induced subposets as above.

Conclusion

- posets with interfaces for concurrency
- instead of concurrent monoid, small category with lax tensor
  - a “multi-object concurrent monoid”
- gluing-parallel iposets include sp-posets and interval orders
- generation is “almost free”
- characterization by forbidden substructures?

Ongoing and Future Work

- Concurrent Kleene algebra:
  - concurrent monoid $\rightsquigarrow$ concurrent Kleene algebra
  - concurrent category $\rightsquigarrow$ bicategory with lax tensors?
- CKA with domain:
  - domain elements are "structure-less" iposets
  - relation to higher-dimensional modal logic?
  - higher-dimensional modal Kleene algebra
- Languages of higher-dimensional automata:
  - sets of interval orders
  - concatenation of HDA $\approx$ gluing of (sets of) interval orders
  - theory of regular languages for concurrency?
- Real-time concurrency:
  - higher-dimensional timed automata
  - languages are sets of real-time interval orders?
  - relation to real-time Petri nets?