On the combinatorics of Series-Parallel programs

Another title wich is not the good one

CLA Workshop 2016

Olivier Bodini, **Matthieu Dien**,
Antoine Genitrini and Frédéric Peschanski

UPMC - LIP6 - APR Team

May 26th, 2016

Outline

Outline

Introduction

> *The approach* (about analytic combinatorics) *is predicated on the idea that combinatorial structures are typically defined by simple formal rules that are the key to learning their properties.*
>
> Flajolet & Sedgewick dans *Introduction to AofA 1st ed.* (1995)

Concurrency theory

Program

```
P = ν(B)ν(C)
    [fork.(start₁.<B>end₁.<C>0
          || start₂.<B>end₂.0
          || <B>start₃.<C>end₃.0)]
```

Program runs

-> fork		-> fork
-> start2		-> start1
-> start1		-> start2
-> end2	ou	-> end1
-> start3		-> start3
-> end1		-> end3
-> end3		-> end2

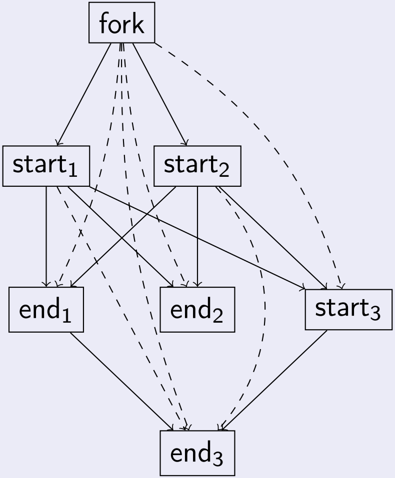
$\rightarrow$ 16 possible runs

Combinatorial and Broadcasting Language (CBL)

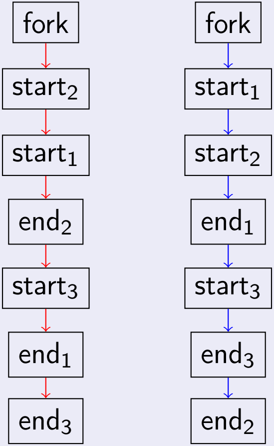
- concurrent language with synchronisation barriers
- CBL deadlock free programs are DAG (directed acyclic graphs) isomorphics

Concurrency theory → Order theory

Partial order (poset)

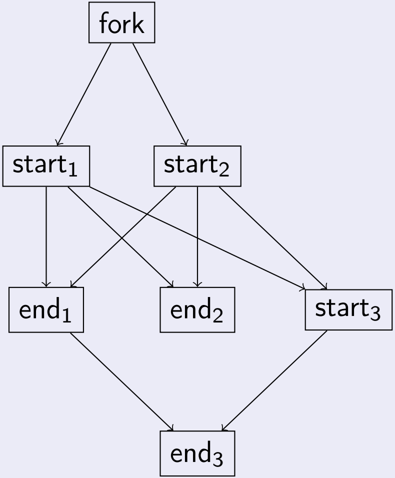


Linear extensions

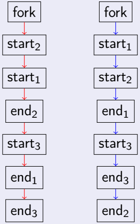


Concurrency theory → Order theory → (Analytic) Combinatorics

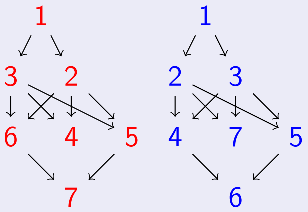
Covering DAG



Linear extensions



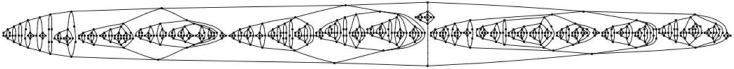
Increasing labellings



Theoretical problems

Problems

1. quantitative study of the “combinatorial explosion” phenomena
   - relation between the program structure and its linear extensions number
   - study of the charateristic patterns of the runs
2. uniform random generation study
   - extend the Boltzmann generation to increasing structures
   - study of increasing labelling algorithms, not much studied too



Practical problems

Problems

1. quantitative study of "combinatorial explosion" phenomena
   $\Rightarrow$ algorithm to count the linear extensions number
2. study of the uniform random generation
   $\Rightarrow$ uniform random generation algorithm

Important results

- the counting problem is $\sharp P$-complet [Brightwell & Winkler '91]
- an uniform random generation algorithm of expected complexity $\mathcal{O}(n^3 \log n)$ already exists [Huber '06]

Applications

- Random testing [Oudinet'10, Sen'07]
- Monte Carlo model checking [Grosu & Smolka '05]

Bestiary

Uniform ranfom generation | Counting $\qquad$ $n =$ synctactic size

Diamonds* [BDFGH'16]
$\mathcal{O}(n \log n) \mid \mathcal{O}(n)$

Trees [BGP'16]
$\mathcal{O}(n \log n) \mid \mathcal{O}(n)$

$\subset \qquad \supset$

Series Parallel posets
$\mathcal{O}(n\sqrt{n})^* \mid \mathcal{O}(n)$ [Möhring'87]

$\cap$

Cycle-free posets
$\mathcal{O}(n^3 \log n) \mid \mathcal{O}(?)$

$\cap$

Posets
$\mathcal{O}(n^3 \log n)$ [Huber'06] | ♯P-complet [BW'91]

$*$: our contributions

Bestiary

Uniform ranfom generation | Counting

n = synctactic size

Diamonds* [BDFGH'16]
$\mathcal{O}(n \log n)$ | $\mathcal{O}(n)$

Trees [BGP'16]
$\mathcal{O}(n \log n)$ | $\mathcal{O}(n)$

$\subset$ $\supset$

Series Parallel posets
$\mathcal{O}(n\sqrt{n})^*$ | $\mathcal{O}(n)$ [Möhring'87]

$\cap$

Cycle-free posets
$\mathcal{O}(n^3 \log n)$ | $\mathcal{O}(?)$

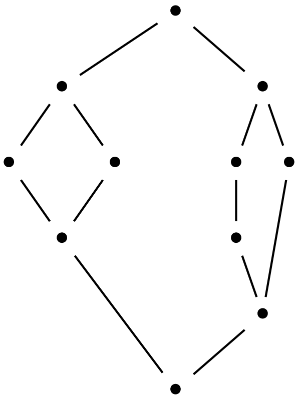
$\cap$

Posets
$\mathcal{O}(n^3 \log n)$ [Huber'06] | $\sharp$P-complet [BW'91]

$*$: our contributions

Outline

Combinatorial specifications

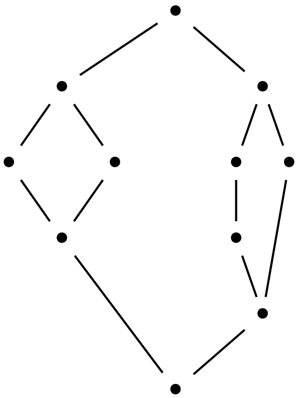


Unlabelled structures

$$\mathcal{S} = \bullet + (\bullet \times G(\mathcal{S}) \times \bullet)$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Combinatorial specifications

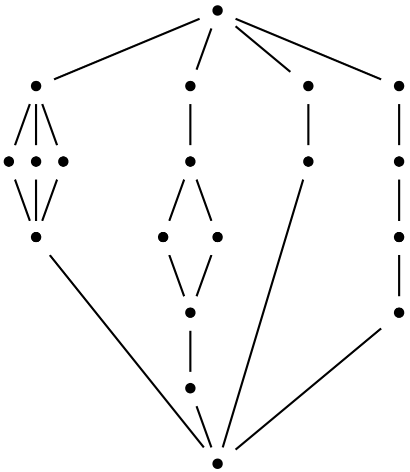


Unlabelled structures

$$\mathcal{S} = \bullet + \left(\bullet \times (\mathcal{E} + \mathcal{S}^2) \times \bullet\right)$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Combinatorial specifications

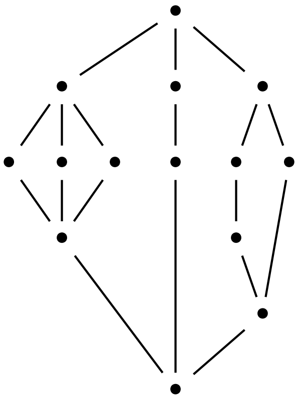


Unlabelled structures

$$\mathcal{S} = \bullet + (\bullet \times \mathrm{Seq}(\mathcal{S}) \times \bullet)$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Combinatorial specifications

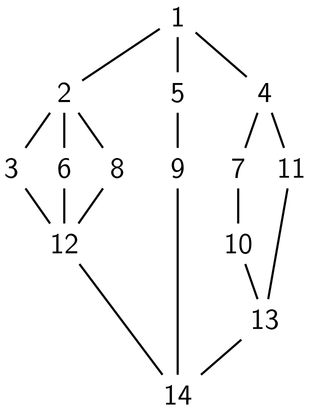


Unlabelled structures

$$\mathcal{S} = \bullet + \left(\bullet \times (\mathcal{E} + \mathcal{S}^2 + \mathcal{S}^3) \times \bullet\right)$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Combinatorial specifications



Unlabelled structures

$$\mathcal{S} = \bullet + (\bullet \times G(\mathcal{S}) \times \bullet)$$

Increasing labellings

$$\mathcal{I} = \bullet_{lbl} + (\bullet^{\square}_{lbl} \star G(\mathcal{I}) \star \bullet^{\blacksquare}_{lbl})$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

The symbolic method [Flajolet, Sedgewick '10]

From combinatorics to complex analysis

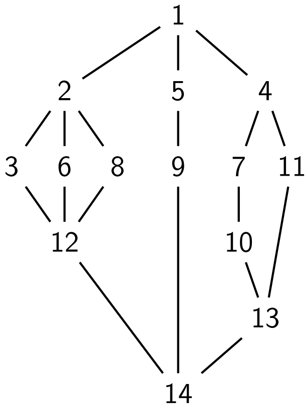
Let a_n be the number of objects of size n in $\mathcal{A}$. Let A, the exponential generating function of $\mathcal{A}$:

$$A(z) = \sum_{n \geqslant 0} \frac{a_n}{n!} z^n$$

The symbolic method gives the translation of specifications into functional equations satisfied by the generating functions.

Specification	Generating function
$\mathcal{E}$	1
$\bullet$	z
$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$A = B + C$
$\mathcal{A} = \mathcal{B} \star \mathcal{C}$	$A = B \times C$
$\mathcal{A} = \mathcal{B}^{\square} \star \mathcal{C}$	$A = \int B' \times C$

Generating Functions



Unlabelled structures

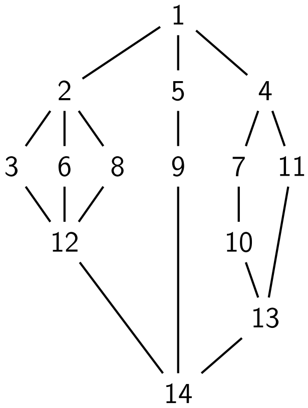
$$\mathcal{S} = \bullet + (\bullet \times G(\mathcal{S}) \times \bullet)$$

Increasing labellings

$$\mathcal{I} = \bullet_{lbl} + (\bullet^{\square}_{lbl} \star G(\mathcal{I}) \star \bullet^{\blacksquare}_{lbl})$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Generating Functions



Unlabelled structures

$$\mathcal{S} = \bullet + (\bullet \times G(\mathcal{S}) \times \bullet)$$

Increasing labellings

$$\mathcal{I} = \bullet_{lbl} + (\bullet^{\square}_{lbl} \star G(\mathcal{I}) \star \bullet^{\blacksquare}_{lbl})$$

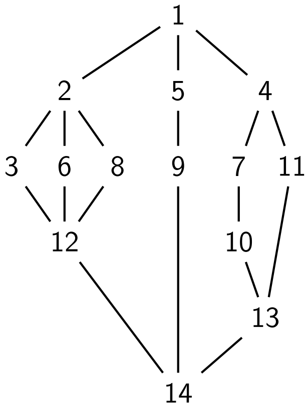
Functionnal equation

(Symbolic method)

$$I(z) = z + \int_0^z \int_0^t G(I(u))\, \mathrm{d}u\, \mathrm{d}t$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Generating Functions



Unlabelled structures

$$\mathcal{S} = \bullet + (\bullet \times G(\mathcal{S}) \times \bullet)$$

Increasing labellings

$$\mathcal{I} = \bullet_{lbl} + (\bullet^{\square}_{lbl} \star G(\mathcal{I}) \star \bullet^{\blacksquare}_{lbl})$$

Differential equation

$$\begin{cases} I'' = G(I) \\ I(0) = 0 \\ I'(0) = 1 \end{cases}$$

O. Bodini, M. D., X. Fontaine, A. Genitrini, H.-K. Hwang 2016

Elliptic cases : binary and ternary diamonds

Weierstrass case : binary diamonds

$$\mathcal{B} = \bullet + (\bullet^{\square} \star (\mathcal{E} + (\mathcal{B} \star \mathcal{B})) \star \bullet^{\blacksquare}) \qquad B'' = 1 + B^2$$

$$b_n = 6\frac{(n+1)!}{\rho^{n+2}} \sum_{(k,l)\in\mathbb{Z}^2} \frac{1}{\left(1 + \frac{k\omega_1}{\rho} + \frac{l\omega_2}{\rho}\right)^{n+2}} \underset{n\to\infty}{\sim} 6\frac{(n+1)!}{\rho^{n+2}}$$

Jacobi's case : ternary diamonds

$$\mathcal{T} = \bullet^{\square} \star (\mathcal{E} + (\mathcal{T} \star \mathcal{T} \star \mathcal{T})) \star \bullet^{\blacksquare} \qquad T'' = 1 + T^3$$

$$t_n = \frac{\sqrt{2}\; n!}{\rho^{n+1}} \sum_{(k,l)\in\mathbb{Z}^2} \frac{1}{(1 + C_{k,l})^{n+1}} - \frac{1}{(2 + C_{k,l})^{n+1}} \underset{n\to\infty}{\sim} 6\sqrt{2}\frac{(n+1)!}{\rho^{n+1}}$$

with $C_{k,l} = \frac{3k}{2} + i\frac{\sqrt{3}}{2}(k+2l)$

Increasing labellings asymptotics

For diamonds of size n

non-plane diamonds ($G = \mathrm{Set}$) :

$$e_n = \frac{2^{n+1}\ (n-1)!}{\pi^n} \sum_{j=-\infty}^{+\infty} \frac{1}{(1+4j)^n}$$

plane diamands of fixed arity ($G = P(x)$ and $deg(P) = m$) :

$$e_n = n! \left(\frac{\sqrt{2(m+1)}}{(m-1)\sqrt{b_m}} \right)^{\frac{2}{m-1}} \frac{n^{-\frac{m-3}{m-1}}}{\Gamma(\frac{2}{m-1})} \rho^{-n-\frac{2}{m-1}} \left(1 + \mathcal{O}\left(n^{-\frac{4}{m-1}}\right)\right)$$

plane general diamonds ($G = \mathrm{Seq}$) :

$$e_n = \frac{n!\rho^{1-n}}{n^2\sqrt{2\log n}} \left(\sum_{0 \leqslant k < K} \frac{P_k(\log\log n)}{(\log n)^k} + \mathcal{O}\left(\frac{(\log\log n)^K}{(\log n)^K} \right) \right)$$

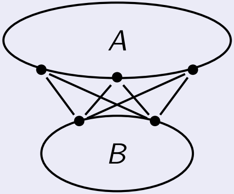
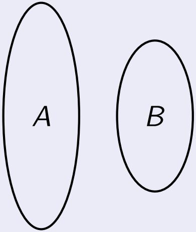
Outline

A more realistic model : the series parallel posets

Definitions

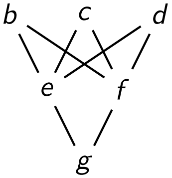
The class of Series Parallel posets is the class containing the singleton order and stable by disjoint union and series composition

Operators definition

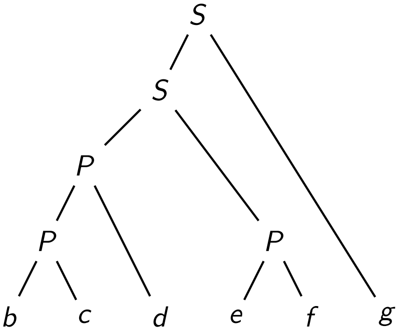
Series composition : $\forall x \in A, y \in B, x \leq y$	Disjoint union : $\forall x \in A, y \in B$, x and y are not comparable
	

Decomposition tree

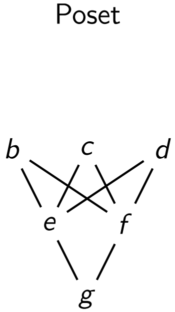
Poset

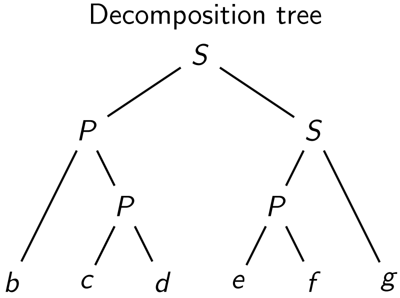


Decomposition tree



Decomposition tree

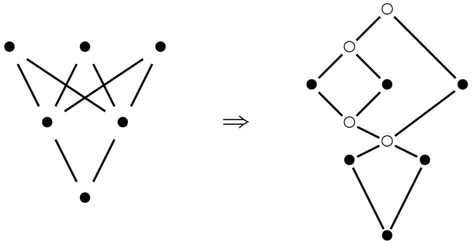




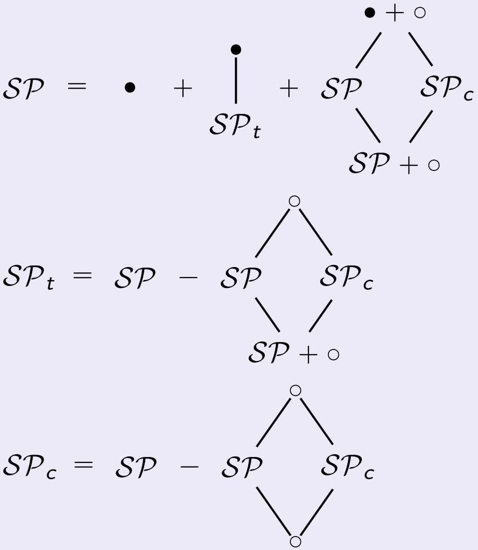
An unambiguous decomposition

Principles

- *binarize* nodes with arity more than 2 by left leaning
- add an unique top and an unique bottom



An unambiguous decomposition



An unambiguous decomposition

Specification

Using this specification, we obtain :

- $\mathcal{SP}$ the specification of all the series parallel posets
- $\mathcal{SP}_c$ the specification of the connected series parallel posets

$\Rightarrow$ an algebraic series solution in both cases

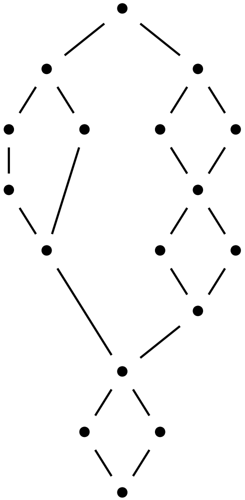
$$0 = (u + z - uz)\mathcal{SP}^3 + u(z + uz)\mathcal{SP}^2 + (z - 1)\mathcal{SP} + z$$

An interresting subclass

Forgetting the two kinds of nodes, we remark that the structures are isomorphic to unary-ternary trees.

$$\mathcal{F} = z + z\mathcal{F} + z\mathcal{F}^3$$

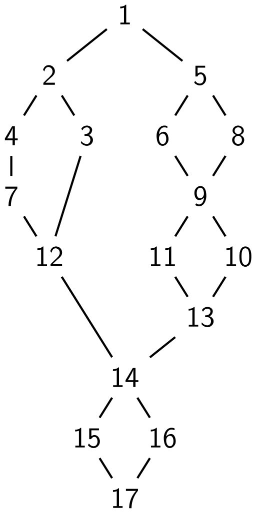
Increasing labellings of Series Parallels



Unlabelled sructures

$$\mathcal{F} = \mathcal{Z} + \mathcal{Z} \cdot \mathcal{F} + \mathcal{Z} \cdot \mathcal{F}^2 \cdot \mathcal{F}$$

Increasing labellings of Series Parallels



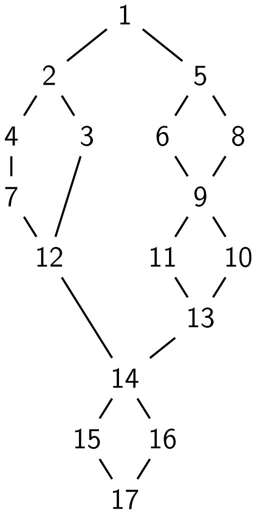
Unlabelled sructures

$$\mathcal{F} = \mathcal{Z} + \mathcal{Z} \cdot \mathcal{F} + \mathcal{Z} \cdot \mathcal{F}^2 \cdot \mathcal{F}$$

Increasing labellings

$$\mathcal{C} = \mathcal{Z} + \mathcal{Z}^{\square} \star \mathcal{C} + \mathcal{Z}^{\square} \star \mathcal{C}^2 \; ? \; \mathcal{C}$$

Increasing labellings of Series Parallels



Unlabelled sructures

$$\mathcal{F} = \mathcal{Z} + \mathcal{Z} \cdot \mathcal{F} + \mathcal{Z} \cdot \mathcal{F}^2 \cdot \mathcal{F}$$

Increasing labellings

$$\mathcal{C} = \mathcal{Z} + \mathcal{Z}^{\square} \star \mathcal{C} + \mathcal{Z}^{\square} \star \mathcal{C}^2 \boxed{\star} \mathcal{C}$$

Current work : the ordered product

Definition

$$\mathcal{C} = \mathcal{A} \boxdot \mathcal{B} \qquad C(z) = \mathcal{B}(\mathcal{L}[A](z) \times \mathcal{L}[B](z))$$

avec $$\mathcal{L}[g](z) = \sum_{n=0}^{\infty} n! g_n z^n \qquad \mathcal{B}[g](z) = \sum_{n=0}^{\infty} g_n \frac{z^n}{n!}$$

Property

- fit in the symbolic method as a boxed product generalisation (which is a particular case of the ordered product, when one of the operand is an atom)
- commutative and associative
- holonomy stability

Current work : the ordered product

A transfer theorem

Let A and B exponential generating functions, then

- if $[z^n]A = o([z^n]B)$ then $[z^n]A \boxdot B \sim [z^{n-val(A)}]B$
- else, $[z^n]A = \Theta([z^n]B)$ and $[z^n]A \boxdot B$ depends of the growing speed of $[z^n]A$

Current work : the ordered product

A transfer theorem

Let A and B exponential generating functions, then

- if $[z^n]A = o([z^n]B)$ then $[z^n]A \boxdot B \sim [z^{n-val(A)}]B$
- else, $[z^n]A = \Theta([z^n]B)$ and $[z^n]A \boxdot B$ depends of the growing speed of $[z^n]A$

A hard problem

What is the asymptotic behaviour of the increasing labellings number of Series Parallel :

$$\mathcal{C} = \mathcal{Z} + \mathcal{Z}^{\square} \star \mathcal{C} + \mathcal{Z}^{\square} \star \mathcal{C}^2 \boxdot \mathcal{C}$$
$$[z^n]C = ??$$

Current work : the ordered product

A transfer theorem

Let A and B exponential generating functions, then

- if $[z^n]A = o([z^n]B)$ then $[z^n]A \boxdot B \sim [z^{n-val(A)}]B$
- else, $[z^n]A = \Theta([z^n]B)$ and $[z^n]A \boxdot B$ depends of the growing speed of $[z^n]A$

A hard problem

What we think :

$$C(z) \underset{z \to \rho_C}{=} K(z-\rho_C)^{-2} + K'(z-\rho_C)^{-1} + \mathcal{O}(1)$$

and C is a rational fraction of Weierstrass $\wp$ functions.

Outline

Counting algorithm

Möhring'87

Note $\mathscr{LE}(P)$ the number of linear extensions of the poset P. Let P and Q two Series parallel posets, then

- $\mathscr{LE}(P.Q) = \mathscr{LE}(P) \cdot \mathscr{LE}(Q)$
- $\mathscr{LE}(P \parallel Q) = \binom{|P|+|Q|}{|P|} \cdot \mathscr{LE}(P) \cdot \mathscr{LE}(Q)$

Counting algorithm

Möhring'87

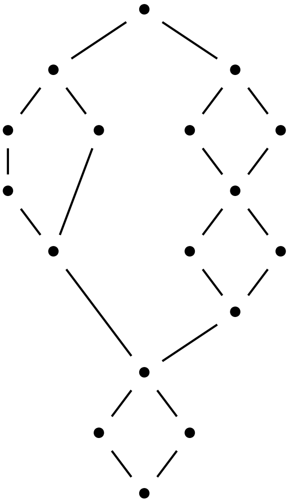
Note $\mathscr{LE}(P)$ the number of linear extensions of the poset P. Let P and Q two Series parallel posets, then

- $\mathscr{LE}(P.Q) = \mathscr{LE}(P) \cdot \mathscr{LE}(Q)$
- $\mathscr{LE}(P \parallel Q) = \binom{|P|+|Q|}{|P|} \cdot \mathscr{LE}(P) \cdot \mathscr{LE}(Q)$

Hook-length formula

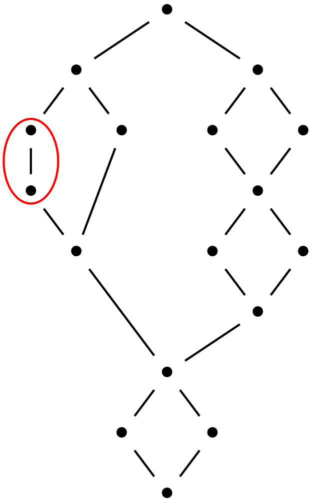
$$\mathscr{LE}(P) = \frac{\prod_{Y \in \mathcal{P}a_P} |Y|!}{\prod_{X \mathcal{S}e_P} |X|!}$$

Counting algorithm



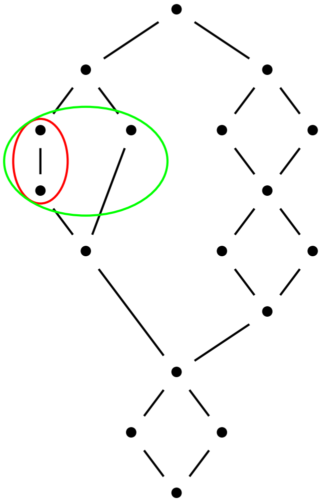
$\mathscr{L}\mathscr{E}(P) =$ ________________

Counting algorithm



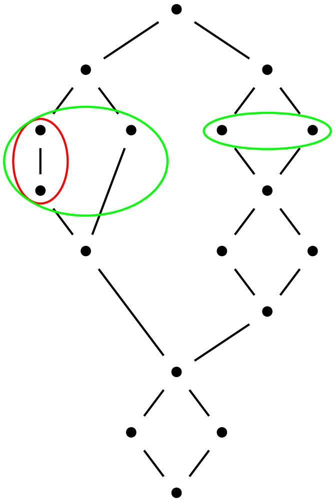
$$\mathscr{L}\mathscr{E}(P) = \frac{}{2!}$$

Counting algorithm



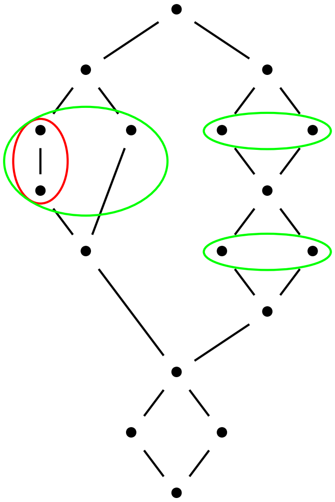
$$\mathscr{L}\mathscr{E}(P) = \frac{3!}{2!}$$

Counting algorithm



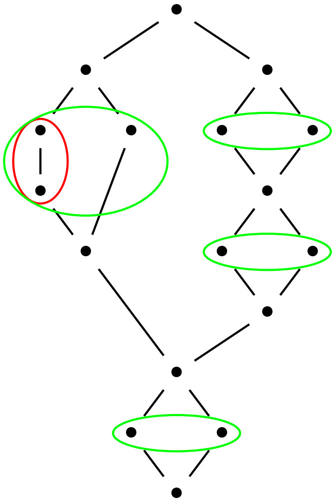
$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!}{2!}$$

Counting algorithm



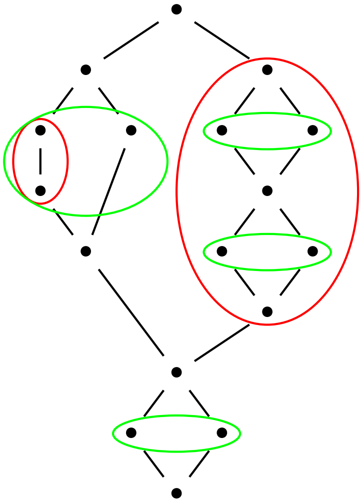
$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!\ 2!}{2!}$$

Counting algorithm



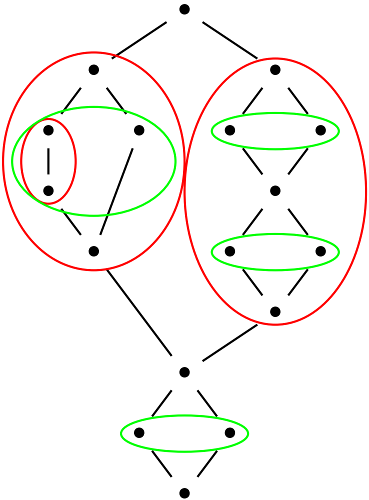
$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!\ 2!\ 2!}{2!}$$

Counting algorithm



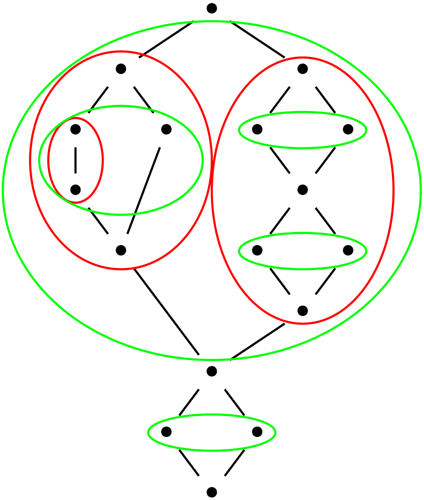
$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!\ 2!\ 2!}{2!\ 7!}$$

Counting algorithm



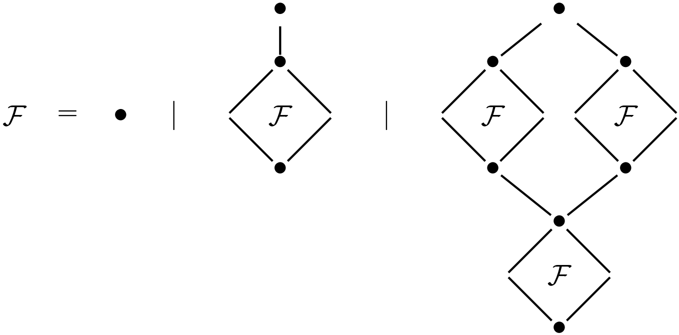
$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!\ 2!\ 2!}{2!\ 7!\ 5!}$$

Counting algorithm



$$\mathscr{L}\mathscr{E}(P) = \frac{3!\ 2!\ 2!\ 2!\ 12!}{2!\ 7!\ 5!} = 19008$$

Uniform random generation

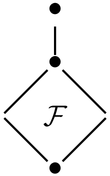


Uniform random generation

partial order	$\Rightarrow$	linear extension
$\bullet$	$\Rightarrow$	return (1)

Uniform random generation

partial order $\Rightarrow$ linear extension

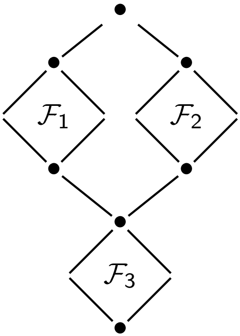


$\Rightarrow$

$(x_2, \dots x_n) :=$ draw_lin_ext$(\mathcal{F})$
return $(1, x_2 + 1, \dots, x_n + 1)$

Uniform random generation

partial order $\Rightarrow$ linear extension



$\Rightarrow$

$\mathbf{x} :=$ draw_lin_ext$(\mathcal{F}_1)$
$\mathbf{y} :=$ draw_lin_ext$(\mathcal{F}_2)$
$\mathbf{z} :=$ draw_lin_ext$(\mathcal{F}_3)$
$\mathbf{t} :=$ shuffle$(\mathbf{x}, \mathbf{y})$ $\quad |\mathbf{t}| = |\mathbf{x}| + |\mathbf{y}|$
return $(1,\ \mathbf{t} + \mathbb{1},\ \mathbf{z} + \mathbb{1} \cdot (|\mathbf{t}| + 1))$

Average complexity

The average complexity of `draw_lin_ext`, in memory writings, is in $\mathcal{O}(n\sqrt{n})$.
Moreover, the algorithm is **bit-optimal**.

Still 492 days to finish my phd ...

Current work

- work on ☒ (in the case of Series Parallel posets)
- accurate analysis of the random generation algorithm

Futur works

- study more general classes (like Cycle-free posets)
- add expressivity to the starting language with non-determinism
- test practical applications (implementation of a statistical model checker)