Gábor CZÉDLI April 5, 2023

Generating the principal filters of quasiorder lattices and, in connection with it, a way from Zádori's method to a secret key cryptography

Czédli Gábor 2023. április 5.

Kvázirendezés-hálók nagy főfiltereinek generálása és ezzel kapcsolatban: út a Zádori-módszertől egy titkosírásig

This is the pdf version. During the talk, Xournal++ (version 1.1.3) (/ˌzə-nl̩ ˈplʌsˈplʌs/), a free and open-source note-taking software was running.

5 PhD Students with 6 papers

(2 #)

References:

Delbrin Ahmed and CzG: (1+1+2)-generated lattices of quasiorders. ASM 2021, pp 415–427

I. Chajda and CzG: How to generate the involution lattice of quasiorders?, Studia Sci. M.Hung 1996, pp 415-427

CzG: Lattice generation of small equivalences of a countable set. Order 13 (1996), 11–16

CzG: Four-generated large equivalence lattices. ASM 1996, pp 47-49

CzG: (1+1+2)-generated equivalence lattices, J. Algebra, 221 (1999), 439–462

CzG: Four-generated quasiorder lattices and their atoms... Comm.Alg. (2017) pp.4037-4049

CzG:Four-generated direct powers of partition lattices and authentication. Publ.Debrecen(2021)447-472

CzG: Generating some large filters of quasiorder lattices. arXiv:2302.13911

CzG: Generating Boolean lattices by few elements and a protocol for authentication and cryptography based on an NP-complete problem. arXiv: http:2303.10790

2023 } most of this talk

CzG and J.Kulin: A concise ... lattices of quasiorders and transitive relations. ASM(2017) pp.3-12

CzG and Lillian Oluoch: Four-element gen. sets ... and their direct products. ASM(2020) pp.405-448

T.Dolgos (M.Maróti's student): Gen. equ. and quasiorder lattices over finite sets. BSc Thesis, SZTE 2015

J. Kulin: Quasiorder lattices are five-generated. Discuss. Math. Gen. Alg. Appl.=2016) 59-70

Lillian Oluoch and Amenah Al-Najafi: Discuss. Math. Gen. Algebra and Appl. 42(2022) 327–338

H. Strietz: Finite partition lattices are four-generated. Proc. Lattice Th. Conf. Ulm, 1975, pp 257-259

G.Takách: Three-generated quasiorder lattices. Discuss. Math. Alg.& Stoch.(1996) pp. 81-98

Vilmos Totik: Oral communication. Somewhen in the nineties.

L. Zádori: Generation of finite partition lattices. Lectures in universal algebra (Proc. Colloq.Szeged, 1983),North-Holland, 1986,pp. 573–586.

Part 1. Generating Boolean lattices (3 #)

$B_n = (B_n; \vee, \wedge)$ stands for the Boolean lattice with n atoms. Note that $B_n \cong \mathrm{Pow}(\{1, \ldots, n\})$.

For a lattice L, $\mathrm{mg}(L)$ is the smallest k such that L has a k-element generating set.

(a) What is $\mathrm{mg}(B_4)$? **4**

(b) What is $\mathrm{mg}(B_{1000})$? **13**

(c) Monster := $B_{112\,186\,277\,816\,662\,845\,432}$
(Subscript $\approx 1.12 \cdot 10^{20}$) What is mg(Monster)? **70**

(#4)

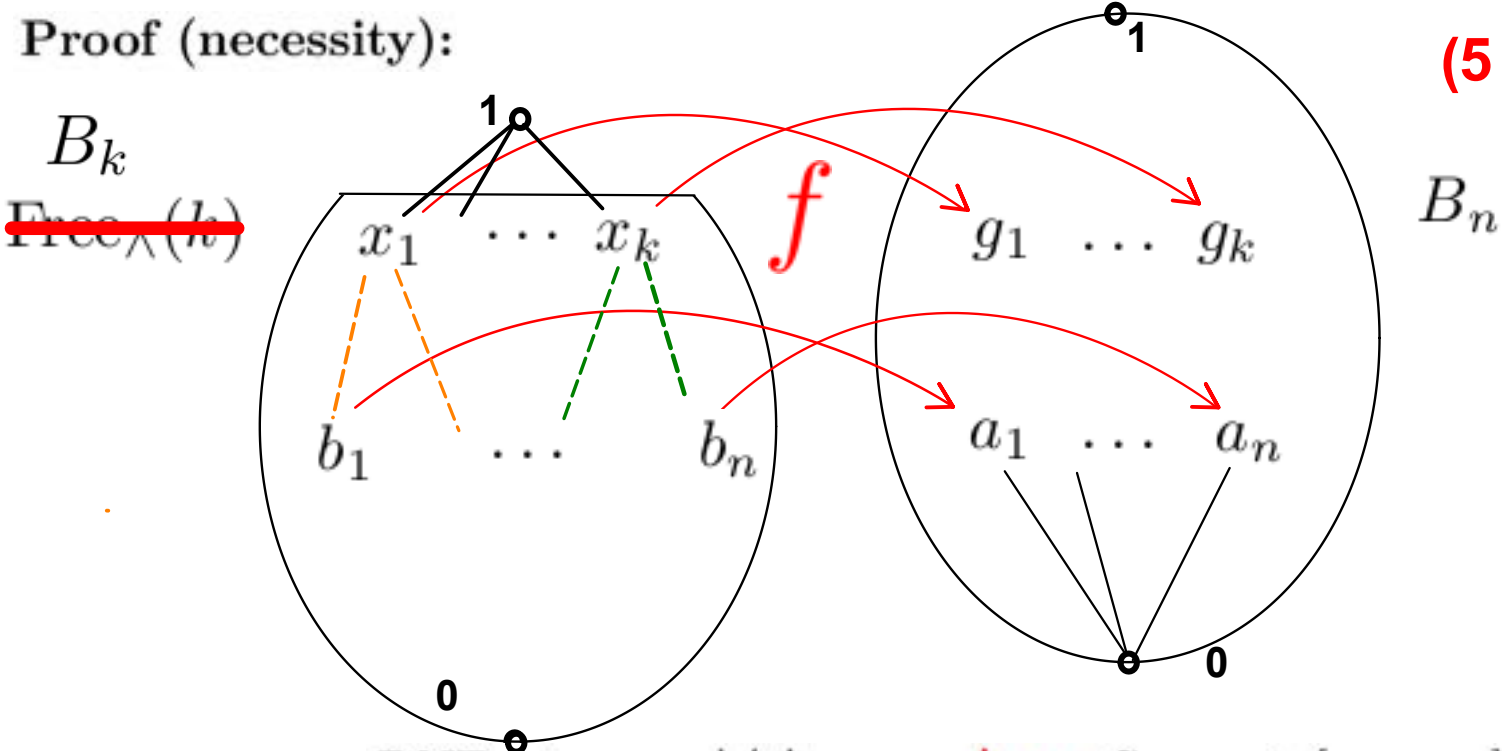
$$\mathrm{Sp}(n) := \binom{n}{\lfloor n/2 \rfloor}$$

, let $\mathrm{LASp}(n)$ be the smallest $k \in \mathbb{N}^+$ such that $n \leq \mathrm{Sp}(k)$.

Theorem. For $n \in \mathbb{N}^+$, $\mathrm{mg}(B_n) = \mathrm{LASp}(n)$.

Proof (necessity):

(5 #)

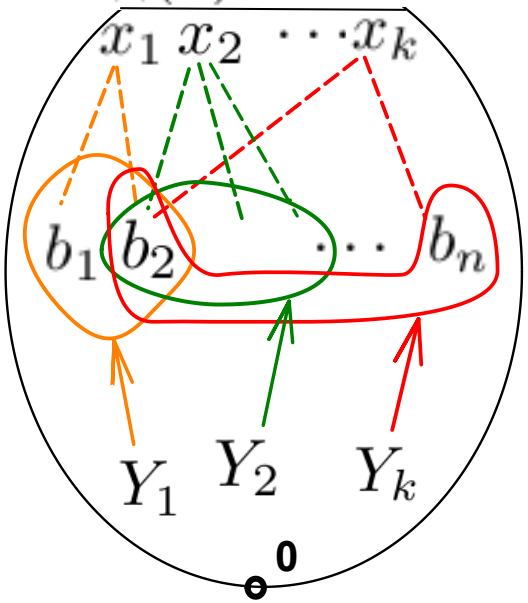


DNF $\Rightarrow a_1 = \bigvee \bigwedge g_{ij} = \bigwedge g_j$. So $a_1 \in [g_1, \ldots g_k]_\wedge$. Hence $a_1 \in f(\mathrm{Free}_\wedge(k))$. Thus $\exists b_1$ such that $f(b_1) = a_1$. As the a_i's form an antichain, so do the b_j's.

Sperner's thm. $\Rightarrow n \le \mathrm{Sp}(k)$. $\surd$

Sufficiency. For $n' < n$, $\exists B_n \twoheadrightarrow B_{n'}$. So we can assume $n = \mathrm{Sp}(k)$ rather than "$\le$". Taking the meet of $\lfloor k/2 \rfloor$ free generators in $\forall$ ways, we get an $\mathrm{Sp}(k) = \binom{k}{\lfloor k/2 \rfloor} =$ n-element antichain $\{b_1, \ldots, b_n\} \subseteq \mathrm{Free}_\wedge(k)$.

$\mathrm{Free}_\wedge(k)$



Let $Y_j := \{b_i : b_i \leq x_j\}$ for $j = 1, \ldots, k$. In particular, $Y_2 := \{b_i : b_i \leq x_2\}$. Let $H_\ell := \bigcap\{Y_j : b_\ell \in Y_j\}$. E.g., $H_2 := \bigcap\{Y_j : b_2 \in Y_j\}$. Clearly, $\{b_2\} \subseteq H_2$.

For contradiction, suppose $b_3 \in H_2$.

So $b_3 \in \bigcap\{Y_j : b_2 \in Y_j\}$.

I.e., $\forall j,\ b_2 \in Y_j \Rightarrow b_3 \in Y_j$.

I.e., $\forall j,\ b_2 \leq x_j \Rightarrow b_3 \leq x_j$.

$\subseteq$

So $b_2 = \bigwedge\{x_j : b_2 \leq x_j\} \geq \bigwedge\{x_j : b_3 \leq x_j\} = b_3$; contradiction. Thus $\{b_2\} = H_2$, whence $\{b_2\}$ belongs to $[Y_1, \ldots, Y_k]$. So do each of $\{b_1\}, \ldots, \{b_n\}$. So $\{Y_1, \ldots, Y_k\}$ generates $\mathrm{Pow}(\{b_1, \ldots, b_n\}) \cong B_n$, q.e.d.

Part 2. Generating quasiorder lattices and equivalence lattices

$\mathcal{Q}(A)$ or $\mathcal{Q}(|A|)$ is the lattice of quasiorders (=preorders) of A. $\mathcal{E}(A)$ or $\mathcal{E}(|A|)$: the equivalence lattice of A.

$n \in \mathbb{N}^+$: Zádori(1983), Strietz(1975): $\mathcal{E}(n)$ is 4-generated.

$\kappa_0 := \aleph_0$, $\kappa_{n+1} := 2^{\kappa_n}$. Chajda and Czédli (1996): if $\lambda \leq \kappa_n$, then $\mathcal{Q}(\lambda)$ is 3^*-generated and so 6-generated.

Czédli(1996): $\mathcal{E}(\lambda)$ is 4-generated provided that λ is accessible ($\forall\lambda$ is such in Kuratowski's model of ZFC).

Takách (1996): if λ is *accessible* then $\mathcal{Q}(\lambda)$ is 3^*-generated and so 6-generated.

Dolgos (2015): if $\lambda \leq \aleph_0$, then $\mathcal{Q}(\lambda)$ is 5-generated.

Kulin (2016): for λ accessible, $\mathcal{Q}(\lambda)$ is 5-generated.

Czédli–Kulin (2017) $4 \neq \lambda$ is accessible $\Rightarrow$ $\mathcal{Q}(\lambda)$ is 4-generated.

Czédli (2017): if $3 \leq \lambda$, then $\mathcal{Q}(\lambda)$ is **not** 3-generated.

For almost $\forall$ accessible λ, $\mathcal{Q}(\lambda)$ and $\mathcal{E}(\lambda)$ are (1+1+2)-generated: Strietz, Zádori, Czédli, Kulin, Delbrin Ahmed, Lillian Oluoch.

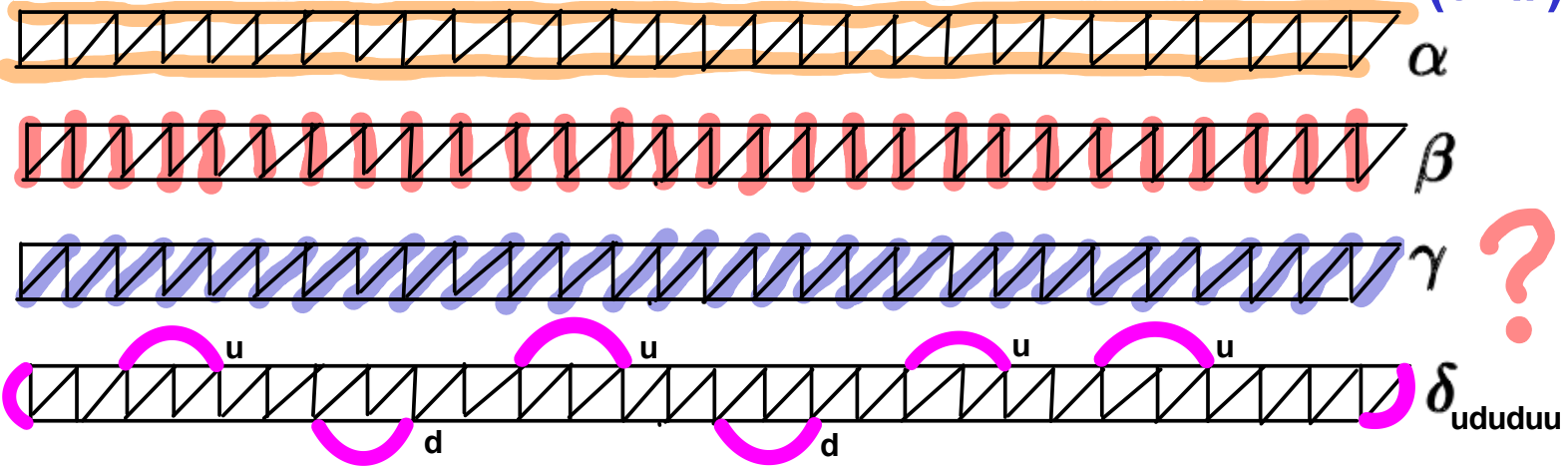
Tran(λ) is 8-generated for $\lambda \leq \aleph_0$ by T. Dolgos; it is 6- and 3^*-generated for λ accessible by Czédli and Kulin (2017)

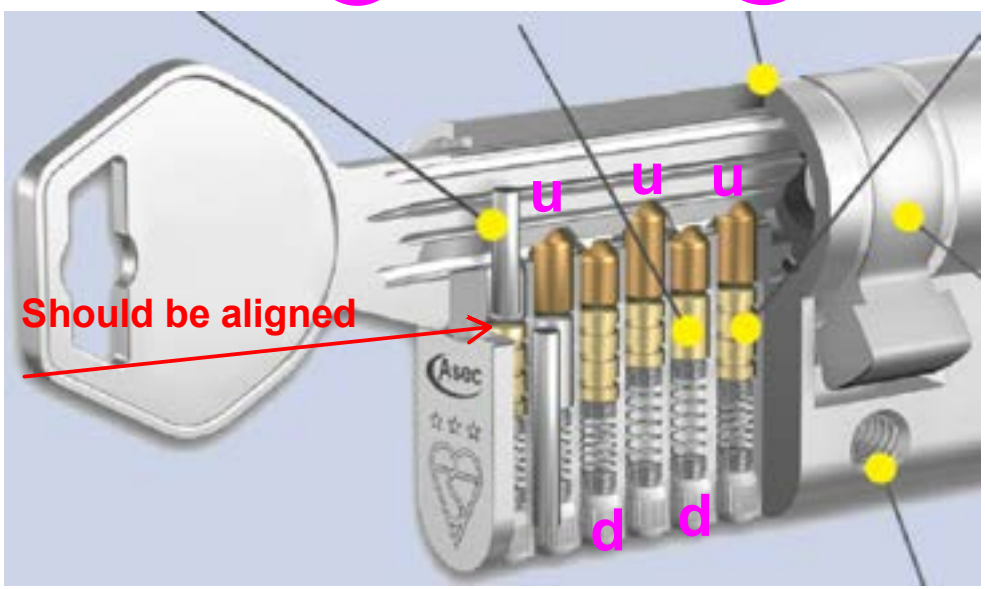
Many open problems, **e.g.**, is $\mathcal{Q}(4)$ 4-gen.? For $n \in \{4, 5, 7, 8, 9, 10, 12\}$, is $\mathcal{Q}(n)$ (1+1+2)-gen.? Reduce "3^*-"? Inaccessible λ?

Vilmos Totik: proved (in a minute or so!) that my method is not appropriate for an inaccessible λ. (Partly, this is why my attention turned to better plans, leading to today's talk.)

Plan: 25 min.

Zádori's generators: $[\alpha, \beta, \gamma, \delta] = \mathcal{E}(57)$





Some other tricks are also needed.

Even $\mathcal{E}(100)^{3 \cdot 10^{89}}$ is 4-generated. $\mathcal{E}(100)$ has many 4-element generating sets.

An excerpt from Czédli and Oluoch's paper (2000):

Proof of Theorem 4.4. In addition to earlier experience with generating sets of partitions lattices, see the References section, an idea is also borrowed from locksmithing. In a traditional pin tumbler cylinder lock, there are *pins* of different (to be more precise, not necessarily equal) lengths, and a key can turn the plug if and only if the heights of its *ridges* match these lengths. The $\sum\{[\![I_i^{\varphi}]\!]^{\mathrm{e}} : x_i = 1,\ 1 \le i \le m\}$

4.4. In addition to earlier experien
e the References section, an idea
litional pin tumbler cylinder lock, t
ıecessarily equal) lengths, and a ke
s *ridges* match these lengths. The

Plan: 30 min.

Gábor Czédli and Lillian Oluoch: Four-element generating se Acta Sci. Math. (Szeged) 86, 405–448 (2020) **(11 #)**

n	4	5	6	7	8	9
$100p(n)$	3.6630037	1.9595531	1.613014768			
N	10 000 000	10 000 000	10 000 000	15 000 000	500 000	25 000
time	8 minutes	27 min	3h+33min	102 hours	95 h	166 h
s	367 221	196 243	161 768	238 223	8 244	438
$100\widehat{p}(n)$	3.67221	1.96243	1.61768	1.58815	1.64880	1.75200

≈ a week

grow

Percentages of the generating ones among the 4-element subsets of $\mathcal{E}(n)$; the paper contains some other tables and confidence intervals, too.

G. Czédli: Four-generated direct powers Publicationes Math. (Debrecen) 99 (2021), 447–472.

n	4	5	6	7	8	9
$\lvert\mathrm{Part}(n)\rvert$	15	52	203	877	4 140	21 147
$\lvert\forall$8-sets$\rvert$	6435	$7.53 \cdot 10^8$	$6.22 \cdot 10^{13}$	$8.41 \cdot 10^{18}$	$2.13 \cdot 10^{24}$	$9.91 \cdot 10^{29}$
$\lvert$tested$\rvert$	100 000	10 000	10 000	6000	1000	284
$\lvert$found$\rvert$	89 780	7 690	7913	5044	848	248
%	89.78	76.90	79.13	84.01	84.80	90.19

73 hours

grow

The same percenteges for 8-element subsets of $\mathcal{E}(n)$.

Plan: 33 min.

For a poset $P = (P; \leq)$, $\mathcal{Q}^\star(P)$ is the lattice of those quasiorders of P that extend "$\leq_P$"; it is a filter of $\mathcal{Q}(A)$.

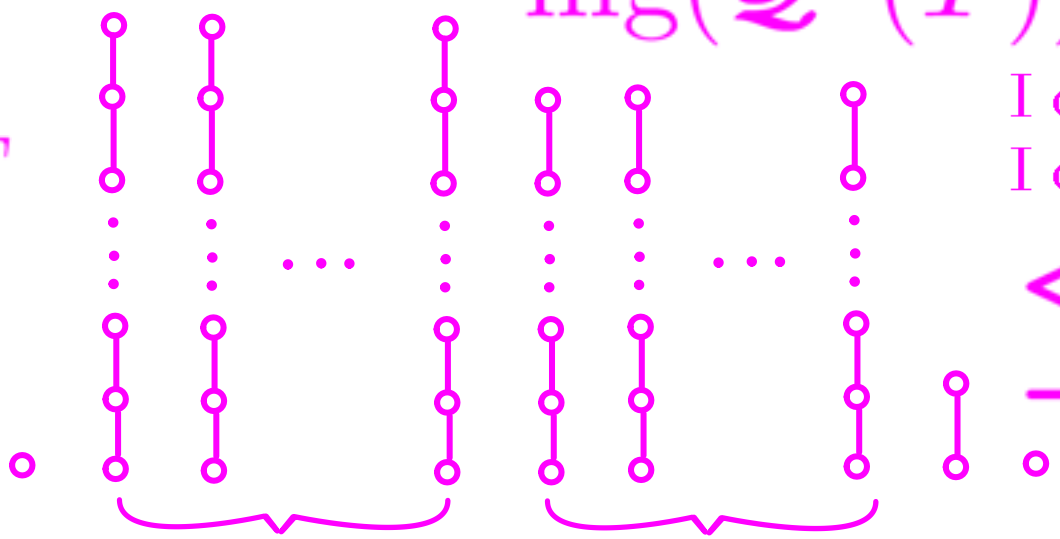
$\mathcal{Q}^\star(P)$ is a filter of $\mathcal{Q}(P)$.

P is a chain of length $4 \cdot 10^9$ (four billion)

$$\mathrm{mg}(\mathcal{Q}^\star(P)) =? 35$$

$$\mathrm{mg}(\mathcal{Q}^\star(T)) =?$$

T



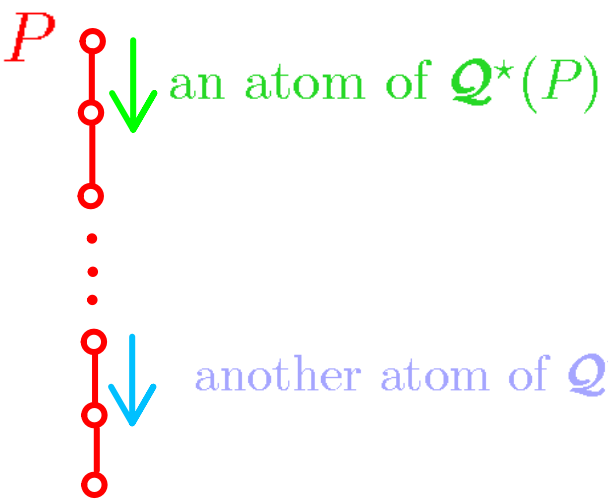
I do not know.
I only know that

$$\leq 43$$

ten billion chains of length four billion

six billion chains of size four billion, i.e., of length 4 billion minus 1

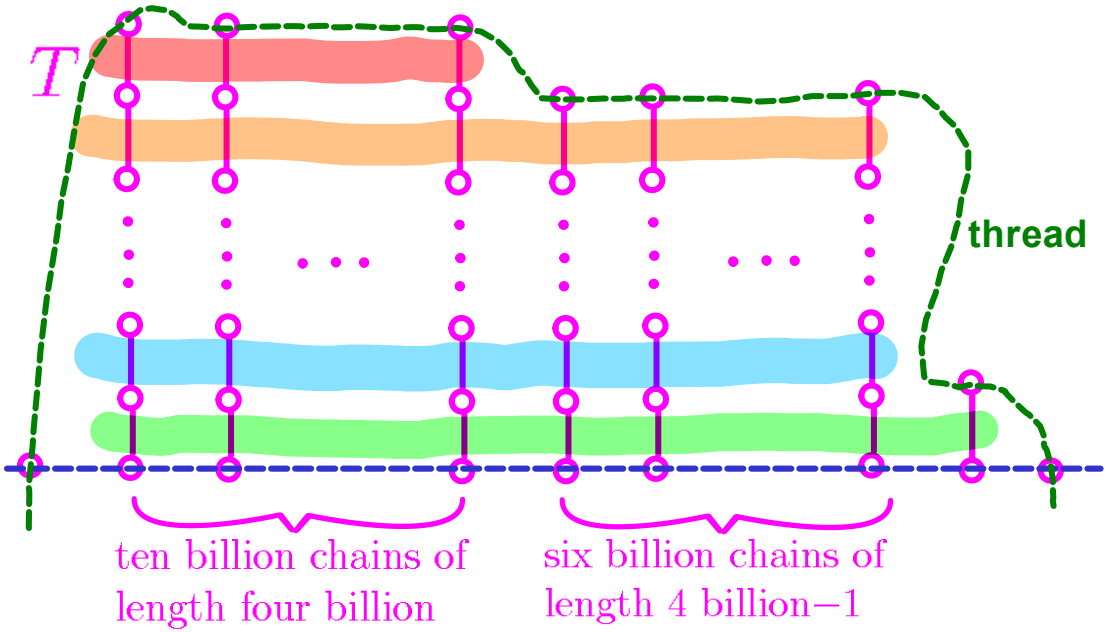
Czédli: Generating some large filters (2023) arXiv:2302.13911 proves more; here I prove only: **Theorem: $= 35$ and ≤ 43.** **(#13)**



Dongseok Kim, Young Soo Kwon, and Jaeun Lee (2014)

$\mathcal{Q}^\star(P) \cong B_{4\,000\,000\,000}$,
so the previous theorem applies.

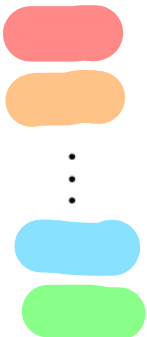
We have proved the "$= 35$" part of the theorem.



(14 #)

The strips consists of "down edges" and partition the set "$\succ$" of down edges.

Strips:



For a strip X, $\mathrm{q}^*(X)$ is the smallest $\in \boldsymbol{\mathcal{Q}}^\star(T)$ that includes X.

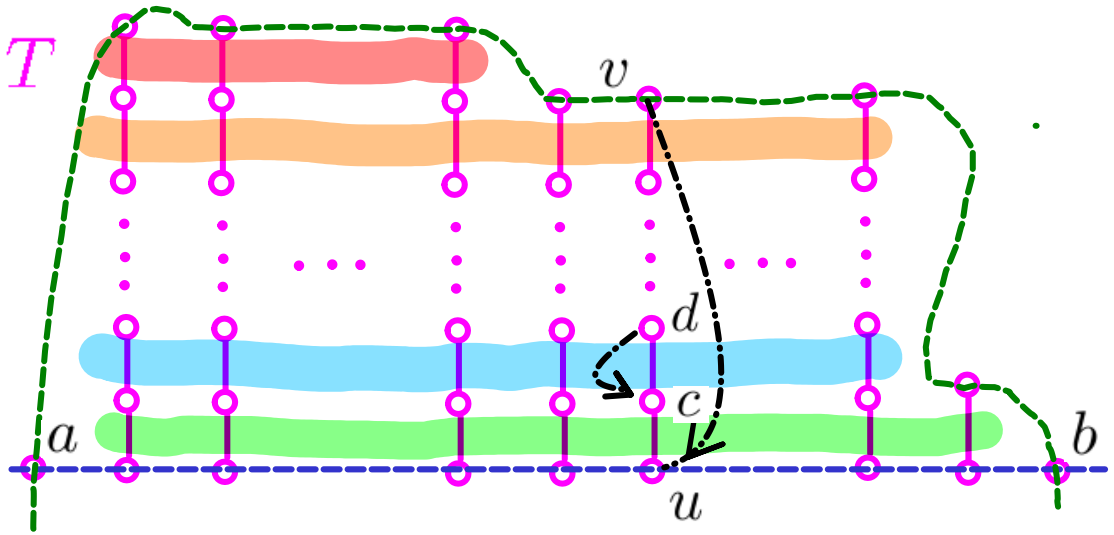
In other words, $\mathrm{q}^*(X)$ is the quasiorder generated by the poset order $\leq_T$ and the down-edges belonging to X.

Let fb := 4 billion. Let B_{fb} denote the powerset lattice of the set of strips. Let $\{S_1, \ldots, S_{35}\}$ be a generating set of B_{fb}.

Denote: $\mathrm{q}^*(S_i) := \bigvee\{\mathrm{q}^*(X) : X \text{ is a strip belonging to } S_i\}$.

Boolean part $\Rightarrow$ $\forall$ strip X, X is the $\bigcap$ of some S_i's. Hence $\mathrm{q}^*(X)$ is the $\bigcap$ of some $\mathrm{q}^*(S_i)$'s. But $\bigcap = \bigwedge$ in $\boldsymbol{\mathcal{Q}}^\star(T)$.

Thus, for $\forall$ strip X, $\mathrm{q}^*(X) \in [\mathrm{q}^*(S_1), \ldots, \mathrm{q}^*(S_{35})]$.



Indeed, say, $\mathrm{q}^*(v,a) \in [G]$ as the four generators of $\mathcal{Q}^\star$(green thread) are in G, etc.

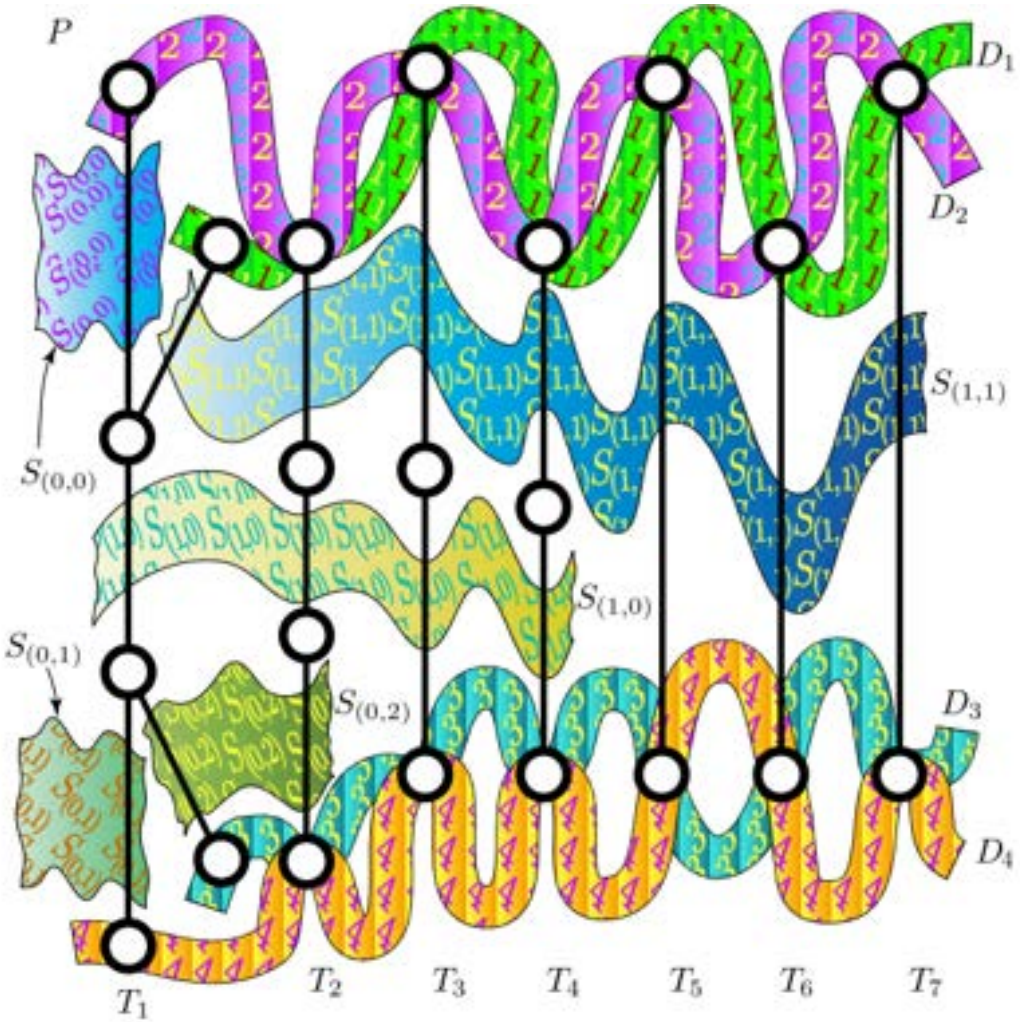
To the 35 $\mathrm{q}^*(S_i)$, add the four generators of $\mathcal{Q}$(green thread), and do the same with the blue thread. Now we have a 43-element subset G of $\mathcal{Q}^\star(T)$. This is how we get $\mathrm{q}^*(v,u)$:

$$\mathrm{q}^*(v,u) = \big(\mathrm{q}^*(v,a) \vee \mathrm{q}^*(a,u)\big) \wedge \big(\mathrm{q}^*(v,b) \vee \mathrm{q}^*(b,u)\big) \in [G].$$

We have seen (at) that for $X =$ $\mathrm{q}^*(X) \in [G]$. Hence, $\mathrm{q}^*(d,c) = \mathrm{q}^*(v,u) \wedge \mathrm{q}^*(X) \in [G]$.

The rest of argument (e.g., how to get $\mathrm{q}^*(x,y)$ with x and y in different components) is omitted. "≤ 43": q.e.d.

The paper takes care of more involved posets, too. E.g.:



I am just preparing a revised version of the paper ...

Part 3. Authentication and cryptography

For $\vec{h} = (h_1, \ldots, h_k) \in B_n^k$, this k-dimensional vector is *generating* if $[h_1, \ldots, h_k] = B_n$.

As LASp(1000) = 13, B_{1000} has a 13-dimensional generating vector. As $k > 13$ grows, B_{1000} has more and more k-dimensional generating vectors.

```
n=1000 k=40  Tested:100000 Generating:    42;   506.867 sec
n=1000 k=50  Tested:100000 Generating: 59003;  1305.780 sec
n=1000 k=50  Tested:100000 Generating: 59162; 10965.266 sec
n=1000 k=60  Tested:100000 Generating: 96878;  1967.896 sec
n=1000 k=70  Tested:100000 Generating: 99823;  2321.272 sec
n=1000 k=80  Tested:100000 Generating: 99990;  2647.147 sec
n=1000 k=90  Tested:100000 Generating: 99999;  2974.364 sec
n=1000 k=100 Tested:100000 Generating:100000;  3265.869 sec
```

The program chose one hundred thousand random k-dimensional vectors and counted the generating ones among them.

When no anomymous usage is permitted: **secret key** cryptography and authentication are used. In our model:

Kate communicates with her Bank.

Let, say, $k := 300$, $n := 1000$, and $t := 100$.

Secret key: a k-dimensional generating vector $\vec{h} = (h_1, \ldots, h_k) \in B_n^k$ of B_n.

$\vec{p} = (p_1, \ldots, p_t)$: a vector of k-ary lattice terms.

Protocol: The Bank sends a random $\vec{p}$ to Kate and she sends $\vec{p}(\vec{h}) \in B_n^t$ back. And vice versa. $\vec{p}(\vec{h})$ is also used as a key of Vernam's cipher.

Adversary: intercepts and/or sends messages. As $\vec{p}$ is always new, he cannot use an intercepted $\vec{p}(\vec{h})$ again. As $\vec{p}(\vec{h})$ can take $|B_n|^t = 2^{nt}$ many values, he cannot find a valid $\vec{p}(\vec{h})$ by chance *unless* he can solve the equation $\vec{p}(\vec{x}) = \vec{c}$, where $\vec{c} := \vec{p}(\vec{h}) \in B_n^t$ is an intercepted constant with the corresponding intercepted vector $\vec{p}$ of lattice terms and $\vec{x} \in B_n^k$ (playing the role of $\vec{h}$) is the unknown.

unless he can solve $\vec{p}(\vec{x}) = \vec{c}$, where $\vec{c} := \vec{p}(\vec{h}) \in B_n^t$ and $\vec{x} \in B_n^k$ **(19 #)**

Observation: even if $n = 1$ and $t = 2$, solving $\vec{p}(\vec{x}) = \vec{c}$ is an NP-complete problem.

Proof: Suppose for contradiction that $\exists$ a polynomial f and an algorithm Γ such that for $\forall$ input $\vec{p}(\vec{x}) = \vec{c}$ of size s, if this input has a solution for $\vec{x}$, then Γ finds a solution in $f(s)$ time.

Then we can solve the 3COLOR as follows. Let G be a simple graph of size s_1. For each vertex i, take the *color variables* r_i, w_i, and g_i. For each edge e: i——j, we take:

Φ_e : $(r_i \vee w_i \vee g_i) \wedge (r_j \vee w_j \vee g_j) = 1$ and

Ψ_e $(r_i \wedge r_j) \vee (w_i \wedge w_j)) \vee (g_i \wedge g_j) = 0$.

} nonempty, disjoint color sets

Let $p_1 := \bigwedge_{e \in \mathrm{Edge}(G)} \Phi_e$, $p_2 := \bigvee_{e \in \mathrm{Edge}(G)} \Psi_i$, $\vec{p} = (p_1, p_2)$.

Let $\vec{x} := (\dots, r_i, w_i, g_i, \dots)$. Let s be the size of $\vec{p}(\vec{x}) = (1, 0)$.

Then $s = \mathrm{const} \cdot s_1$. **Run** Γ for the red input in $f(s)$ steps. After at most $f(s)$ many steps, stop Γ. Then print "YES, G is 3-colorable" if Γ found a solution in that many steps, and print "NO" otherwise. This "magenta algorithm" needs $\approx 2f(s) = 2f(\mathrm{const} \cdot s_1)$ steps, which is a polynomial of s_1. (Factor 2: since we need to count the steps of Γ.)

We have solved 3COLOR in polynomial time. Q.e.d.

Plan: 58min

(20 #)

The value of NP-completeness: Even though we have proved NP-completeness (and so NP-hardness) in case of $n = 1$ and $t = 2$, the Adversary can easily empty Kate's bank account at every 4-th random trial if $n = 1$ and $t = 2$. (Indeed, $\vec{p}(\vec{h}) \in B_n^t = \{0,1\}^2$ can take only four values.) So even if a cryptographic protocol is based on an NP-complete problem, it can be very vulnerable!

Fortunately, for n big, $\vec{p}(\vec{h})$ can take $(2^n)^t$ many values. Hence, the Adversary can hardly find the right one accidentally.

感谢您的关注

Plan: 60min