Breaking the Barrier: A Computation of the Ninth Dedekind Number

Christian Jäkel

Dresden University of Technology

ICFCA 2023

- based on the preprint: *A computation of the ninth Dedekind number* by C. Jäkel, 3. April 2023
- a paper is currently in review

Introduction

- fast growing integer sequence
- very difficult to compute
- introduced in 1897 by Richard Dedekind

Dedekind numbers are the solution to a counting problem, that determines...

Introduction

- fast growing integer sequence
- very difficult to compute
- introduced in 1897 by Richard Dedekind

Dedekind numbers are the solution to a counting problem, that determines...

The Number of Monotone Boolean Functions on n Variables

$f : \{0,1\}^n \to \{0,1\},$ monotone w.r.t. $0 \leq 1$

Example $n = 2$:

- $f(x,y) = 0,\ f(x,y) = 1$
- $f(x,y) = x,\ f(x,y) = y$
- $f(x,y) = x \wedge y,\ f(x,y) = x \vee y$

The monotonic Boolean functions are precisely those that can be defined by combining the inputs using only the operators $\wedge$ and $\vee$.

The Number of Monotone Boolean Functions on n Variables

$f : \{0,1\}^n \to \{0,1\},$ monotone w.r.t. $0 \leq 1$

Example $n = 2$:

- $f(x,y) = 0$, $f(x,y) = 1$
- $f(x,y) = x$, $f(x,y) = y$
- $f(x,y) = x \wedge y$, $f(x,y) = x \vee y$

The monotonic Boolean functions are precisely those that can be defined by combining the inputs using only the operators $\wedge$ and $\vee$.

The Number of Antichains in the Powerset Lattice with n Generators

- powerset lattice $\mathbb{2}^X := (2^X, \subseteq),$ with $\#X = n$
- collection of subsets $\Sigma \subseteq 2^X$:

$$\forall \sigma, \tau \in \Sigma : \ \sigma \not\subseteq \tau \text{ and } \tau \not\subseteq \sigma$$

Example $X = \{x, y\}$:

$$\emptyset, \{\{x, y\}\}, \{\{x\}\}, \{\{y\}\}, \{\{x\}, \{y\}\}, \{\emptyset\}$$

Antichains are elements of the space $\mathbb{2}^{2^X}$.

The Number of Antichains in the Powerset Lattice with n Generators

- powerset lattice $\mathfrak{2}^X := (2^X, \subseteq)$, with $\#X = n$
- collection of subsets $\Sigma \subseteq 2^X$:

$$\forall \sigma, \tau \in \Sigma : \ \sigma \not\subseteq \tau \text{ and } \tau \not\subseteq \sigma$$

Example $X = \{x, y\}$:

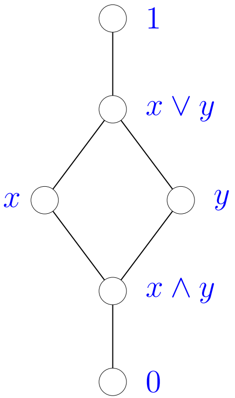
$$\emptyset, \{\{x, y\}\}, \{\{x\}\}, \{\{y\}\}, \{\{x\}, \{y\}\}, \{\emptyset\}$$

Antichains are elements of the space $\mathfrak{2}^{2^X}$.

The Number of Elements of the Free Distributive Lattice with n Generators

Let $\mathbb{D}_n = (\mathrm{D}(n), \leq)$ denote the *free distributive lattice* with n generators.

Example with generators x and y:



Dedekind's Problem

The determination of $\mathbb{D}_n$'s cardinality is known as *Dedekind's Problem*.

Berman, J., Köhler, P.: (1976):

"... probably the oldest unsolved problem in lattice theory."

Kisielewicz Andrzej (1988):

Provided an arithmetic formula for the Dedekind numbers in closed form, but it can only be applied to small values of n.

Dedekind's Problem

The determination of $\mathbb{D}_n$'s cardinality is known as *Dedekind's Problem*.

Berman, J., Köhler, P.: (1976):

"... probably the oldest unsolved problem in lattice theory."

Kisielewicz Andrzej (1988):

Provided an arithmetic formula for the Dedekind numbers in closed form, but it can only be applied to small values of n.

Dedekind's Problem

The determination of $\mathbb{D}_n$'s cardinality is known as *Dedekind's Problem*.

Berman, J., Köhler, P.: (1976):

"... probably the oldest unsolved problem in lattice theory."

Kisielewicz Andrzej (1988):

Provided an arithmetic formula for the Dedekind numbers in closed form, but it can only be applied to small values of n.

Dedekind Numbers

Let $\mathrm{d}(n) := \#\mathbb{D}_n$ denote the n-th Dedekind number.

n	$\mathrm{d}(n)$	Year
0	2	
1	3	
2	6	1897, Dedekind
3	20	
4	168	
5	7581	1940, Church
6	7828354	1946, Ward
7	2414682040998	1965, Church
8	56130437228687557907788	1991, Wiedemann

$$2023: \; d(9) = 286386577668298411128469151667598498812366$$

Table of Contents

Shifting Theorem

- let $\mathbb{2}^n := (2^n, \subseteq)$ denote the powerset lattice with n generators
- it holds that $\mathbb{D}_{n+k} \cong \mathbb{D}_n^{\mathbb{2}^k}$

Theorem

The Dedekind number $\mathrm{d}(n+k)$ is equal to the number of monotone mappings from $\mathbb{2}^k$ into $\mathbb{D}_n$.

Shifting Theorem

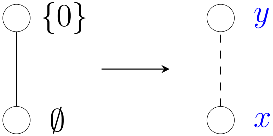
- let $\mathbb{2}^n := (2^n, \subseteq)$ denote the powerset lattice with n generators
- it holds that $\mathbb{D}_{n+k} \cong \mathbb{D}_n^{\mathbb{2}^k}$

Theorem
The Dedekind number $\mathrm{d}(n+k)$ *is equal to the number of monotone mappings from* $\mathbb{2}^k$ *into* $\mathbb{D}_n$.

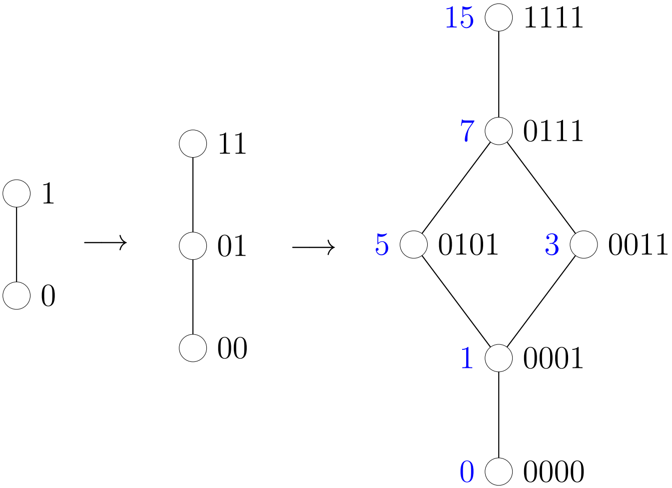
Generation and Numerical Representation

Corollary

There is a one to one correspondence of elements from $\mathbb{D}_{n+1}$ and pairs (x, y) of elements x, y from $\mathbb{D}_n$, such that $x \leq y$.

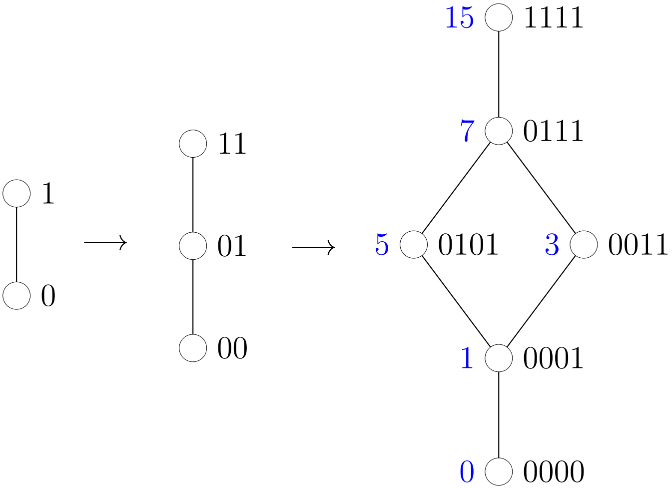


Generation and Numerical Representation



The integer values define a linear order, denoted by $\sqsubseteq$.

Generation and Numerical Representation



The integer values define a linear order, denoted by $\sqsubseteq$.

Equivalent Lattices

- finite lattice $\mathbb{L} = (L, \vee, \wedge, \bot, \top)$ or $\mathbb{L} = (L, \leq)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(x) \leq \varphi(y)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is anti isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(y) \leq \varphi(x)$

Definition

Two lattices $\mathbb{L}_1$ and $\mathbb{L}_2$ are *equivalent* iff they are isomorphic or anti isomorphic. We write $\mathbb{L}_1 \equiv \mathbb{L}_2$. It holds that $\equiv$ is an equivalence relation.

Equivalent Lattices

- finite lattice $\mathbb{L} = (L, \vee, \wedge, \bot, \top)$ or $\mathbb{L} = (L, \leq)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(x) \leq \varphi(y)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is anti isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(y) \leq \varphi(x)$

Definition

Two lattices $\mathbb{L}_1$ and $\mathbb{L}_2$ are *equivalent* iff they are isomorphic or anti isomorphic. We write $\mathbb{L}_1 \equiv \mathbb{L}_2$. It holds that $\equiv$ is an equivalence relation.

Equivalent Lattices

- finite lattice $\mathbb{L} = (L, \vee, \wedge, \bot, \top)$ or $\mathbb{L} = (L, \leq)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(x) \leq \varphi(y)$
- $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is anti isomorphism $:\Leftrightarrow x \leq y \Leftrightarrow \varphi(y) \leq \varphi(x)$

Definition

Two lattices $\mathbb{L}_1$ and $\mathbb{L}_2$ are *equivalent* iff they are isomorphic or anti isomorphic. We write $\mathbb{L}_1 \equiv \mathbb{L}_2$. It holds that $\equiv$ is an equivalence relation.

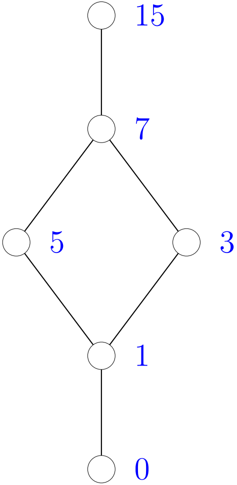
Equivalent Intervals

- for $a, b \in L$, $a \leq b$: *interval* $[a, b] := \{x \mid x \in L,\ a \leq x \leq b\}$
- set of all intervals $\mathrm{Int}(\mathbb{L})$
- $I, J \in \mathrm{Int}(\mathbb{L})$ are equivalent iff they are (anti) isomorphic
- factorization $\mathrm{Int}(\mathbb{L})/_{\equiv}$

Equivalent Intervals

- for $a, b \in L$, $a \leq b$: *interval* $[a, b] := \{x \mid x \in L,\ a \leq x \leq b\}$
- set of all intervals $\mathrm{Int}(\mathbb{L})$
- $I, J \in \mathrm{Int}(\mathbb{L})$ are equivalent iff they are (anti) isomorphic
- factorization $\mathrm{Int}(\mathbb{L})/_{\equiv}$

Equivalent Intervals of $\mathbb{D}_2$



$\{[0,0],[1,1],[3,3],[5,5],[7,7],[15,15]\}$,

$\{[0,1],[1,3],[1,5],[3,7],[5,7],[7,15]\}$,

$\{[0,3],[0,5],[3,15],[5,15]\}$,

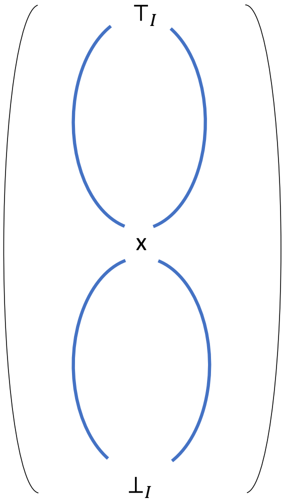
$\{[1,7]\}$,

$\{[0,7],[1,15]\}$,

$\{[0,15]\}$

Notation

I:



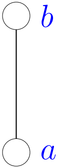
For $I \in \mathrm{Int}(\mathbb{L})$ and every $x \in I$, let:

$$\perp_{\mathrm{I}}(x) := \#[\perp_{\mathrm{I}}, x],$$

$$\top_{\mathrm{I}}(x) := \#[x, \top_{\mathrm{I}}].$$

Subscript I can be omitted.

Enumeration of $\mathbb{D}_{n+1}$, via $\mathbb{2}^1 \xrightarrow{\leq} \mathbb{D}_n$



$$\mathrm{d}(n+1) = \#\,\mathrm{Int}(\mathbb{D}_n)$$

$$= \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_\equiv} \#[I]$$

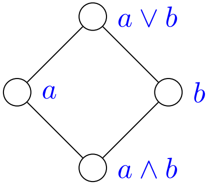
Enumeration of $\mathbb{D}_{n+1}$, via $\mathbb{2}^1 \xrightarrow{\leq} \mathbb{D}_n$

$$\mathrm{d}(n+1) = \#\,\mathrm{Int}(\mathbb{D}_n)$$

$$= \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}} \#[I]$$

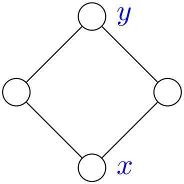
Enumeration of $\mathbb{D}_{n+2}$, via $\mathbb{2}^2 \xrightarrow{\leq} \mathbb{D}_n$

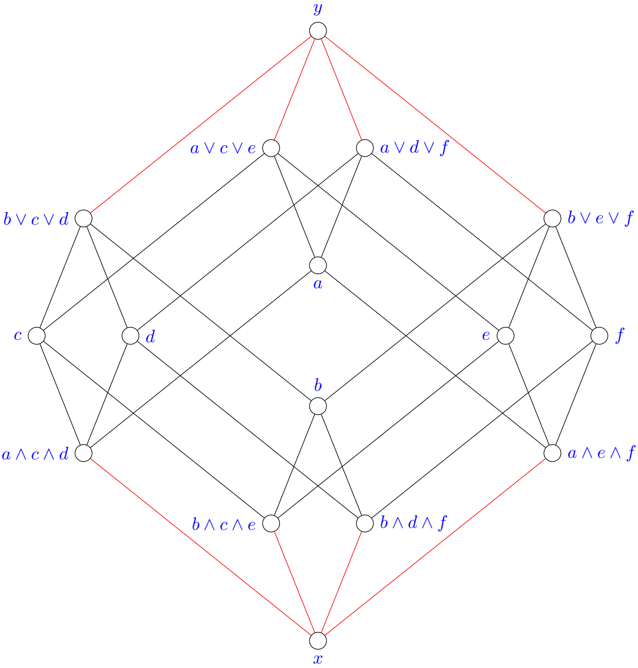
$$\mathrm{d}(n+2) = \sum_{a,b \in \mathrm{D}(n)} \bot(a \wedge b) \cdot \top(a \vee b)$$



Enumeration of $\mathbb{D}_{n+2}$, via $2^2 \xrightarrow{\leq} \mathbb{D}_n$

$$\mathrm{d}(n+2) = \sum_{I \in \mathrm{Int}(\mathbb{D}_n)} (\#I)^2 = \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}} (\#I)^2 \cdot \#[I]$$




y
$a \vee c \vee e$
$a \vee d \vee f$
$b \vee c \vee d$
$b \vee e \vee f$
a
c
d
e
f
b
$a \wedge c \wedge d$
$a \wedge e \wedge f$
$b \wedge c \wedge e$
$b \wedge d \wedge f$
x

Enumeration of $\mathbb{D}_{n+4}$, via $\mathbb{2}^4 \xrightarrow{\leq} \mathbb{D}_n$

$$\mathrm{d}(n+4) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_\equiv} \#[I] \cdot \sum_{a,b,c,d,e,f \in I} X, \text{ with:}$$

$$\begin{aligned} X = {} & \bot(a \wedge c \wedge d) \cdot \top(b \vee c \vee d) \\ & \cdot \bot(b \wedge c \wedge e) \cdot \top(a \vee c \vee e) \\ & \cdot \bot(a \wedge e \wedge f) \cdot \top(b \vee e \vee f) \\ & \cdot \bot(b \wedge d \wedge f) \cdot \top(a \vee d \vee f). \end{aligned}$$

Enumeration of $\mathbb{D}_{n+4}$, via $\mathbb{2}^4 \xrightarrow{\leq} \mathbb{D}_n$

Theorem

For $I \in \mathrm{Int}(\mathbb{D}_n)$ *and* $a, b, c, d \in I$, *we define matrices:*

$$\begin{aligned}
\alpha_{ab}(c,d) &:= \bot(a \wedge c \wedge d) \cdot \top(b \vee c \vee d), \\
\beta_{ab}(c,d) &:= \bot(b \wedge c \wedge d) \cdot \top(a \vee c \vee d), \\
\gamma_{ab} &:= \alpha_{ab} \cdot \beta_{ab}.
\end{aligned}$$

It holds that:

$$\mathrm{d}(n+4) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}} \#[I] \cdot \sum_{a,b \in I} \mathrm{Tr}(\gamma_{ab}^2).$$

Proof.

$$d(n+4) =$$

$$\sum_{[I]\in \mathrm{Int}(\mathbb{D}_n)/_\equiv} \#[I]\cdot \sum_{a,b\in I}\sum_{c,d\in I}\sum_{e,f\in I} \alpha_{ab}(c,d)\beta_{ab}(c,e)\alpha_{ab}(e,f)\beta_{ab}(d,f)$$

$\gamma_{ab}(d,e) = \sum_{c\in I} \alpha_{ab}(d,c)\beta_{ab}(c,e)$ and $\gamma_{ab}(e,d) = \sum_{f\in I} \alpha_{ab}(e,f)\beta_{ab}(f,d)$

$$\sum_{d,e\in I} \gamma_{ab}(d,e)\gamma_{ab}(e,d) = \sum_{d\in I} \gamma^2_{ab}(d,d) = \mathrm{Tr}(\gamma^2_{ab})$$

$\square$

Proof.

$$d(n+4) =$$

$$\sum_{[I]\in \mathrm{Int}(\mathbb{D}_n)/_\equiv} \#[I]\cdot \sum_{a,b\in I} \sum_{c,d\in I} \sum_{e,f\in I} \alpha_{ab}(c,d)\beta_{ab}(c,e)\alpha_{ab}(e,f)\beta_{ab}(d,f)$$

$\gamma_{ab}(d,e) = \sum_{c\in I} \alpha_{ab}(d,c)\beta_{ab}(c,e)$ and $\gamma_{ab}(e,d) = \sum_{f\in I} \alpha_{ab}(e,f)\beta_{ab}(f,d)$

$$\sum_{d,e\in I} \gamma_{ab}(d,e)\gamma_{ab}(e,d) = \sum_{d\in I} \gamma_{ab}^2(d,d) = \mathrm{Tr}(\gamma_{ab}^2)$$

□

Proof.

$$d(n+4) =$$

$$\sum_{[I]\in \mathrm{Int}(\mathbb{D}_n)/_\equiv} \#[I] \cdot \sum_{a,b\in I} \sum_{c,d\in I} \sum_{e,f\in I} \alpha_{ab}(c,d)\beta_{ab}(c,e)\alpha_{ab}(e,f)\beta_{ab}(d,f)$$

$\gamma_{ab}(d,e) = \sum_{c\in I} \alpha_{ab}(d,c)\beta_{ab}(c,e)$ and $\gamma_{ab}(e,d) = \sum_{f\in I} \alpha_{ab}(e,f)\beta_{ab}(f,d)$

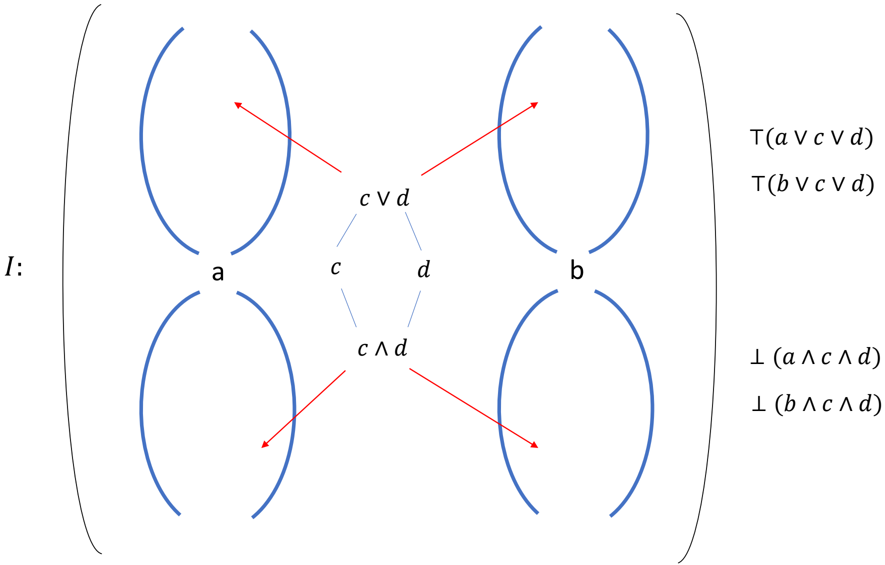
$$\sum_{d,e\in I} \gamma_{ab}(d,e)\gamma_{ab}(e,d) = \sum_{d\in I} \gamma_{ab}^2(d,d) = \mathrm{Tr}(\gamma_{ab}^2)$$

□

Matrix Multiplication

- standard algorithmic problem
- a lot of specialized libraries
- high performance on GPUs can be achieved
- to compute $\mathrm{d}(5+4)$, maximal matrix dimension is 7581×7581

Symmetry w.r.t. a and b



Symmetry w.r.t. a and b

$$(a,b) \in I \times I\,|_{\sqsubseteq}: \quad \omega(a,b) := \begin{cases} 1, & a = b \\ 2, & a \sqsubset b \end{cases}$$

Let $\varphi : I \to I$ be an (anti) isomorphism and $(a,b), (\tilde{a}, \tilde{b}) \in I \times I\,|_{\sqsubseteq}$:

$(a,b) \sim (\tilde{a}, \tilde{b}) :\Longleftrightarrow$

$$\exists \varphi : I \to I,\ \varphi(a) = \tilde{a} \text{ and } \varphi(b) = \tilde{b}.$$

Symmetry w.r.t. a and b

$$(a,b) \in I \times I\,|_{\sqsubseteq}: \quad \omega(a,b) := \begin{cases} 1, & a = b \\ 2, & a \sqsubset b \end{cases}$$

Let $\varphi : I \to I$ be an (anti) isomorphism and $(a,b), (\tilde{a},\tilde{b}) \in I \times I\,|_{\sqsubseteq}$:

$(a,b) \sim (\tilde{a},\tilde{b}) :\Longleftrightarrow$

$$\exists \varphi : I \to I,\ \varphi(a) = \tilde{a} \text{ and } \varphi(b) = \tilde{b}.$$

Enumeration of $\mathbb{D}_{n+4}$, via $2^4 \xrightarrow{\leq} \mathbb{D}_n$

Theorem

For $I \in \mathrm{Int}(\mathbb{D}_n)$ *and* $a, b, c, d \in I$, *we define matrices:*

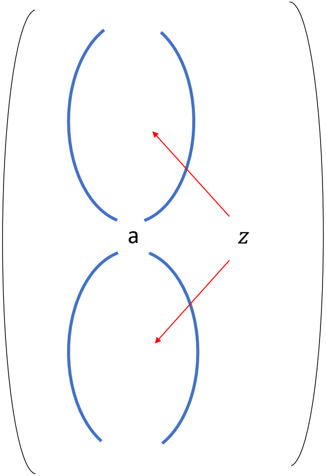
$$\begin{aligned} \alpha_{ab}(c,d) &:= \bot(a \wedge c \wedge d) \cdot \top(b \vee c \vee d), \\ \beta_{ab}(c,d) &:= \bot(b \wedge c \wedge d) \cdot \top(a \vee c \vee d), \\ \gamma_{ab} &:= \alpha_{ab} \cdot \beta_{ab}. \end{aligned}$$

It holds that $\mathrm{d}(n+4) =$

$$\sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}} \#[I] \cdot \sum_{[(a,b)] \in (I \times I|_{\sqsubseteq})/_{\sim}} \omega(a,b) \cdot \#[(a,b)] \cdot \mathrm{Tr}(\gamma_{ab}^2).$$

Proof.

I:



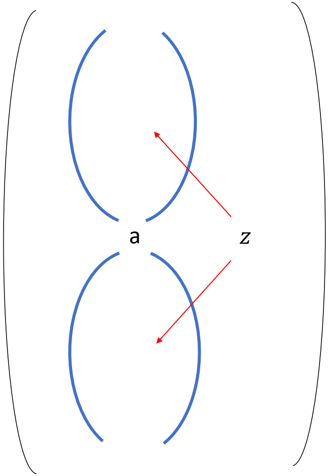
$$\sum_{z \in I} \top(a \vee z) \perp (a \wedge z)$$

$$\sum_{z \in I} \top(\varphi(a \vee z)) \perp (\varphi(a \wedge z))$$

$$\sum_{\tilde{z} \in I} \top(\tilde{a} \vee \tilde{z}) \perp (\tilde{a} \wedge \tilde{z})$$

Proof.

I:

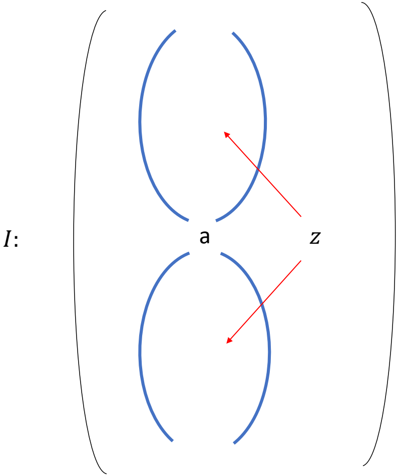


$$\sum_{z \in I} \top(a \vee z) \bot(a \wedge z)$$

$$\sum_{z \in I} \top(\varphi(a \vee z)) \bot(\varphi(a \wedge z))$$

$$\sum_{\tilde{z} \in I} \top(\tilde{a} \vee \tilde{z}) \bot(\tilde{a} \wedge \tilde{z})$$

Proof.



$$\sum_{z \in I} \top(a \vee z) \bot(a \wedge z)$$

$$\sum_{z \in I} \top(\varphi(a \vee z)) \bot(\varphi(a \wedge z))$$

$$\sum_{\tilde{z} \in I} \top(\tilde{a} \vee \tilde{z}) \bot(\tilde{a} \wedge \tilde{z})$$

Computation of $\mathrm{d}(n+4)$

1. generate elements and intervals of $\mathbb{D}_n$
2. compute equivalence classes $\mathrm{Int}(\mathbb{D}_n)/_\equiv$, save one representative for each class, save the cardinality of each class
3. for every equivalence class representative I of $\mathrm{Int}(\mathbb{D}_n)/_\equiv$, compute equivalence classes of $(I \times I \mid_\sqsubseteq)/_\sim$, save one representative and the cardinality
4. run the computation according to the last theorem

Computation of $\mathrm{d}(n+4)$

1. generate elements and intervals of $\mathbb{D}_n$
2. compute equivalence classes $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, save one representative for each class, save the cardinality of each class
3. for every equivalence class representative I of $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, compute equivalence classes of $(I \times I \mid_{\sqsubseteq})/_{\sim}$, save one representative and the cardinality
4. run the computation according to the last theorem

Computation of $\mathrm{d}(n+4)$

1. generate elements and intervals of $\mathbb{D}_n$
2. compute equivalence classes $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, save one representative for each class, save the cardinality of each class
3. for every equivalence class representative I of $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, compute equivalence classes of $(I \times I \mid_{\sqsubseteq})/_{\sim}$, save one representative and the cardinality
4. run the computation according to the last theorem

Computation of $\mathrm{d}(n+4)$

1. generate elements and intervals of $\mathbb{D}_n$
2. compute equivalence classes $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, save one representative for each class, save the cardinality of each class
3. for every equivalence class representative I of $\mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, compute equivalence classes of $(I \times I\ |_{\sqsubseteq})/_{\sim}$, save one representative and the cardinality
4. run the computation according to the last theorem

Lattice → Formal Context

- consider a finite lattice $\mathbb{L} = (L, \leq)$
- *join irreducibles* $J(\mathbb{L})$ and *meet irreducibles* $M(\mathbb{L})$
- *formal context* $\mathbb{K} := (J(\mathbb{L}), M(\mathbb{L}), R)$ with $R := \leq|_{J(\mathbb{L}) \times M(\mathbb{L})}$

- $\mathbb{K}^d := (M, J, R^d)$ *dual context* to $\mathbb{K} = (J, M, R)$
- $\mathbb{K}_1 = (J_1, M_1, R_1)$, $\mathbb{K}_2 = (J_2, M_2, R_2)$, *context isomorphism* $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$, with $jR_1m \Leftrightarrow \alpha(j)R_2\beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2^d$ exists

Lattice → Formal Context

- consider a finite lattice $\mathbb{L} = (L, \leq)$
- *join irreducibles* $J(\mathbb{L})$ and *meet irreducibles* $M(\mathbb{L})$
- *formal context* $\mathbb{K} := (J(\mathbb{L}), M(\mathbb{L}), R)$ with $R := \leq|_{J(\mathbb{L}) \times M(\mathbb{L})}$

- $\mathbb{K}^d := (M, J, R^d)$ *dual context* to $\mathbb{K} = (J, M, R)$
- $\mathbb{K}_1 = (J_1, M_1, R_1)$, $\mathbb{K}_2 = (J_2, M_2, R_2)$, *context isomorphism* $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$, with $jR_1m \Leftrightarrow \alpha(j)R_2\beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2^d$ exists

Lattice $\to$ Formal Context

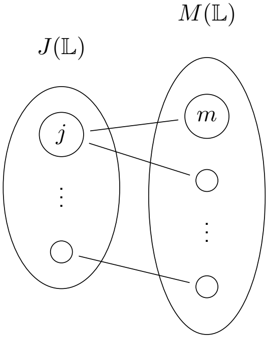
- consider a finite lattice $\mathbb{L} = (L, \leq)$
- *join irreducibles* $J(\mathbb{L})$ and *meet irreducibles* $M(\mathbb{L})$
- *formal context* $\mathbb{K} := (J(\mathbb{L}), M(\mathbb{L}), R)$ with $R := \leq|_{J(\mathbb{L}) \times M(\mathbb{L})}$

- $\mathbb{K}^d := (M, J, R^d)$ *dual context* to $\mathbb{K} = (J, M, R)$
- $\mathbb{K}_1 = (J_1, M_1, R_1)$, $\mathbb{K}_2 = (J_2, M_2, R_2)$, *context isomorphism* $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$, with $jR_1m \Leftrightarrow \alpha(j)R_2\beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ exists $\Longleftrightarrow$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \to \mathbb{K}_2^d$ exists

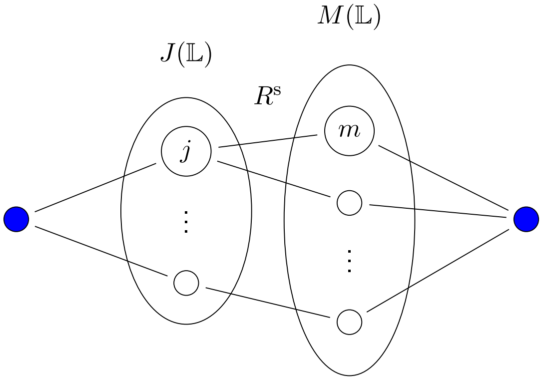
Formal Context $\to$ Bipartite Graph

$$\mathbb{K}^{\mathrm{s}} = (J^{\mathrm{s}}, M^{\mathrm{s}}, R^{\mathrm{s}}) := \mathbb{K}\dot{\cup}\mathbb{K}^d := (J\dot{\cup}M, J\dot{\cup}M, R\dot{\cup}R^d)$$

$R\dot{\cup}R^d$	J	M
J	$\emptyset$	R
M	R^d	$\emptyset$



Context (Anti) Isomorphism $\to$ Graph Isomorphism



Equivalent Lattices

- given: $\mathcal{L} = \{\mathbb{L} \mid \text{finite lattice } \mathbb{L}\}$
- task: compute $\mathcal{L}/_{\equiv}$
- special case: $\operatorname{Int}(\mathbb{L})$ defines set of sublattices

- $\mathcal{K} = \{\mathbb{K} \mid \text{formal context } \mathbb{K}\}$
- task: compute equivalence classes w.r.t. context (anti) isomorphisms

- $\mathcal{K}^{\mathrm{s}} = \{\mathbb{K}^{\mathrm{s}} \mid \text{symmetrization of formal context } \mathbb{K}^{\mathrm{s}}\}$
- task: compute equivalence classes w.r.t. graph isomorphisms

Equivalent Lattices

- given: $\mathcal{L} = \{\mathbb{L} \mid \text{finite lattice } \mathbb{L}\}$
- task: compute $\mathcal{L}/_{\equiv}$
- special case: $\mathrm{Int}(\mathbb{L})$ defines set of sublattices

- $\mathcal{K} = \{\mathbb{K} \mid \text{formal context } \mathbb{K}\}$
- task: compute equivalence classes w.r.t. context (anti) isomorphisms

- $\mathcal{K}^{\mathrm{s}} = \{\mathbb{K}^{\mathrm{s}} \mid \text{symmetrization of formal context } \mathbb{K}^{\mathrm{s}}\}$
- task: compute equivalence classes w.r.t. graph isomorphisms

Equivalent Lattices

- given: $\mathcal{L} = \{\mathbb{L} \mid \text{finite lattice } \mathbb{L}\}$
- task: compute $\mathcal{L}/_{\equiv}$
- special case: $\mathrm{Int}(\mathbb{L})$ defines set of sublattices

- $\mathcal{K} = \{\mathbb{K} \mid \text{formal context } \mathbb{K}\}$
- task: compute equivalence classes w.r.t. context (anti) isomorphisms

- $\mathcal{K}^{\mathrm{s}} = \{\mathbb{K}^{\mathrm{s}} \mid \text{symmetrization of formal context } \mathbb{K}^{\mathrm{s}}\}$
- task: compute equivalence classes w.r.t. graph isomorphisms

Canonical Labeling / Graph Canonization

- canonical form $\mathrm{Canon}(\mathbb{G})$, of a graph $\mathbb{G}$, is a labeled graph that is isomorphic to $\mathbb{G}$
- $\mathbb{G}_1 \cong \mathbb{G}_2 \iff \mathrm{Canon}(\mathbb{G}_1) = \mathrm{Canon}(\mathbb{G}_2)$
- complexity of determining isomorphism classes grows linearly with the number of graphs

- software nauty (No AUTomorphisms, Yes?), by Brendan McKay, can compute a canonical string of a given colored graph
- for example: $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3 \rightarrow$ ":DgXI@G ", ":DgWCgCb", ":DgXI@G "

Canonical Labeling / Graph Canonization

- canonical form $\mathrm{Canon}(\mathbb{G})$, of a graph $\mathbb{G}$, is a labeled graph that is isomorphic to $\mathbb{G}$
- $\mathbb{G}_1 \cong \mathbb{G}_2 \iff \mathrm{Canon}(\mathbb{G}_1) = \mathrm{Canon}(\mathbb{G}_2)$
- complexity of determining isomorphism classes grows linearly with the number of graphs

- software nauty (No AUTomorphisms, Yes?), by Brendan McKay, can compute a canonical string of a given colored graph
- for example: $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3 \rightarrow$ ":DgXI@G ", ":DgWCgCb", ":DgXI@G "

Algorithm to Compute Equivalent Intervals

Data: $\mathrm{Int}(\mathbb{D}_\mathbb{n})$
Result: $\mathrm{Int}(\mathbb{D}_\mathbb{n})/_\equiv$
for $[x, y] \in \mathrm{Int}(\mathbb{D}_\mathbb{n})$ **do**
- compute a formal context that represents $[x, y]$;
- transform the context to a colored bipartite graph;
- compute a canonical string with "nauty";

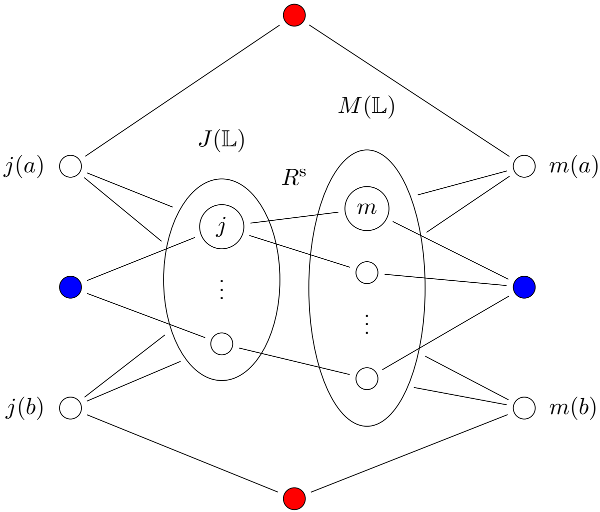
end
- count occurrences of each string;

Number of Equivalent Intervals

n	$\# \mathrm{Int}(\mathbb{D}_{\mathbb{n}})$	$\# \mathrm{Int}(\mathbb{D}_{\mathbb{n}})/_{\equiv}$	reduction
2	20	6	30%
3	168	18	10%
4	7581	134	1.77%
5	7828354	9919	0.13%
6	2414682040998	175396936	0.0073%

Equivalent (a, b) Values from $I \times I$

- for every $[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}$, compute the the bipartite colored graph as before
- iterate over every $(a, b) \in (I \times I) \mid_{\sqsubseteq}$
- extend the bipartite colored graph with data about a and b
- compute a canonical string with "nauty"
- count occurrences of each string



$M(\mathbb{L})$
$J(\mathbb{L})$
$j(a)$
$m(a)$
R^{s}
m
j
$j(b)$
$m(b)$

n	pairs treated	equivalence classes
2	56	33
3	1127	446
4	274409	80741
5	8646896880	4257682565

Largest interval of $\mathbb{D}_5$ is $[0, 4294967295]$. There, we get a reduction $57471561 \to 140736$, or 34 days to 2 hours on an A100 GPU.

Algorithm to Compute $\mathrm{d}(n+4)$

Data: $\mathrm{Int}(\mathbb{D}_{\mathrm{n}})/_{\equiv}$ and $\forall [I] \in \mathrm{Int}(\mathbb{D}_{\mathrm{n}})/_{\equiv} : \ (I \times I \mid_{\sqsubseteq})/_{\sim}$
Result: $\mathrm{d}(n+4)$
for $[I] \in \mathrm{Int}(\mathbb{D}_{\mathrm{n}})/_{\equiv}$ **do**
  for $[(a,b)] \in (I \times I \mid_{\sqsubseteq})/_{\sim}$ **do**
    - generate the matrices α_{ab} and β_{ab};
    - compute the matrix product $\gamma_{ab} = \alpha_{ab} \cdot \beta_{ab}$;
    - compute the trace of γ_{ab}^2;
    - multiply the trace with $\omega(a,b)$ and $\#[(a,b)]$;
  end
  - sum up each value from above;
  - multiply the sum with $\#[I]$;
end

The Computation

- $\perp(\cdot)$ and $\top(\cdot)$ are computed by the CPU host and transferred to a GPU device
- matrix generation, matrix product and trace computation are done on a GPU device
- Nvidia CUDA's "cublasDgemmStridedBatched" kernel is used to multiply a batch of matrices
- sanity check GPU vs. CPU
- estimating maximal values for matrix multiplication and trace computation

- $d(8)$ in about 3s on a A10/A100, or 9s on a Nvidia Quadro M2200, or 8s on Intel Core i7-7920HQ single thread
- $d(9)$ took 5311 A100 hours, or 27.6 days real time

The Computation

- $\perp(\cdot)$ and $\top(\cdot)$ are computed by the CPU host and transferred to a GPU device
- matrix generation, matrix product and trace computation are done on a GPU device
- Nvidia CUDA's "cublasDgemmStridedBatched" kernel is used to multiply a batch of matrices
- sanity check GPU vs. CPU
- estimating maximal values for matrix multiplication and trace computation

- $\mathrm{d}(8)$ in about 3s on a A10/A100, or 9s on a Nvidia Quadro M2200, or 8s on Intel Core i7-7920HQ single thread
- $\mathrm{d}(9)$ took 5311 A100 hours, or 27.6 days real time

The Computation

- $\perp(\cdot)$ and $\top(\cdot)$ are computed by the CPU host and transferred to a GPU device
- matrix generation, matrix product and trace computation are done on a GPU device
- Nvidia CUDA's "cublasDgemmStridedBatched" kernel is used to multiply a batch of matrices
- sanity check GPU vs. CPU
- estimating maximal values for matrix multiplication and trace computation

- $d(8)$ in about 3s on a A10/A100, or 9s on a Nvidia Quadro M2200, or 8s on Intel Core i7-7920HQ single thread
- $d(9)$ took 5311 A100 hours, or 27.6 days real time

The Computation

- $\perp(\cdot)$ and $\top(\cdot)$ are computed by the CPU host and transferred to a GPU device
- matrix generation, matrix product and trace computation are done on a GPU device
- Nvidia CUDA's "cublasDgemmStridedBatched" kernel is used to multiply a batch of matrices
- sanity check GPU vs. CPU
- estimating maximal values for matrix multiplication and trace computation

- $\mathrm{d}(8)$ in about 3s on a A10/A100, or 9s on a Nvidia Quadro $\mathrm{M}2200$, or 8s on Intel Core i7-7920HQ single thread
- $\mathrm{d}(9)$ took 5311 A100 hours, or 27.6 days real time

Confirmation

Lennart Van Hirtum, Patrick De Causmaecker, Jens Goemaere, Tobias Kenter, Heinrich Riebler, Michael Lass and Christian Plessl:

A computation of D(9) using FPGA Supercomputing.

$$\mathrm{d}(n+2) = \sum_{a,b \in D_n} \perp(a) \cdot 2^{\#C_{a,b}} \cdot \top(b)$$

It took 47000 FPGA hours, or about 3 month real time.

Confirmation

Lennart Van Hirtum, Patrick De Causmaecker, Jens Goemaere, Tobias Kenter, Heinrich Riebler, Michael Lass and Christian Plessl:

A computation of D(9) using FPGA Supercomputing.

$$\mathrm{d}(n+2) = \sum_{a,b \in D_n} \perp(a) \cdot 2^{\#C_{a,b}} \cdot \top(b)$$

It took 47000 FPGA hours, or about 3 month real time.

Thank you for your attention!

$$286386577668298411128469151667598498812366$$