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

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

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

Example $n = 2$:

- $f(x, y) = 0$, $f(x, y) = 1$
- $f(x, y) = x$, $f(x, y) = y$
- $f(x, y) = x \land y$, $f(x, y) = x \lor y$

The monotonic Boolean functions are precisely those that can be defined by combining the inputs using only the operators $\land$ and $\lor$. 
The Number of Monotone Boolean Functions on $n$ Variables

\[ f : \{0, 1\}^n \rightarrow \{0, 1\}, \text{ monotone w.r.t. } 0 \leq 1 \]

Example $n = 2$:

- $f(x, y) = 0, \ f(x, y) = 1$
- $f(x, y) = x, \ f(x, y) = y$
- $f(x, y) = x \land y, \ f(x, y) = x \lor y$

The monotonic Boolean functions are precisely those that can be defined by combining the inputs using only the operators $\land$ and $\lor$. 
The Number of Antichains in the Powerset Lattice with \( n \) Generators

- powerset lattice \( \mathcal{P}^X := (2^X, \subseteq) \), 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 \( \mathcal{P}^{2^X} \).
The Number of Antichains in the Powerset Lattice with \( n \) Generators

- powerset lattice \( \mathcal{P}^X := (\mathcal{P}^X, \subseteq) \), with \( \#X = n \)
- collection of subsets \( \Sigma \subseteq \mathcal{P}^X \):
  \[
  \forall \sigma, \tau \in \Sigma : \sigma \nsubseteq \tau \text{ and } \tau \nsubseteq \sigma
  \]

Example \( X = \{x, y\} \):

\[
\emptyset, \{\{x, y\}\}, \{\{x\}\}, \{\{y\}\}, \{\{x\}, \{y\}\}, \{\emptyset\}
\]

Antichains are elements of the space \( \mathcal{P}^{\mathcal{P}^X} \).
The Number of Elements of the Free Distributive Lattice with \( n \) Generators

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

Example with generators \( x \) and \( y \):

![Diagram of a distributive lattice 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 $d(n) := \#D_n$ denote the $n$-th Dedekind number.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$d(n)$</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1897, Dedekind</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>1940, Church</td>
</tr>
<tr>
<td>4</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7581</td>
<td>1946, Ward</td>
</tr>
<tr>
<td>6</td>
<td>7828354</td>
<td>1965, Church</td>
</tr>
<tr>
<td>7</td>
<td>2414682040998</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>56130437228687557907788</td>
<td>1991, Wiedemann</td>
</tr>
</tbody>
</table>

2023: $d(9) = 286386577668298411128469151667598498812366$
Shifting Theorem

• let $2^n := (2^n, \subseteq)$ denote the powerset lattice with $n$ generators
• it holds that $D_{n+k} \cong D_{2^k}$

Theorem

The Dedekind number $d(n + k)$ is equal to the number of monotone mappings from $2^k$ into $D_n$. 
Shifting Theorem

• let \(2^n := (2^n, \subseteq)\) denote the powerset lattice with \(n\) generators
• it holds that \(D_{n+k} \cong D_{2^k}\)

**Theorem**

The Dedekind number \(d(n + k)\) is equal to the number of monotone mappings from \(2^k\) into \(D_n\).
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 $x \leq y$. 

\[
\begin{array}{c}
\{0\} \\
\emptyset \\
x \\
y
\end{array}
\]
The integer values define a linear order, denoted by $\sqsubseteq$. 
The integer values define a linear order, denoted by $\sqsubseteq$. 

The image shows a diagram with nodes labeled with binary numbers, illustrating the generation and numerical representation. Each node represents a digit in a sequence, and the arrows indicate the order or transformation between these numbers. The diagram helps to visualize how the integers are ordered and how they can be represented numerically.
Equivalent Lattices

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

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 $\mathbb{I} = (L, \vee, \wedge, \bot, \top)$ or $\mathbb{I} = (L, \leq)$
- $\varphi : \mathbb{I}_1 \rightarrow \mathbb{I}_2$ is isomorphism $: \iff x \leq y \iff \varphi(x) \leq \varphi(y)$
- $\varphi : \mathbb{I}_1 \rightarrow \mathbb{I}_2$ is anti isomorphism $: \iff x \leq y \iff \varphi(y) \leq \varphi(x)$

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

• finite lattice $\mathbb{L} = (L, \lor, \land, \bot, \top)$ or $\mathbb{L} = (L, \leq)$

• $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is isomorphism $\iff x \leq y \iff \varphi(x) \leq \varphi(y)$

• $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ is anti isomorphism $\iff x \leq y \iff \varphi(y) \leq \varphi(x)$

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 $\text{Int}(\mathbb{L})$
- $I, J \in \text{Int}(\mathbb{L})$ are equivalent iff they are (anti) isomorphic
- factorization $\text{Int}(\mathbb{L})/\equiv$
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 $\text{Int}(\mathbb{L})$
- $I, J \in \text{Int}(\mathbb{L})$ are equivalent iff they are (anti) isomorphic
- factorization $\text{Int}(\mathbb{L})/\equiv$
Equivalent Intervals of $\mathbb{D}_2$

\[
\{[0, 0], [1, 1], [3, 3], [5, 5], [7, 7], [15, 15]\}, \\
\{[0, 1], [1, 3], [1, 5], [3, 7], [5, 7], [7, 15]\}, \\
\{[0, 3], [0, 5], [3, 15], [5, 15]\}, \\
\{[1, 7]\}, \\
\{[0, 7], [1, 15]\}, \\
\{[0, 15]\}
\]
For $I \in \text{Int}(\mathbb{L})$ and every $x \in I$, let:

$$\bot_I(x) := \#[\bot_I, x],$$

$$\top_I(x) := \#[x, \top_I].$$

Subscript $I$ can be omitted.
Enumeration of $D_{n+1}$, via $\mathcal{P}^1 \xrightarrow{\leq} D_n$

$$d(n + 1) = \# \text{Int}(D_n)$$

$$= \sum_{[I] \in \text{Int}(D_n) / \equiv} \#[I]$$
Enumeration of $\mathbb{D}_{n+1}$, via $\mathcal{P}^1 \xrightarrow{\leq} \mathbb{D}_n$

\[
d(n + 1) = \# \text{Int}(\mathbb{D}_n)
= \sum_{[I] \in \text{Int}(\mathbb{D}_n)/\equiv} \#[I]
\]
Enumeration of $\mathbb{D}_{n+2}$, via $2^2 \xrightarrow{\leq} \mathbb{D}_n$

$$d(n+2) = \sum_{a,b \in \mathbb{D}(n)} \bot (a \land b) \cdot \top (a \lor b)$$
Enumeration of $\mathbb{D}_{n+2}$, via $\mathbb{P}^2 \xrightarrow{\leq} \mathbb{D}_n$

\[ d(n + 2) = \sum_{I \in \text{Int}(\mathbb{D}_n)} (#I)^2 = \sum_{[I] \in \text{Int}(\mathbb{D}_n)/\equiv} (#I)^2 \cdot \#[I] \]
Enumeration of $\mathbb{D}_{n+4}$, via $\mathcal{P}^4 \xrightarrow{\leq} \mathbb{D}_n$

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

$$X = \bot (a \land c \land d) \land \top (b \lor c \lor d) \land \\
\cdot \bot (b \land c \land e) \land \top (a \lor c \lor e) \land \\
\cdot \bot (a \land e \land f) \land \top (b \lor e \lor f) \land \\
\cdot \bot (b \land d \land f) \land \top (a \lor d \lor f).$$
Enumeration of $\mathbb{D}_{n+4}$, via $\mathcal{P}^4 \xrightarrow{\leq} \mathbb{D}_n$

**Theorem**

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

\[
\alpha_{ab}(c, d) := \bot(a \land c \land d) \cdot \top(b \lor c \lor d), \\
\beta_{ab}(c, d) := \bot(b \land c \land d) \cdot \top(a \lor c \lor d), \\
\gamma_{ab} := \alpha_{ab} \cdot \beta_{ab}.
\]

*It holds that:*

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

\[ d(n + 4) = \]

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

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

\[ \sum_{d,e \in I} \gamma_{ab}(d, e) \gamma_{ab}(e, d) = \sum_{d \in I} \gamma_{ab}^2(d, d) = \text{Tr}(\gamma_{ab}^2) \]
Proof.

\[ d(n + 4) = \sum_{[I] \in \text{Int}(\mathbb{D}_n) / \equiv} \# [I] \cdot \sum_{a, b \in I} \sum_{c, d \in I} \sum_{e, f \in I} \alpha_{ab}(c, d) \beta_{ab}(c, e) \alpha_{ab}(e, f) \beta_{ab}(d, f) \]

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

\[ \sum_{d, e \in I} \gamma_{ab}(d, e) \gamma_{ab}(e, d) = \sum_{d \in I} \gamma_{ab}^2(d, d) = \text{Tr}(\gamma_{ab}^2) \]
Proof.

\[ d(n + 4) = \]

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

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

\[ \sum_{d, e \in I} \gamma_{ab}(d, e) \gamma_{ab}(e, d) = \sum_{d \in I} \gamma_{ab}^2(d, d) = \text{Tr} \left( \gamma_{ab}^2 \right) \]
Matrix Multiplication

- standard algorithmic problem
- a lot of specialized libraries
- high performance on GPUs can be achieved
- to compute $d(5 + 4)$, maximal matrix dimension is $7581 \times 7581$
Symmetry w.r.t. $a$ and $b$

$I$: $a \vdash b$

$\top (a \lor c \lor d)$
$\top (b \lor c \lor d)$

$\bot (a \land c \land d)$
$\bot (b \land c \land d)$
Symmetry w.r.t. $a$ and $b$

\[(a, b) \in I \times I \mid \sqsubseteq: \quad \omega(a, b) := \begin{cases} 1, & a = b \\ 2, & a \sqsubseteq b \end{cases}\]

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

\[(a, b) \sim (\tilde{a}, \tilde{b}) : \iff \exists \varphi : I \to I, \varphi(a) = \tilde{a} \text{ and } \varphi(b) = \tilde{b}.\]
Symmetry w.r.t. $a$ and $b$

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

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

$$(a, b) \sim (\tilde{a}, \tilde{b}) : \iff \exists \varphi : I \to I, \varphi(a) = \tilde{a} \text{ and } \varphi(b) = \tilde{b}.$$
Enumeration of \( \mathbb{D}_{n+4} \), via \( 2^4 \xrightarrow{\leq} \mathbb{D}_n \)

**Theorem**

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

\[
\alpha_{ab}(c, d) := \bot (a \land c \land d) \cdot \top (b \lor c \lor d),
\]
\[
\beta_{ab}(c, d) := \bot (b \land c \land d) \cdot \top (a \lor c \lor d),
\]
\[
\gamma_{ab} := \alpha_{ab} \cdot \beta_{ab}.
\]

It holds that \( d(n + 4) = \)

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

\[
\sum_{z \in I} T(a \lor z) \bot (a \land z)
\]

\[
\sum_{z \in I} T(\varphi(a \lor z)) \bot (\varphi(a \land z))
\]

\[
\sum_{\tilde{z} \in I} T(\tilde{a} \lor \tilde{z}) \bot (\tilde{a} \land \tilde{z})
\]
Proof.

\[
\sum_{z \in I} \top(a \vee z) \perp (a \land z)
\]

\[
\sum_{z \in I} \top(\varphi(a \vee z)) \perp (\varphi(a \land z))
\]

\[
\sum_{\tilde{z} \in I} \top(\tilde{a} \vee \tilde{z}) \perp (\tilde{a} \land \tilde{z})
\]
Proof.

\[
\sum_{z \in I} \top(a \lor z) \bot (a \land z)
\]

\[
\sum_{z \in I} \top(\phi(a \lor z)) \bot (\phi(a \land z))
\]

\[
\sum_{\tilde{z} \in I} \top(\tilde{a} \lor \tilde{z}) \bot (\tilde{a} \land \tilde{z})
\]
Computation of $d(n + 4)$

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

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

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

1. generate elements and intervals of $\mathbb{D}_n$
2. compute equivalence classes $\text{Int}(\mathbb{D}_n)/\equiv$, save one representative for each class, save the cardinality of each class
3. for every equivalence class representative $I$ of $\text{Int}(\mathbb{D}_n)/\equiv$, compute equivalence classes of $(I \times I |_{\equiv})/\sim$, save one representative and the cardinality
4. run the computation according to the last theorem
Lattice $\rightarrow$ 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 $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2$, with $j R_1 m \iff \alpha(j) R_2 \beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2^d$ exists
Lattice $\rightarrow$ 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 $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2,$ with $j R_1 m \Leftrightarrow \alpha(j) R_2 \beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2^d$ exists
Lattice $\rightarrow$ 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 $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2$, with $j R_1 m \Leftrightarrow \alpha(j) R_2 \beta(m)$
- lattice isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism: $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2$ exists
- lattice anti isomorphism $\varphi : \mathbb{L}_1 \rightarrow \mathbb{L}_2$ exists $\iff$ context isomorphism $(\alpha, \beta) : \mathbb{K}_1 \rightarrow \mathbb{K}_2^d$ exists
Formal Context $\rightarrow$ Bipartite Graph

$$K^s = (J^s, M^s, R^s) := K \cup K^d := (J \cup M, J \cup M, R \cup R^d)$$
Context (Anti) Isomorphism $\rightarrow$ Graph Isomorphism
Equivalent Lattices

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

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

- $\mathcal{K}^s = \{ K^s \mid \text{symmetrization of formal context } K^s \}$
- 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: $\text{Int}(\mathbb{L})$ defines set of sublattices

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

• $\mathcal{K}^s = \{ \mathbb{K}^s \mid \text{symmetrization of formal context } \mathbb{K}^s \}$
• 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: $\text{Int}(\mathbb{L})$ defines set of sublattices

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

- $\mathcal{K}^s = \{\mathbb{K}^s \mid \text{symmetrization of formal context } \mathbb{K}^s\}$
- task: compute equivalence classes w.r.t. graph isomorphisms
• canonical form $\text{Canon}(G)$, of a graph $G$, is a labeled graph that is isomorphic to $G$

• $G_1 \cong G_2 \iff \text{Canon}(G_1) = \text{Canon}(G_2)$

• 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: $G_1, G_2, G_3 \rightarrow ":\text{DgXI}@G ", ":\text{DgWCgCb}"$, $":\text{DgXI}@G "$
Canonical Labeling / Graph Canonization

- canonical form $\text{Canon}(G)$, of a graph $G$, is a labeled graph that is isomorphic to $G$
- $G_1 \cong G_2 \iff \text{Canon}(G_1) = \text{Canon}(G_2)$
- 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: $G_1, G_2, G_3 \rightarrow ":\text{DgXI}@G", ":\text{DgWCgCb}, ":\text{DgXI}@G "$
Algorithm to Compute Equivalent Intervals

**Data:** \( \text{Int}(\mathbb{D}_n) \)

**Result:** \( \text{Int}(\mathbb{D}_n)/\equiv \)

**for** \([x, y] \in \text{Int}(\mathbb{D}_n)\) **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

<table>
<thead>
<tr>
<th>$n$</th>
<th>$# \text{Int}(\mathbb{D}_n)$</th>
<th>$# \text{Int}(\mathbb{D}_n)/\equiv$</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>6</td>
<td>30%</td>
</tr>
<tr>
<td>3</td>
<td>168</td>
<td>18</td>
<td>10%</td>
</tr>
<tr>
<td>4</td>
<td>7581</td>
<td>134</td>
<td>1.77%</td>
</tr>
<tr>
<td>5</td>
<td>7828354</td>
<td>9919</td>
<td>0.13%</td>
</tr>
<tr>
<td>6</td>
<td>2414682040998</td>
<td>175396936</td>
<td>0.0073%</td>
</tr>
</tbody>
</table>
Equivalent \((a, b)\) Values from \(I \times I\)

- for every \([I] \in \text{Int}(\mathbb{D}_n)/\equiv\), compute the the bipartite colored graph as before
- iterate over every \((a, b) \in (I \times I) \mid \subseteq\)
- extend the bipartite colored graph with data about \(a\) and \(b\)
- compute a canonical string with "nauty"
- count occurrences of each string
Largest interval of $\mathbb{D}_5$ is $[0, 4294967295]$. There, we get a reduction $57471561 \rightarrow 140736$, or 34 days to 2 hours on an A100 GPU.

<table>
<thead>
<tr>
<th>$n$</th>
<th>pairs treated</th>
<th>equivalence classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>56</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>1127</td>
<td>446</td>
</tr>
<tr>
<td>4</td>
<td>274409</td>
<td>80741</td>
</tr>
<tr>
<td>5</td>
<td>8646896880</td>
<td>4257682565</td>
</tr>
</tbody>
</table>
Algorithm to Compute $d(n + 4)$

Data: $\text{Int}(\mathcal{D}_n)/\equiv$ and $\forall[I] \in \text{Int}(\mathcal{D}_n)/\equiv : (I \times I |\subseteq)/\sim$

Result: $d(n + 4)$

for $[I] \in \text{Int}(\mathcal{D}_n)/\equiv$ do
  for $[(a, b)] \in (I \times I |\subseteq)/\sim$ do
    - generate the matrices $\alpha_{ab}$ and $\beta_{ab}$;
    - compute the matrix product $\gamma_{ab} = \alpha_{ab} \cdot \beta_{ab}$;
    - compute the trace of $\gamma_{ab}^2$;
    - 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

- $\bot(\cdot)$ 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

• $\bot(\cdot)$ 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

- $\bot(\cdot)$ 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

- \( \bot(\cdot) \) 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
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.

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

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

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

It took 47000 FPGA hours, or about 3 month real time.
Thank you for your attention!

28638657766829841111284691516675984988812366