CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Computing Generalized Convolutions Faster Than Brute Force

IPEC 2022

Barış Can Esmer[1] Ariel Kulik[1] Dániel Marx[1]
Philipp Schepper[1] Karol Węgrzycki[2]

[1] CISPA Helmholtz Center for Information Security, Germany
[2] Saarland University and Max Planck Institute for Informatics, Saarbrücken, Germany

September 7, 2022

Motivation

Convolutions are used in parameterized and exact algorithms for

- Hamiltonian Cycle,
- Feedback Vertex Set,
- Steiner Tree, . . .

Motivation

Convolutions are used in parameterized and exact algorithms for

- Hamiltonian Cycle,
- Feedback Vertex Set,
- Steiner Tree, ...

Common application:
Improved computation of the join nodes for DPs on tree decompositions.

Motivation

Convolutions are used in parameterized and exact algorithms for

- Hamiltonian Cycle,
- Feedback Vertex Set,
- Steiner Tree, . . .

Common application:
Improved computation of the join nodes for DPs on tree decompositions.

Different variants of convolutions are studied:

- (Generalized) Subset Convolution
- Cover Product
- XOR Product, . . .

Motivation

Convolutions are used in parameterized and exact algorithms for

- Hamiltonian Cycle,
- Feedback Vertex Set,
- Steiner Tree, ...

Common application:
Improved computation of the join nodes for DPs on tree decompositions.

Different variants of convolutions are studied:

- (Generalized) Subset Convolution
- Cover Product
- XOR Product, ...

Difficulty:
Normally each problem needs a separate type of convolution.

Motivation

Convolutions are used in parameterized and exact algorithms for

- Hamiltonian Cycle,
- Feedback Vertex Set,
- Steiner Tree, ...

Common application:
Improved computation of the join nodes for DPs on tree decompositions.

Different variants of convolutions are studied:

- (Generalized) Subset Convolution
- Cover Product
- XOR Product, ...

Difficulty:
Normally each problem needs a separate type of convolution.

Goal:
Unify the convolution procedures under one umbrella.

Convolutions

Classical convolution:
For two functions $g, h\colon \mathbb{Z} \to \mathbb{Z}$ this is

$$(g * h)(c) \coloneqq \sum_{a} g(a) \cdot h(c - a) = \sum_{a+b=c} g(a) \cdot h(b) \qquad \forall\, c \in \mathbb{Z}.$$

Convolutions

Classical convolution:
For two functions $g, h\colon \mathbb{Z} \to \mathbb{Z}$ this is

$$(g * h)(c) := \sum_{a} g(a) \cdot h(c-a) = \sum_{a+b=c} g(a) \cdot h(b) \qquad \forall\, c \in \mathbb{Z}.$$

We *do not* study this!

Cover Product:

Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys.

Convolutions

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Definition (f-Convolution (van Rooij 2021))

Fix: A finite domain D and a function $f\colon D \times D \to D$.

Convolutions

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Definition (f-Convolution (van Rooij 2021))

Fix: A finite domain D and a function $f\colon D \times D \to D$.
In: Two functions $g, h\colon D^n \to \mathbb{Z}$.

Convolutions

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Definition (f-Convolution (van Rooij 2021))

Fix: A finite domain D and a function $f : D \times D \to D$.
In: Two functions $g, h\colon D^n \to \mathbb{Z}$.
Out: The f-Convolution of g and h, denoted by $(g \circledast_f h) : D^n \to \mathbb{Z}$,

Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Definition (f-Convolution (van Rooij 2021))

Fix: A finite domain D and a function $f\colon D \times D \to D$.
In: Two functions $g, h\colon D^n \to \mathbb{Z}$.
Out: The f-Convolution of g and h, denoted by $(g \circledast_f h) : D^n \to \mathbb{Z}$, defined as

$$(g \circledast_f h)(\mathbf{v}) \coloneqq \sum_{\substack{\mathbf{u},\mathbf{w} \in D^n \\ \text{s.t. } \mathbf{v} = f(\mathbf{u},\mathbf{w})}} g(\mathbf{u}) \cdot h(\mathbf{w}) \qquad \forall\, \mathbf{v} \in D^n.$$

Here, $f(\mathbf{u}, \mathbf{v})$ denotes the coordinate-wise application of f for two vectors $\mathbf{u}, \mathbf{v} \in D^n$.

Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Cover Product:
Let S be a set. For two functions $g, h\colon 2^S \to \mathbb{Z}$, the cover product is

$$(g *_{\mathrm{CP}} h)(C) \coloneqq \sum_{A \cup B = C} g(A) \cdot h(B) \qquad \forall\, C \subseteq S.$$

Conditions at the sums varys. $\implies$ Use a more general definition.

Definition (f-Convolution (van Rooij 2021))

Fix: A finite domain D and a function $f\colon D \times D \to D$.
In: Two functions $g, h\colon D^n \to \mathbb{Z}$.
Out: The f-Convolution of g and h, denoted by $(g \circledast_f h) : D^n \to \mathbb{Z}$, defined as

$$(g \circledast_f h)(\mathbf{v}) \coloneqq \sum_{\substack{\mathbf{u},\mathbf{w} \in D^n \\ \text{s.t. } \mathbf{v} = f(\mathbf{u},\mathbf{w})}} g(\mathbf{u}) \cdot h(\mathbf{w}) \qquad \forall\, \mathbf{v} \in D^n.$$

Here, $f(\mathbf{u}, \mathbf{v})$ denotes the coordinate-wise application of f for two vectors $\mathbf{u}, \mathbf{v} \in D^n$.
For Cover Product we set $D = \{0, 1\}$ and f as addition with maximum of 1.

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

Only for special cases faster algorithms are known.

Theorem (van Rooij 2021, Umans 2019 + Yates 1937)

f-CONVOLUTION can be solved

- in time $|D|^{n} \cdot n^{\mathcal{O}(1)}$ if f is addition (with maximum or modulo), or maximum, and
- in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ if f is a finite-group operation

  with $\omega < 2.373$ being the matrix-multiplication exponent.

Computing Generalized Convolutions

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

Only for special cases faster algorithms are known.

Theorem (van Rooij 2021, Umans 2019 + Yates 1937)

f-CONVOLUTION can be solved

- in time $|D|^{n} \cdot n^{\mathcal{O}(1)}$ if f is addition (with maximum or modulo), or maximum, and
- in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ if f is a finite-group operation
  with $\omega < 2.373$ being the matrix-multiplication exponent.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-CONVOLUTION can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f\colon D \times D \to D$.

Theorem (Brute Force Approach)

f-Convolution can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-Convolution can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f : D \times D \to D$.

Example for $|D| = 6$:

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-CONVOLUTION can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f\colon D \times D \to D$.

Example for $|D| = 6$:

- Naive algorithm: $|D|^{2n} \cdot n^{\mathcal{O}(1)} = 36^n \cdot n^{\mathcal{O}(1)}$

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-CONVOLUTION can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f \colon D \times D \to D$.

Example for $|D| = 6$:

- Naive algorithm: $|D|^{2n} \cdot n^{\mathcal{O}(1)} = 36^n \cdot n^{\mathcal{O}(1)}$
- Our result: $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)} = 30^n \cdot n^{\mathcal{O}(1)}$

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-Convolution can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-Convolution can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f\colon D \times D \to D$.

Example for $|D| = 6$:

- Naive algorithm: $|D|^{2n} \cdot n^{\mathcal{O}(1)} = 36^n \cdot n^{\mathcal{O}(1)}$
- Our result: $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)} = 30^n \cdot n^{\mathcal{O}(1)}$

Main idea: "Reduce" f-Convolution to addition with modulo.

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-CONVOLUTION can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-CONVOLUTION can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f\colon D \times D \to D$.

Example for $|D| = 6$:

- Naive algorithm: $|D|^{2n} \cdot n^{\mathcal{O}(1)} = 36^n \cdot n^{\mathcal{O}(1)}$
- Our result: $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)} = 30^n \cdot n^{\mathcal{O}(1)}$

Main idea: "Reduce" f-CONVOLUTION to addition with modulo.

1. Find a small cyclic partition of the function f.

Computing Generalized Convolutions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Brute Force Approach)

f-Convolution can be solved in time $|D|^{2n} \cdot n^{\mathcal{O}(1)}$ by a brute-force approach.

We improve the naive computation for *all* functions f.

Main Theorem (simplified)

f-Convolution can be solved in time $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)}$ for all $f\colon D \times D \to D$.

Example for $|D| = 6$:

- Naive algorithm: $|D|^{2n} \cdot n^{\mathcal{O}(1)} = 36^n \cdot n^{\mathcal{O}(1)}$
- Our result: $(\frac{5}{6}|D|^2)^n \cdot n^{\mathcal{O}(1)} = 30^n \cdot n^{\mathcal{O}(1)}$

Main idea: "Reduce" f-Convolution to addition with modulo.

1. Find a small cyclic partition of the function f.
2. Give a general algorithm based on cyclic partitions to compute the convolution.

Cyclic Minors

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition (Cyclic Minor)

For $A, B \subseteq D$ and $k \in \mathbb{N}$, (A, B, k) is a *cyclic minor* of f if the restriction of f to A and B is addition modulo k, after relabeling the sets A, B, and D.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

The function $f : D \times D \to D$ with $D = \{a, b, c, d\}$.

Cyclic Minors

Definition (Cyclic Minor)

For $A, B \subseteq D$ and $k \in \mathbb{N}$, (A, B, k) is a *cyclic minor* of f if the restriction of f to A and B is addition modulo k, after relabeling the sets A, B, and D.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

The function $f : D \times D \to D$ with $D = \{a, b, c, d\}$.

f		b		d
b		a		b
c		b		c
d		c		a

The function f restricted to $A = \{b, c, d\}$ and $B = \{b, d\}$.

Cyclic Minors

Definition (Cyclic Minor)

For $A, B \subseteq D$ and $k \in \mathbb{N}$, (A, B, k) is a *cyclic minor* of f if the restriction of f to A and B is addition modulo k, after relabeling the sets A, B, and D.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

The function $f: D \times D \to D$ with $D = \{a, b, c, d\}$.

f		b		d
b		a		b
c		b		c
d		c		a

The function f restricted to $A = \{b, c, d\}$ and $B = \{b, d\}$.

$+$		0		1
0		0		1
1		1		2
2		2		0

A relabeling shows that (A, B, k) is a cyclic minor of f with $A = \{b, c, d\}$, $B = \{b, d\}$, and $k = 3$.

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\mathrm{cost}(\mathcal{P}) \coloneqq \sum_{i=1}^{m} k_i$.

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \dots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \dots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

Cyclic Partitions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\operatorname{cost}(\mathcal{P}) \coloneqq \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) \coloneqq \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) \coloneqq \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Cyclic Partitions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \dots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \dots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\text{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\text{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-CONVOLUTION.

Cyclic Partitions

Definition (Cyclic Partition)

A *cyclic partition* of f is a set $\mathcal{P} = \{(A_1, B_1, k_1), \ldots, (A_m, B_m, k_m)\}$ if (A_i, B_i, k_i) is a cyclic minor of f and $A_1 \times B_1, \ldots, A_m \times B_m$ is a partition of $D \times D$.
The *cost* of the cyclic partition $\mathcal{P}$ is $\operatorname{cost}(\mathcal{P}) := \sum_{i=1}^{m} k_i$.

f	a	b	c	d
a	a	d	b	d
b	c	a	d	b
c	b	b	a	c
d	d	c	c	a

There is always a cyclic partition $\mathcal{P}_0$ of cost $|D|^2$:
$\mathcal{P}_0 = \{\ (x, y, 1) \mid x, y \in D\}$.

We show that non-trivial cyclic partitions always exist.

A cyclic partition of f with cost 8:

$$\mathcal{P} = \{(\{b, c, d\}, \{b, d\}, 3), (\{a, c\}, \{a, c\}, 2), (\{b, d\}, \{a, c\}, 2), (\{a\}, \{b, d\}, 1)\}.$$

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\operatorname{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-Convolution.

This improves $\mathbf{4^{2n}} \cdot n^{\mathcal{O}(1)} = \mathbf{16^n} \cdot n^{\mathcal{O}(1)}$ to $(\mathbf{8^n + 4^n}) \cdot n^{\mathcal{O}(1)} = \mathbf{8^n} \cdot n^{\mathcal{O}(1)}$.

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\mathrm{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-CONVOLUTION.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v}=f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

How to Use Cyclic Partitions

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\text{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-Convolution.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v}=f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

Main idea:

1. For each coordinate i: Guess the minor (enumerate them) for the values $\mathbf{u}_i, \mathbf{w}_i$.

How to Use Cyclic Partitions

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\text{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-Convolution.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v}=f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

Main idea:

1. For each coordinate i: Guess the minor (enumerate them) for the values $\mathbf{u}_i, \mathbf{w}_i$.
2. Filter and relabel the values according to the cyclic minors.

How to Use Cyclic Partitions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\text{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-Convolution.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v} = f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

Main idea:

1. For each coordinate i: Guess the minor (enumerate them) for the values $\mathbf{u}_i, \mathbf{w}_i$.
2. Filter and relabel the values according to the cyclic minors.
3. Compute the contribution of the values [van Rooij '21] (it is addition with modulo).

How to Use Cyclic Partitions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\mathrm{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-Convolution.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v}=f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

Main idea:

1. For each coordinate i: Guess the minor (enumerate them) for the values $\mathbf{u}_i, \mathbf{w}_i$.
2. Filter and relabel the values according to the cyclic minors.
3. Compute the contribution of the values [van Rooij '21] (it is addition with modulo).

Running time:

$$\sum_{t \in [1..m]^n} \left(\prod_{i=1}^{n} k_{t[i]} \right) = \left(\sum_{i=1}^{m} k_i \right)^n = \mathrm{cost}(\mathcal{P})^n$$

How to Use Cyclic Partitions

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Lemma (Cyclic Partitions are Useful)

Let $\mathcal{P}$ be a cyclic partition of f.
There is a $(\text{cost}(\mathcal{P})^n + |D|^n) \cdot n^{\mathcal{O}(1)}$ time algorithm for f-CONVOLUTION.

For all $\mathbf{v} \in D^n$, we want to compute

$$(g \circledast_f h)(\mathbf{v}) := \sum_{\mathbf{u},\mathbf{w} \in D^n \text{ s.t. } \mathbf{v}=f(\mathbf{u},\mathbf{w})} g(\mathbf{u}) \cdot h(\mathbf{w}).$$

Main idea:

1. For each coordinate i: Guess the minor (enumerate them) for the values $\mathbf{u}_i, \mathbf{w}_i$.
2. Filter and relabel the values according to the cyclic minors.
3. Compute the contribution of the values [van Rooij '21] (it is addition with modulo).

Running time:

$$\sum_{t \in [1..m]^n} \left(\prod_{i=1}^{n} k_{t[i]} \right) = \left(\sum_{i=1}^{m} k_i \right)^n = \text{cost}(\mathcal{P})^n$$

We show how to find a "good" cyclic partition.

Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

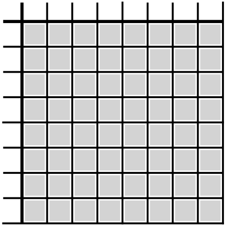
Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.



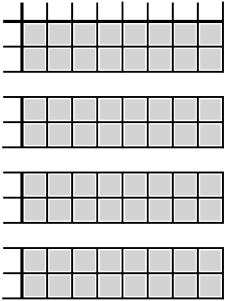
Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1 Partition D into $|D|/2$ sets D_i of size two.

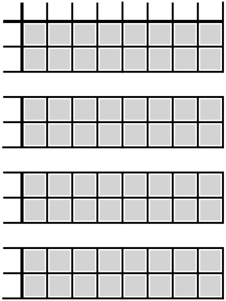


Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.

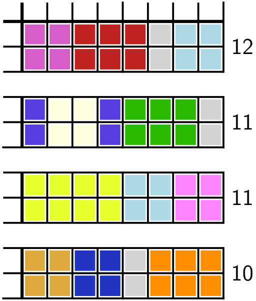


Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.



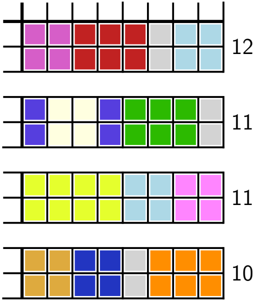
Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.
3. Combine them to one cyclic partition $\mathcal{P}$ of f by setting $\mathcal{P} := \bigcup_i \mathcal{P}_i$.



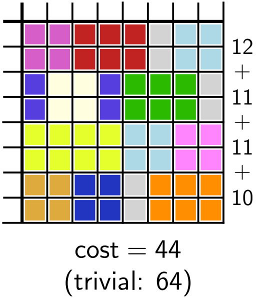
Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.
3. Combine them to one cyclic partition $\mathcal{P}$ of f by setting $\mathcal{P} := \bigcup_i \mathcal{P}_i$.



Finding Cyclic Partitions: Part I

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Lemma (Existence of Non-Trivial Cyclic Partitions)

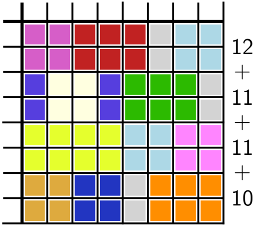
There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.
3. Combine them to one cyclic partition $\mathcal{P}$ of f by setting $\mathcal{P} := \bigcup_i \mathcal{P}_i$.

The cyclic partition $\mathcal{P}$ of f has cost

$$\text{cost}(\mathcal{P}) = \sum_{i=1}^{|D|/2} \text{cost}(\mathcal{P}_i) \leq \frac{|D|}{2} \cdot \frac{5}{3}|D| = \frac{5}{6}|D|^2.$$



cost = 44
(trivial: 64)

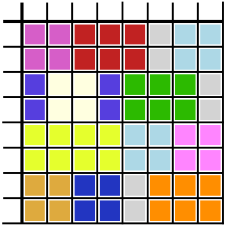
Finding Cyclic Partitions: Part I

Lemma (Existence of Non-Trivial Cyclic Partitions)

There is a cyclic partition $\mathcal{P}$ of f such that $\text{cost}(\mathcal{P}) \leq \frac{5}{6} \cdot |D|^2$.

Proof:

1. Partition D into $|D|/2$ sets D_i of size two.
2. For each i: Construct a cyclic partition $\mathcal{P}_i$ of f restricted to D_i and D with cost at most $5/3 \cdot |D|$.
3. Combine them to one cyclic partition $\mathcal{P}$ of f by setting $\mathcal{P} := \bigcup_i \mathcal{P}_i$.



cost = 44
(trivial: 64)

The cyclic partition $\mathcal{P}$ of f has cost

$$\text{cost}(\mathcal{P}) = \sum_{i=1}^{|D|/2} \text{cost}(\mathcal{P}_i) \leq \frac{|D|}{2} \cdot \frac{5}{3}|D| = \frac{5}{6}|D|^2.$$

Next: Find a "small" cyclic partition of f restricted to D_i and D.

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

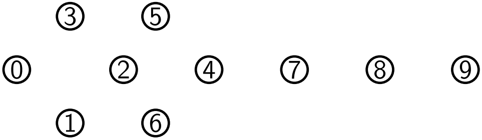
	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \dots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.

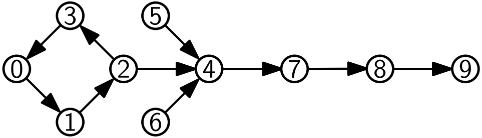


Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.

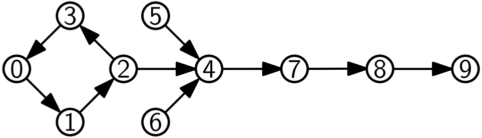


Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.



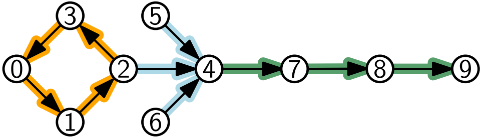
Partition the edges of the graph into nice subgraphs

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.



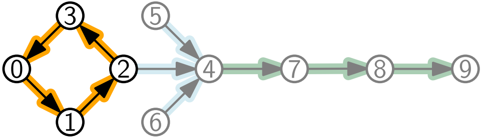
Partition the edges of the graph into nice subgraphs

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.



Partition the edges of the graph into nice subgraphs which are

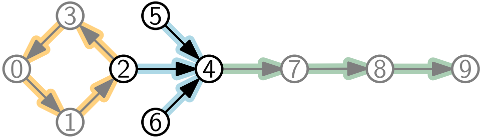
- cycles: directly yield cyclic minors

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9
x	0	1	2	3	2	5	6	4	7	8
y	1	2	3	0	4	4	4	7	8	9

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.



Partition the edges of the graph into nice subgraphs which are

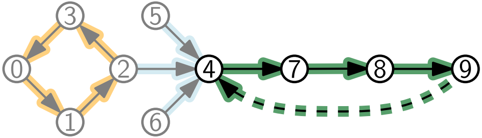
- cycles: directly yield cyclic minors
- stars: can be decomposed into trivial cyclic minors of cost 1

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	2	5	6	4	7	8	9
y	1	2	3	0	4	4	4	7	8	9	4

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x, d), f(y, d)) \mid d \in D\}$.



Partition the edges of the graph into nice subgraphs which are

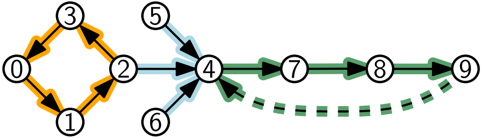
- cycles: directly yield cyclic minors
- stars: can be decomposed into trivial cyclic minors of cost 1
- paths: yield cyclic minors after adding one artificial edge

Finding Cyclic Partitions: Part II

Consider f restricted to $D_i = \{x, y\}$ and D (assume $D = \{0, \ldots, 9\}$):

	0	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	2	5	6	4	7	8	9
y	1	2	3	0	4	4	4	7	8	9	4

Construct a graph G with $V(G) = D$ and $E(G) = \{(f(x,d), f(y,d)) \mid d \in D\}$.



Partition the edges of the graph into nice subgraphs which are

- cycles: directly yield cyclic minors
- stars: can be decomposed into trivial cyclic minors of cost 1
- paths: yield cyclic minors after adding one artificial edge

Our bound is obtained by balancing different ways of decomposing the graph.

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f : L \times R \to T$. This includes the case when f is undefined for certain inputs.

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f : L \times R \to T$. This includes the case when f is undefined for certain inputs.
- One entry of the f-convolution can be computed in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ where $\omega < 2.373$ is the matrix-multiplication exponent.

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f : L \times R \to T$. This includes the case when f is undefined for certain inputs.
- One entry of the f-convolution can be computed in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ where $\omega < 2.373$ is the matrix-multiplication exponent.

Open questions:

Conclusion

Main Theorem

Let D be a finite set and $f : D \times D \rightarrow D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f: L \times R \rightarrow T$. This includes the case when f is undefined for certain inputs.
- One entry of the f-convolution can be computed in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ where $\omega < 2.373$ is the matrix-multiplication exponent.

Open questions:

- Can we show a lower bound?

Conclusion

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-CONVOLUTION with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f : L \times R \to T$. This includes the case when f is undefined for certain inputs.
- One entry of the f-convolution can be computed in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ where $\omega < 2.373$ is the matrix-multiplication exponent.

Open questions:

- Can we show a lower bound?
- Is $|D|^{(2-\epsilon)n} \cdot n^{\mathcal{O}(1)}$ (for some $\epsilon > 0$) or even $|D|^n \cdot n^{\mathcal{O}(1)}$ achievable for all f? (Cyclic partitions seem to be useful to achieve such improvements.)

Conclusion

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Main Theorem

Let D be a finite set and $f : D \times D \to D$. There is an algorithm for f-Convolution with running time $(\frac{5}{6} \cdot |D|^2)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is even, or $(\frac{5}{6} \cdot |D|^2 + \frac{1}{6} \cdot |D|)^n \cdot n^{\mathcal{O}(1)}$ when $|D|$ is odd.

Additional results:

- We generalize the results to f with different finite sets, i.e., $f\colon L \times R \to T$. This includes the case when f is undefined for certain inputs.
- One entry of the f-convolution can be computed in time $|D|^{\omega \cdot n/2} \cdot n^{\mathcal{O}(1)}$ where $\omega < 2.373$ is the matrix-multiplication exponent.

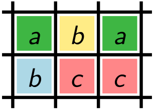
Open questions:

- Can we show a lower bound?
- Is $|D|^{(2-\epsilon)n} \cdot n^{\mathcal{O}(1)}$ (for some $\epsilon > 0$) or even $|D|^n \cdot n^{\mathcal{O}(1)}$ achievable for all f? (Cyclic partitions seem to be useful to achieve such improvements.)

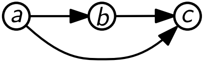
Full version: `arXiv:2209.01623`

Special Structures

Improvements for the following cases might give helpful insights:

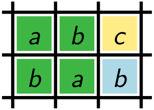


Graph representation:

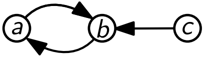


Current cost: 4

Expected cost: 3

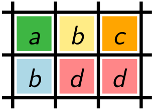


Graph representation:

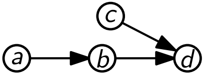


Current cost: 4

Expected cost: 4



Graph representation:



Current cost: 5

Expected cost: 4