Exhaustive search of optimal formulae for bilinear maps

Svyatoslav Covanov
Supervised by Jérémie Detrey and Emmanuel Thomé

Inria, team SPECFUN

November 14th, 2018

Context

For polynomials we have Karatsuba's algorithm: product of $A = a_0 + a_1 X$ by $B = b_0 + b_1 X$ in 3 scalar multiplications.

For matrices we have Strassen's algorithm: product of $\begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix}$ by $\begin{pmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix}$ in 7 scalar multiplications.

Hope: similar formulas for fast computation of other bilinear maps.

Example: polynomial short product, circulant product in finite fields.

Optimal formulas

Mixing existing strategies

Experimental results

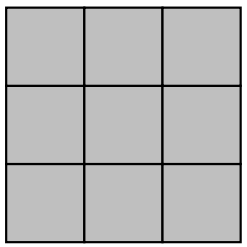
Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

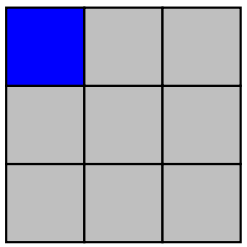
$$a_0b_0 + (a_1b_0 + a_0b_1)X + (a_2b_0 + a_1b_1 + a_0b_2)X^2 + (a_2b_1 + a_1b_2)X^3 + a_2b_2X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

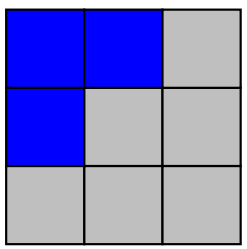
$$a_0b_0 + (a_1b_0 + a_0b_1)X + (a_2b_0 + a_1b_1 + a_0b_2)X^2 + (a_2b_1 + a_1b_2)X^3 + a_2b_2X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

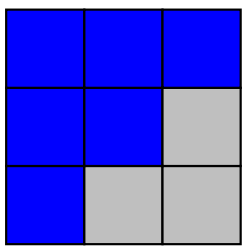
$$a_0b_0 + (a_1b_0 + a_0b_1)X + (a_2b_0 + a_1b_1 + a_0b_2)X^2 + (a_2b_1 + a_1b_2)X^3 + a_2b_2X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

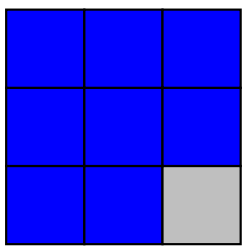
$$a_0b_0 + (a_1b_0 + a_0b_1)X + (a_2b_0 + a_1b_1 + a_0b_2)X^2 + (a_2b_1 + a_1b_2)X^3 + a_2b_2X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1X + a_2X^2$ and $B = b_0 + b_1X + b_2X^2$ modulo X^3

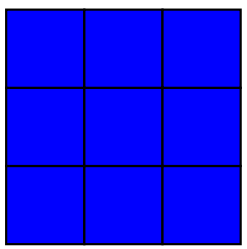
$$a_0b_0 + (a_1b_0 + a_0b_1)X + (a_2b_0 + a_1b_1 + a_0b_2)X^2 + (a_2b_1 + a_1b_2)X^3 + a_2b_2X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1 X + a_2 X^2$ and $B = b_0 + b_1 X + b_2 X^2$ modulo X^3

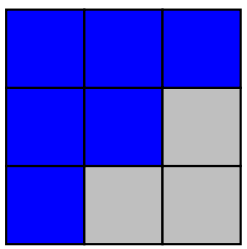
$$a_0 b_0 + (a_1 b_0 + a_0 b_1) X + (a_2 b_0 + a_1 b_1 + a_0 b_2) X^2 + (a_2 b_1 + a_1 b_2) X^3 + a_2 b_2 X^4$$



Short product example

Task (short product): multiply two polynomials in $\mathbb{F}_q[X]$
$A = a_0 + a_1 X + a_2 X^2$ and $B = b_0 + b_1 X + b_2 X^2$ modulo X^3

$$a_0 b_0 + (a_1 b_0 + a_0 b_1) X + (a_2 b_0 + a_1 b_1 + a_0 b_2) X^2 + (a_2 b_1 + a_1 b_2) X^3 + a_2 b_2 X^4$$



$$C = A \cdot B \mod X^3$$

Matrix formalism

$$C = \underbrace{a_0 b_0}_{\Phi_0} + \underbrace{(a_1 b_0 + a_0 b_1)}_{\Phi_1} X + \underbrace{(a_2 b_0 + a_1 b_1 + a_0 b_2)}_{\Phi_2} X^2$$

$$\Phi_0 = \begin{pmatrix} a_0 & a_1 & a_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = a_0 b_0$$

$$\Phi_1 = \begin{pmatrix} a_0 & a_1 & a_2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = a_1 b_0 + a_0 b_1$$

$$\Phi_2 = \begin{pmatrix} a_0 & a_1 & a_2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = a_2 b_0 + a_1 b_1 + a_0 b_2$$

Rank-one matrices

Interest: decompositions of the Φ_i's as linear combinations of

$$\Big(\sum_{0\le i<3} \lambda_i a_i\Big)\Big(\sum_{0\le j<3} \nu_j b_j\Big).$$

We have

$$\Big(\sum \lambda_i a_i\Big)\Big(\sum \nu_j b_j\Big) = \begin{pmatrix} a_0 & a_1 & a_2 \end{pmatrix} \cdot \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \cdot \begin{pmatrix} \nu_0 & \nu_1 & \nu_2 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}.$$

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \cdot \begin{pmatrix} \nu_0 & \nu_1 & \nu_2 \end{pmatrix} = \begin{pmatrix} \lambda_0\nu_0 & \lambda_0\nu_1 & \lambda_0\nu_2 \\ \lambda_1\nu_0 & \lambda_1\nu_1 & \lambda_1\nu_2 \\ \lambda_2\nu_0 & \lambda_2\nu_1 & \lambda_2\nu_2 \end{pmatrix}$$ has rank at most one.

We denote by $\mathcal{G}$ the set of rank-one matrices.

Decomposition with rank-one matrices

$$\underbrace{\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}}_{\Phi_0 = a_0 b_0} \quad \underbrace{\begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix}}_{a_1 b_1} \quad \underbrace{\begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}}_{a_2 b_2} \quad \underbrace{\begin{pmatrix}1&1&0\\1&1&0\\0&0&0\end{pmatrix}}_{(a_0+a_1)(b_0+b_1)} \quad \underbrace{\begin{pmatrix}1&0&1\\0&0&0\\1&0&1\end{pmatrix}}_{(a_0+a_2)(b_0+b_2)}$$

Decomposition with rank-one matrices:

$$\underbrace{\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix}}_{\Phi_1} = \begin{pmatrix}1&1&0\\1&1&0\\0&0&0\end{pmatrix} - \begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix} - \begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix}$$

$$\underbrace{\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}}_{\Phi_2} = \begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix} + \begin{pmatrix}1&0&1\\0&0&0\\1&0&1\end{pmatrix} - \begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix} - \begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}$$

Restatement as a problem of subspace inclusion

For the short product modulo X^3, our new *target* is

$$T = \mathrm{Span}\left(\{\Phi_0, \Phi_1, \Phi_2\}\right).$$

Problem: find all smallest possible $\mathcal{F} \subset \mathcal{G}$ such that

$$T \subset \mathrm{Span}(\mathcal{F}).$$

Example:

$$\mathcal{F} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \right\}.$$

Problem to be solved

Given T a subspace of a matrix space, find all vector spaces of matrices $V = \mathrm{Span}(\mathcal{F})$ of minimal dimension r such that

1. $\mathcal{F} \subset \mathcal{G}$ and
2. $T \subset V$.

Notations:

- $\mathscr{S}_r$ is the set of vector spaces of dimension r generated by rank-one matrices;
- $\mathscr{S}_r(U)$ is the set of vector spaces $V \in \mathscr{S}_r$ such that $U \subset V$;

Problem to be solved

Given T a subspace of a matrix space, find all vector spaces of matrices $V = \mathrm{Span}(\mathcal{F})$ of minimal dimension r such that

1. $\mathcal{F} \subset \mathcal{G}$ and
2. $T \subset V$.

Notations:

- $\mathscr{S}_r$ is the set of vector spaces of dimension r generated by rank-one matrices;
- $\mathscr{S}_r(U)$ is the set of vector spaces $V \in \mathscr{S}_r$ such that $U \subset V$;

New problem:

find the minimal bilinear rank r such that $\mathscr{S}_r(T)$ not empty.

Naive algorithm

Given T of dimension ℓ, start with $r = \ell$ and

1. enumerate $\mathscr{S}_r$;
2. compute $\mathscr{S}_r(T)$;
3. if $\mathscr{S}_r(T)$ is empty, increment r and go back to step 1.

Complexity: (dominated by step 1) $\#\mathscr{S}_r \leq \binom{\#\mathcal{G}}{r}$.

For the short product modulo X^3 over $\mathbb{F}_2$, bilinear rank is $r = 5$.

We have $\#\mathscr{S}_5 = \underbrace{157,535}_{\text{lower bound}} \ll \underbrace{1,906,884}_{\text{upper bound}} = \binom{(2^3-1)^2}{5}$.

Barbulescu, Detrey, Estibals, Zimmerman (BDEZ)

Given T of dimension ℓ, start with $r = \ell$ and

1. enumerate $\mathscr{S}_{r-\ell}$;
2. compute $\mathscr{S}_r(T)$ as $\{T + W \mid W \in \mathscr{S}_{r-\ell} \text{ and } T + W \in \mathscr{S}_r\}$;
3. if $\mathscr{S}_r(T)$ is empty, increment r and go back to step 1.

Correctness: $\mathscr{S}_r(T) \subset T + \mathscr{S}_{r-\ell}$.

Complexity: (dominated by step 1) $\#\mathscr{S}_{r-\ell} \leq \binom{\#\mathcal{G}}{r-\ell}$.

For the short product modulo X^3 over $\mathbb{F}_2$, bilinear rank is $r - \ell = 2$.

We have $\#\mathscr{S}_2 = \underbrace{980}_{\text{lower bound}} < \underbrace{1176}_{\text{upper bound}} = \binom{(2^3-1)^2}{2} \ll \underbrace{157,535}_{\#\mathscr{S}_5}$.

Barbulescu, Detrey, Estibals, Zimmerman (BDEZ)

Given T of dimension ℓ, start with $r = \ell$ and

1. enumerate $\mathscr{S}_{r-\ell}$;
2. compute $\mathscr{S}_r(T)$ as $\{T + W \mid W \in \mathscr{S}_{r-\ell} \text{ and } T + W \in \mathscr{S}_r\}$;
3. if $\mathscr{S}_r(T)$ is empty, increment r and go back to step 1.

Correctness: $\mathscr{S}_r(T) \subset T + \mathscr{S}_{r-\ell}$.

Complexity: (dominated by step 1) $\#\mathscr{S}_{r-\ell} \leq \binom{\#\mathcal{G}}{r-\ell}$.

For the short product modulo X^3 over $\mathbb{F}_2$, bilinear rank is $r - \ell = 2$.

We have $\#\mathscr{S}_2 = \underbrace{980}_{\text{lower bound}} < \underbrace{1176}_{\text{upper bound}} = \binom{(2^3-1)^2}{2} \ll \underbrace{157,535}_{\#\mathscr{S}_5}$.

Faster variant: enumerate $\mathscr{S}_{r-\ell}/\operatorname{Stab}(T)$ (example: $980 \to 68$).

Intermediate strategy (Covanov '17)

Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

set	**cardinality**
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathrm{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathrm{Span}(\Phi_2, \Phi_1))$	6

Intermediate strategy (Covanov '17)

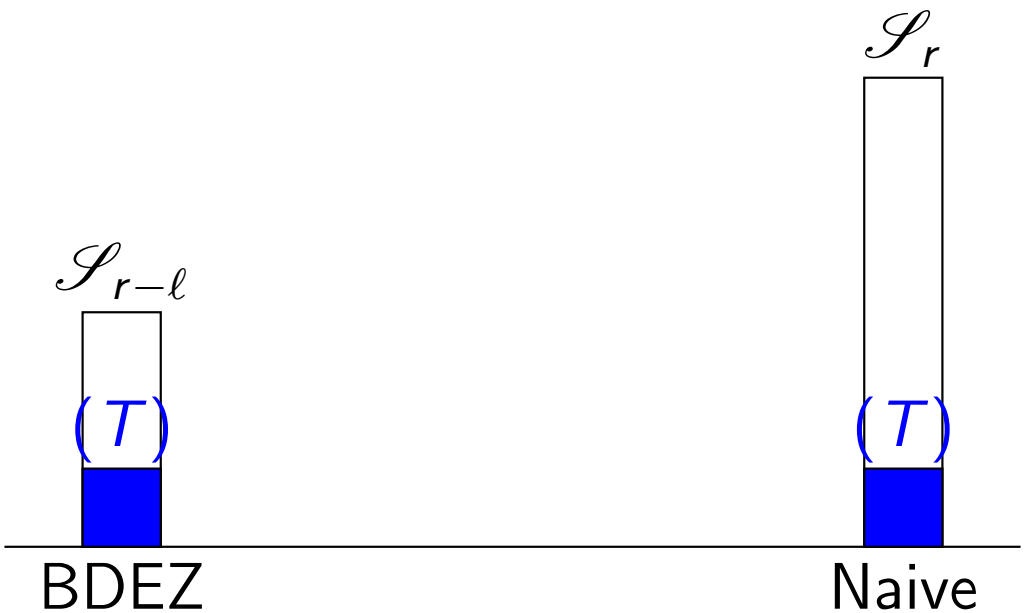
Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

set	**cardinality**
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathrm{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathrm{Span}(\Phi_2, \Phi_1))$	6

Intermediate strategy (Covanov '17)

Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

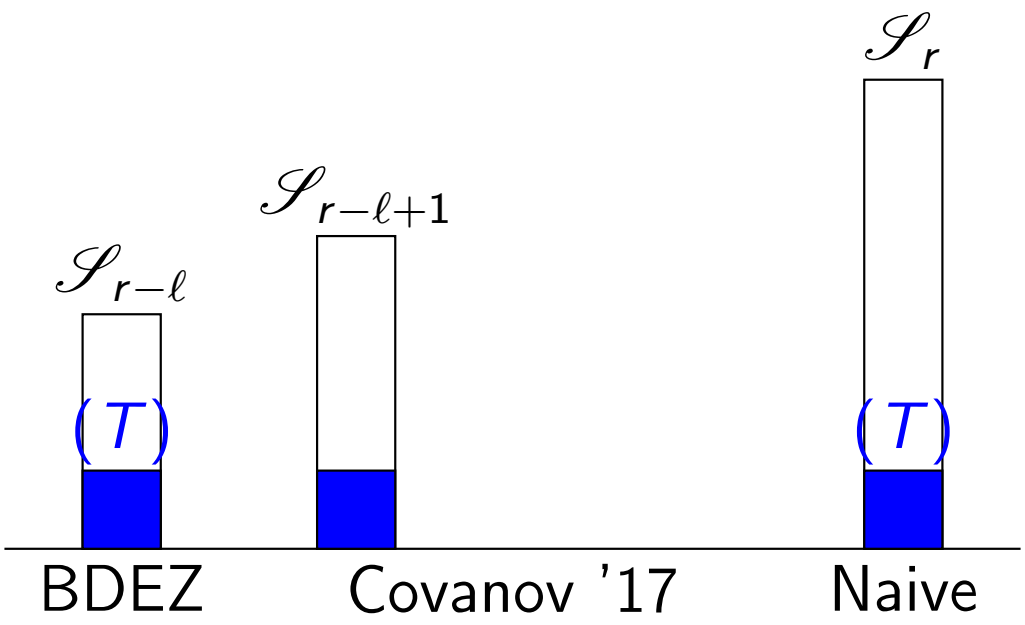
set	**cardinality**
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathsf{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathsf{Span}(\Phi_2, \Phi_1))$	6



Intermediate strategy (Covanov '17)

Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

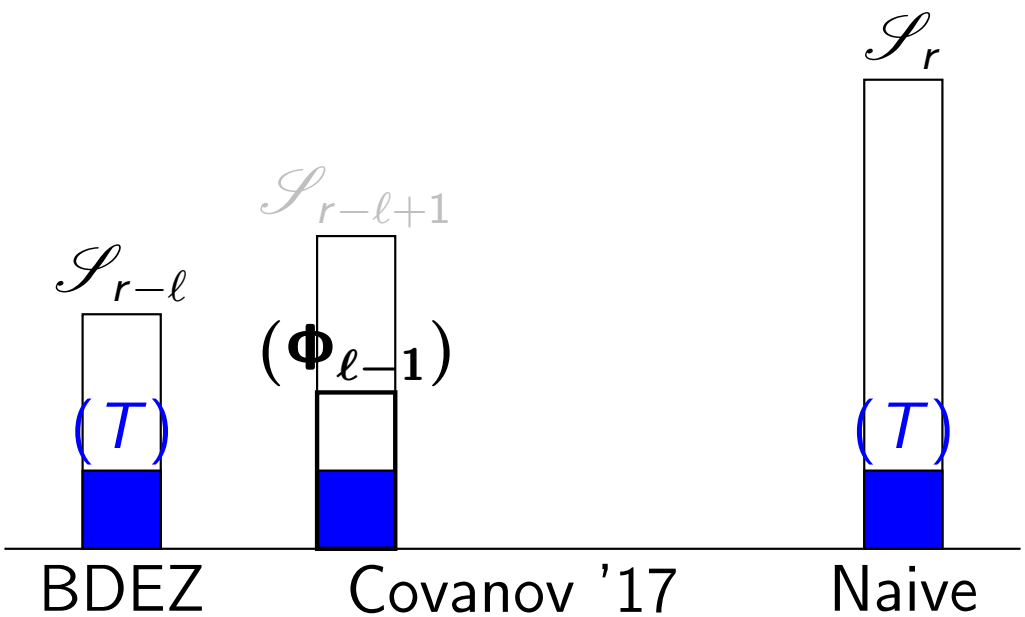
set	**cardinality**
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathrm{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathrm{Span}(\Phi_2, \Phi_1))$	6



Intermediate strategy (Covanov '17)

Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

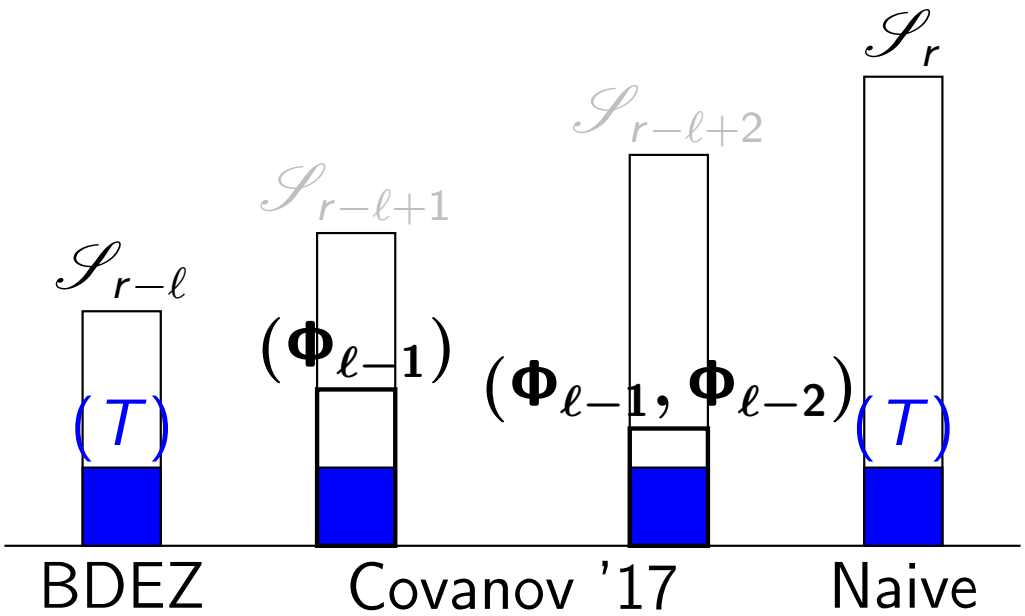
set	cardinality
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathrm{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathrm{Span}(\Phi_2, \Phi_1))$	6



Intermediate strategy (Covanov '17)

Modulo X^3 ($\ell = 3$), $\Phi_{\ell-2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $\Phi_{\ell-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

set	cardinality
$\mathscr{S}_2(\emptyset) = \mathscr{S}_2$	980
$\mathscr{S}_3(\mathrm{Span}(\Phi_2))$	28
$\mathscr{S}_4(\mathrm{Span}(\Phi_2, \Phi_1))$	6



Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Backtracking algorithm:

Recursively add elements of $\mathcal{G}$

$$\Longleftrightarrow$$

Enumerate Span$(g_{i_1}, \ldots, g_{i_{r-\ell+k}})$

Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Backtracking algorithm:	**Group-theoretic algorithm:**
Recursively add elements of $\mathcal{G}$	$G_1 \subset G_2 \subset \cdots \subset \mathrm{GL}_\ell \times \mathrm{GL}_\ell$
$\Longleftrightarrow$	$V_0, \ldots, V_d$ (matrix spaces)
Enumerate $\mathrm{Span}(g_{i_1}, \ldots, g_{i_{r-\ell+k}})$	Enumerate $V_i \circ \sigma_1 \circ \sigma_1 \circ \cdots \circ \sigma_k$

Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Backtracking algorithm:	**Group-theoretic algorithm:**
Recursively add elements of $\mathcal{G}$	$G_1 \subset G_2 \subset \cdots \subset \mathrm{GL}_\ell \times \mathrm{GL}_\ell$
$\Longleftrightarrow$	$V_0, \ldots, V_d$ (matrix spaces)
Enumerate $\mathrm{Span}(g_{i_1}, \ldots, g_{i_{r-\ell+k}})$	Enumerate $V_i \circ \sigma_1 \circ \sigma_1 \circ \cdots \circ \sigma_k$

Complexity: $\approx$ the same in both approaches.

Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Backtracking algorithm:	**Group-theoretic algorithm:**
Recursively add elements of $\mathcal{G}$	$G_1 \subset G_2 \subset \cdots \subset \mathrm{GL}_\ell \times \mathrm{GL}_\ell$
$\Longleftrightarrow$	$V_0, \ldots, V_d$ (matrix spaces)
Enumerate $\mathrm{Span}(g_{i_1}, \ldots, g_{i_{r-\ell+k}})$	Enumerate $V_i \circ \sigma_1 \circ \sigma_1 \circ \cdots \circ \sigma_k$

Complexity: $\approx$ the same in both approaches.

Pruning branches requires:

Properties from rank-one elements	Group invariant

Intuition of the new way to compute $\mathscr{S}_{r-\ell+k}(U)$

Two "equivalent" ways to compute $\mathscr{S}_{r-\ell+k}$

Backtracking algorithm:	**Group-theoretic algorithm:**
Recursively add elements of $\mathcal{G}$	$G_1 \subset G_2 \subset \cdots \subset \mathrm{GL}_\ell \times \mathrm{GL}_\ell$
$\Longleftrightarrow$	$V_0, \ldots, V_d$ (matrix spaces)
Enumerate $\mathrm{Span}(g_{i_1}, \ldots, g_{i_{r-\ell+k}})$	Enumerate $V_i \circ \sigma_1 \circ \sigma_1 \circ \cdots \circ \sigma_k$

Complexity: $\approx$ the same in both approaches.

Pruning branches requires:

Properties from rank-one elements	Group invariant
$U \cap \mathcal{G}$ ($= \emptyset$ in general)	$\mathrm{Stab}(U)$

Stem and covering

Given T of dimension ℓ, start with $r = \ell$ and

1. find a "stem" $\{\mathcal{F}_0, \ldots, \mathcal{F}_{t-1}\}$ giving a "covering" of $\mathscr{S}_r(T)$;
2. enumerate $\mathscr{S}_{r-\ell+k_i}(\mathrm{Span}(\mathcal{F}_i))_i$;
3. compute $\mathscr{S}_r(T)$ as
   $\{T + W \mid W \in \bigcup_i \mathscr{S}_r(\mathrm{Span}(\mathcal{F}_i))_i \text{ and } T + W \in \mathscr{S}_r\}$;
4. if $\mathscr{S}_r(T)$ is empty, increment r and go back to step 1.

Correctness: definition of a stem.
Complexity: difficult to estimate.

Short product modulo X^3: $980 \to 68 \to 6$. Stem: $\{\{\Phi_{\ell-1}, \Phi_{\ell-2}\}\}$.

product	rank	BDEZ (s)	BDEZStab (s)	Contribution (s)
ShProd_4	8	$4.3 \cdot 10^3$	$2.7 \cdot 10$	3.0
ShProd_5	11	$5.7 \cdot 10^{12}$ (est.)	$2.2 \cdot 10^8$ (est.)	$2.4 \cdot 10^3$

Summary of experimental results

Doable with the new method:

- **Short product modulo X^5**
- **Matrix product 3×2 by 2×3**
- **Matrix product 2×3 by 3×2**
- **Circulant product modulo $X^5 - 1$**

Some results:

- New proofs of bilinear rank;
- One equivalence class of formulae for the matrix product 3×2 by 2×3;
- All optimal formulas for the circulant satisfy a non trivial subspace inclusion;
- All optimal formulas for the short product and the circulant product.

Conclusion

- We obtain interesting speed-up for symmetric bilinear maps such as matrix product and short product compared to implementations of BDEZ.
- We introduce the notion of stem and we propose an algorithm to compute faster sets of the form $\mathscr{S}_{r-\ell+k}(U)$.
- Pushing computations further: is it possible to decompose matrix product 3×3 by 3×3?
- What can we do for non symmetric bilinear maps? (polynomial product)