The Automorphism Groups of Linear Codes and Canonical Representatives of Their Semilinear Isometry Classes

Thomas Feulner

University of Bayreuth

November 14, 2009

Linear Code

Linear Code

A (linear) code C is a subspace of $\mathbb{F}_q^n$ of dimension k.

n, k, q are some fixed parameters.

Generator Matrix

Let C be a linear code. $\Gamma \in \mathbb{F}_q^{k \times n}$ is a generator matrix of C, if the rows of Γ form a basis of C.

Set of Generator Matrices of a code

Let Γ be some generator matrix of C. The set of all generator matrices of C is the orbit $GL_k(q)\Gamma$.

Linear Code

Linear Code

A (linear) code C is a subspace of $\mathbb{F}_q^n$ of dimension k.

n, k, q are some fixed parameters.

Generator Matrix

Let C be a linear code. $\Gamma \in \mathbb{F}_q^{k \times n}$ is a generator matrix of C, if the rows of Γ form a basis of C.

Set of Generator Matrices of a code

Let Γ be some generator matrix of C. The set of all generator matrices of C is the orbit $GL_k(q)\Gamma$.

Linear Code

Linear Code

A (linear) code C is a subspace of $\mathbb{F}_q^n$ of dimension k.

n, k, q are some fixed parameters.

Generator Matrix

Let C be a linear code. $\Gamma \in \mathbb{F}_q^{k \times n}$ is a generator matrix of C, if the rows of Γ form a basis of C.

Set of Generator Matrices of a code

Let Γ be some generator matrix of C. The set of all generator matrices of C is the orbit $GL_k(q)\Gamma$.

Equivalence

Definition

Two linear codes C, C' are semilinearly isometric (or equivalent) $\Longleftrightarrow (\varphi, \alpha, \pi)\Gamma$ is a generator matrix of C', with

- a column permutation $\pi \in S_n$
- an automorphism α of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in {\mathbb{F}_q^*}^n$

Equivalence

Definition

Two linear codes C, C' are semilinearly isometric (or equivalent) $\iff (\varphi, \alpha, \pi)\Gamma$ is a generator matrix of C', with

- a column permutation $\pi \in S_n$
- an automorphism α of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in {\mathbb{F}_q^*}^n$

Equivalence

Definition

Two linear codes C, C' are semilinearly isometric (or equivalent) $\iff (\varphi, \alpha, \pi)\Gamma$ is a generator matrix of C', with

- a column permutation $\pi \in S_n$
- an automorphism α of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in {\mathbb{F}_q^*}^n$

Equivalence

Definition

Two linear codes C, C' are semilinearly isometric (or equivalent) $\iff (\varphi, \alpha, \pi)\Gamma$ is a generator matrix of C', with

- a column permutation $\pi \in S_n$
- an automorphism α of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in {\mathbb{F}_q^*}^n$

Semilinear Isometry

Remark

- more general than linear or permutational isometry
- The group is equal to the set of isometries mapping subspaces onto subspaces
- most general definition for linear codes (which guarantees the image to be linear)
- Semilinear isometry is the definition of equivalence in projective geometry

Semilinear Isometry

Remark

- more general than linear or permutational isometry
- The group is equal to the set of isometries mapping subspaces onto subspaces
- most general definition for linear codes (which guarantees the image to be linear)
- Semilinear isometry is the definition of equivalence in projective geometry

Semilinear Isometry

Remark

- more general than linear or permutational isometry
- The group is equal to the set of isometries mapping subspaces onto subspaces
- most general definition for linear codes (which guarantees the image to be linear)
- Semilinear isometry is the definition of equivalence in projective geometry

Semilinear Isometry

Remark

- more general than linear or permutational isometry
- The group is equal to the set of isometries mapping subspaces onto subspaces
- most general definition for linear codes (which guarantees the image to be linear)
- Semilinear isometry is the definition of equivalence in projective geometry

Semilinear Isometry

Remark

- more general than linear or permutational isometry
- The group is equal to the set of isometries mapping subspaces onto subspaces
- most general definition for linear codes (which guarantees the image to be linear)
- Semilinear isometry is the definition of equivalence in projective geometry

Goal

Canonization Algorithm Can

Input: A generator matrix Γ

Output: A generator matrix $\mathrm{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of Γ.

Idea

A similar approach to the calculation of a canonical labelling of a graph (*nauty*, McKay).

Goal

Canonization Algorithm Can

Input: A generator matrix Γ

Output: A generator matrix $\mathrm{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of Γ.

Idea

A similar approach to the calculation of a canonical labelling of a graph (*nauty*, McKay).

Goal

Canonization Algorithm Can

Input: A generator matrix Γ

Output: A generator matrix $\mathrm{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of Γ.

Idea

A similar approach to the calculation of a canonical labelling of a graph (*nauty*, McKay).

Goal

Canonization Algorithm Can

Input: A generator matrix Γ

Output: A generator matrix $\mathrm{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of Γ.

Idea

A similar approach to the calculation of a canonical labelling of a graph (*nauty*, McKay).

Goal

Canonization Algorithm Can

Input: A generator matrix Γ

Output: A generator matrix $\mathrm{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of Γ.

Idea

A similar approach to the calculation of a canonical labelling of a graph (*nauty*, McKay).

Motivation

Use the algorithm to classify linear codes

Build up a database of representatives of each semilinear isometry class.

Leon's Algorithm

- There is only a test on linear isometry of two codes, but no canonical form.
- Difficult to realize such a database without calculation of unique representatives.

Motivation

Use the algorithm to classify linear codes

Build up a database of representatives of each semilinear isometry class.

Leon's Algorithm

- There is only a test on linear isometry of two codes, but no canonical form.
- Difficult to realize such a database without calculation of unique representatives.

$\mathrm{Can}^{\min}$

Example

$\mathrm{Can}^{\min}(\Gamma) = \min\,(A, \varphi, \alpha, \pi)\,\Gamma$

- Some order on $\mathbb{F}_q$ with $0 < 1 < \mu, \forall \mu \in \mathbb{F}_q \setminus \{0, 1\}$
- Colexicographic order on $(\mathbb{F}_q^k, <_{co})$
- Lexicographic ordered n-tuples of columns $((\mathbb{F}_q^k)^n, <_{lex})$

$\mathrm{Can}^{\min}$

Example

$\mathrm{Can}^{\min}(\Gamma) = \min\,(A, \varphi, \alpha, \pi)\,\Gamma$

- Some order on $\mathbb{F}_q$ with $0 < 1 < \mu, \forall \mu \in \mathbb{F}_q \setminus \{0, 1\}$
- Colexicographic order on $(\mathbb{F}_q^k, <_{co})$
- Lexicographic ordered n-tuples of columns $((\mathbb{F}_q^k)^n, <_{lex})$

$\mathrm{Can}^{\min}$

Example

$\mathrm{Can}^{\min}(\Gamma) = \min\left(A, \varphi, \alpha, \pi\right)\Gamma$

- Some order on $\mathbb{F}_q$ with $0 < 1 < \mu, \forall \mu \in \mathbb{F}_q \setminus \{0, 1\}$
- Colexicographic order on $(\mathbb{F}_q^k, <_{co})$
- Lexicographic ordered n-tuples of columns $((\mathbb{F}_q^k)^n, <_{lex})$

$\mathrm{Can}^{\min}$

Example

$\mathrm{Can}^{\min}(\Gamma) = \min\,(A, \varphi, \alpha, \pi)\,\Gamma$

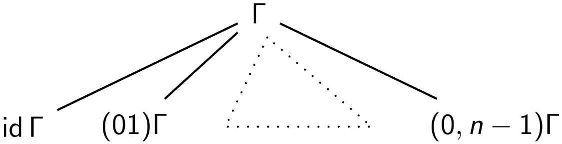
- Some order on $\mathbb{F}_q$ with $0 < 1 < \mu, \forall \mu \in \mathbb{F}_q \setminus \{0, 1\}$
- Colexicographic order on $(\mathbb{F}_q^k, <_{co})$
- Lexicographic ordered n-tuples of columns $((\mathbb{F}_q^k)^n, <_{lex})$

First idea for $\mathrm{Can}^{\mathrm{min}}$: Backtracking on S_n

A naive approach – Run through all possible permutations

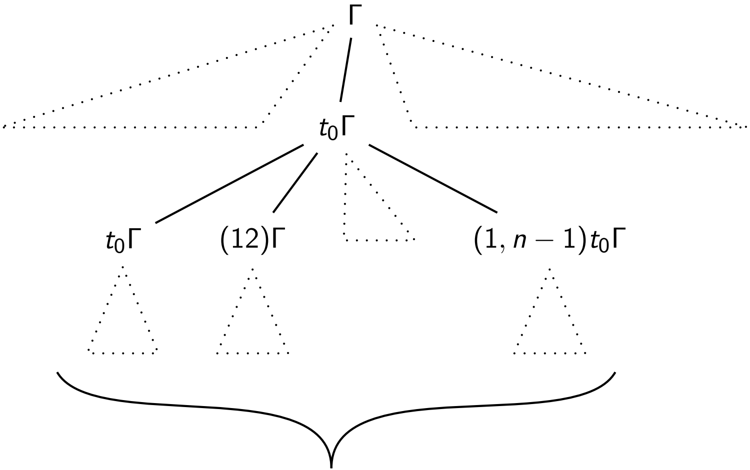
We systematically run through all possible permutations of the columns using a backtracking procedure:

There are n possible choices for the preimage of 0.



First idea for $\mathrm{Can}^{\min}$: Backtracking on S_n

There are $n-1$ choices for the preimage of 1 in each node.



This subtree contains all permutations $S_n^{(1)} t_0$.

i-semicanonical representatives

Suppose we reached some level i (Root is on level 0 by definition). The columns $\{0, \ldots, i-1\}$ will not be permuted anymore!

Definition: i-semicanonical representative

Replace nodes $\pi\Gamma$ by representatives $\Gamma^{(i,\pi)}$ which are minimal on the columns $\{0, \ldots, i-1\}$, i.e.

$$\Pi_i(\Gamma^{(i,\pi)}) := \min \Pi_i((A, \varphi, \alpha)\Gamma).$$

Corollary

$\mathrm{Can}^{\min}(\Gamma) = \min_{\pi \in S_n} \Gamma^{(n,\pi)}$ is the minimum of all n-semicanonical representatives of the leaf nodes.

i-semicanonical representatives

Suppose we reached some level i (Root is on level 0 by definition). The columns $\{0, \ldots, i-1\}$ will not be permuted anymore!

Definition: i-semicanonical representative

Replace nodes $\pi\Gamma$ by representatives $\Gamma^{(i,\pi)}$ which are minimal on the columns $\{0, \ldots, i-1\}$, i.e.

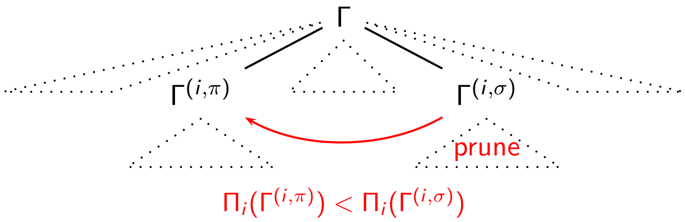
$$\Pi_i(\Gamma^{(i,\pi)}) := \min \Pi_i((A, \varphi, \alpha)\Gamma).$$

Corollary

$\mathrm{Can}^{\min}(\Gamma) = \min_{\pi \in S_n} \Gamma^{(n,\pi)}$ is the minimum of all n-semicanonical representatives of the leaf nodes.

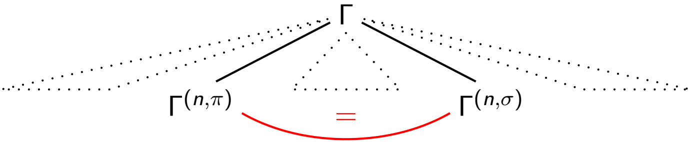
Pruning I

We can compare the i-semicanonical representatives of two nodes on level i.



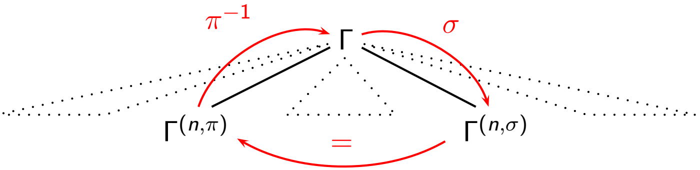
Automorphisms

Two equal leaf nodes give rise to an automorphism.



Automorphisms

Two equal leaf nodes give rise to an automorphism.



Automorphism

Let $\pi, \sigma \in S_n$ be two permutations:

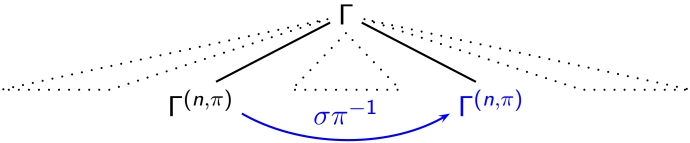
$$\Gamma^{(n,\pi)} = \Gamma^{(n,\sigma)}$$

$$\Downarrow$$

$$\exists (A, \varphi, \alpha) : (A, \varphi, \pi^{-1}\sigma, \alpha) \in Aut(\Gamma)$$

Pruning II – The group of known automorphisms

An automorphism give equal leaf nodes.



Pruning

Let $\pi, \sigma \in S_n$ be two permutations:

$$\Gamma^{(n,\pi)} = \Gamma^{(n,\sigma)}$$

$$\Downarrow \Uparrow$$

$$\exists(A, \varphi, \alpha) : (A, \varphi, \pi^{-1}\sigma, \alpha) \in Aut(\Gamma)$$

Iterative Calculation of *i*-semicanonical representatives

We explain the algorithm by an example over $\mathbb{F}_4 = \{0, 1, x, x^2\}$ where $x^2 + x + 1 = 0$ and $0 < 1 < x < x^2$.

We want to calculate $\Gamma^{(n,\mathrm{id})}$ with

$$\Gamma := \begin{pmatrix} 1 & x & x^2 & 1 & x & 1 \\ x & 0 & x^2 & 1 & 1 & 1 \\ x & 1 & 0 & 1 & 0 & x \end{pmatrix}$$

Gaussian Elimination

Mapping the first column onto the unit vector e_0^T yields an 1-semicanonical representative:

Iterative Calculation of *i*-semicanonical representatives

We explain the algorithm by an example over $\mathbb{F}_4 = \{0, 1, x, x^2\}$ where $x^2 + x + 1 = 0$ and $0 < 1 < x < x^2$.

We want to calculate $\Gamma^{(n,\mathrm{id})}$ with

$$\Gamma := \begin{pmatrix} 1 & x & x^2 & 1 & x & 1 \\ x & 0 & x^2 & 1 & 1 & 1 \\ x & 1 & 0 & 1 & 0 & x \end{pmatrix}$$

Gaussian Elimination

Mapping the first column onto the unit vector e_0^T yields an 1-semicanonical representative:

Example, $i = 1$

$$\Gamma := \begin{pmatrix} 1 & x & x^2 & 1 & x & 1 \\ x & 0 & x^2 & 1 & 1 & 1 \\ x & 1 & 0 & 1 & 0 & x \end{pmatrix}$$

We can multiply Γ by

$$A := \begin{pmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ x & 0 & 1 \end{pmatrix}$$

to get

$$\Gamma^{(1,\mathrm{id})} := \begin{pmatrix} 1 & x & x^2 & 1 & x & 1 \\ 0 & x^2 & x & x^2 & x & x^2 \\ 0 & x & 1 & x^2 & x^2 & 0 \end{pmatrix}$$

Example, $i = 2$

Rule 1

If the column $i - 1$ is linearly independent from the columns with indices $\{0, \ldots, i-2\}$ then map it to the smallest possible unit vector.

Example, $i = 2$

Rule 1

If the column $i - 1$ is linearly independent from the columns with indices $\{0, \ldots, i-2\}$ then map it to the smallest possible unit vector.

$$\Gamma^{(1,\mathrm{id})} := \begin{pmatrix} 1 & x & x^2 & 1 & x & 1 \\ 0 & x^2 & x & x^2 & x & x^2 \\ 0 & x & 1 & x^2 & x^2 & 0 \end{pmatrix}$$

$$\Gamma^{(2,\mathrm{id})} := \begin{pmatrix} 1 & 0 & x & x^2 & x^2 & x^2 \\ 0 & 1 & x^2 & 1 & x^2 & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix}$$

Example, $i = 3$

Question

How can we minimize the column $\begin{pmatrix} x \\ x^2 \\ 0 \end{pmatrix}$?

Equivalently:

What is the stabilizer of $\Pi_2(\Gamma^{(2,\mathrm{id})})$ under the inner group action?

$$\Gamma^{(2,\mathrm{id})} := \begin{array}{cc} & \begin{array}{cccccc} & \mu^{-1} & & & & \end{array} \\ \begin{array}{c} \\ \mu \\ \\ \end{array} & \begin{pmatrix} 1 & 0 & x & x^2 & x^2 & x^2 \\ 0 & 1 & x^2 & 1 & x^2 & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix} \end{array}$$

Example, $i = 3$

Question

How can we minimize the column $\begin{pmatrix} x \\ x^2 \\ 0 \end{pmatrix}$?

Equivalently:

What is the stabilizer of $\Pi_2(\Gamma^{(2,\mathrm{id})})$ under the inner group action?

$$\Gamma^{(2,\mathrm{id})} := \begin{array}{c} \\ x^2 \\ x \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} x & x^2 & & & & \end{array} \\ \begin{pmatrix} 1 & 0 & x & x^2 & x^2 & x^2 \\ 0 & 1 & x^2 & 1 & x^2 & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix} \end{array}$$

Example, $i = 3$

$$
\Gamma^{(2,\mathrm{id})} := \begin{array}{c} \\ x^2 \\ x \\ \\ \end{array}
\begin{array}{c}
\begin{array}{cccccc} x & x^2 & & & & \end{array} \\
\begin{pmatrix}
1 & 0 & x & x^2 & x^2 & x^2 \\
0 & 1 & x^2 & 1 & x^2 & 1 \\
0 & 0 & 0 & 1 & x & x
\end{pmatrix}
\end{array}
$$

$$
\Gamma^{(3,\mathrm{id})} := \begin{pmatrix}
1 & 0 & 1 & x & x & x \\
0 & 1 & 1 & x & 1 & x \\
0 & 0 & 0 & 1 & x & x
\end{pmatrix}
$$

Example, $i = 3$

Stabilizer for $\Pi_3(\Gamma^{(3,\mathrm{id})})$?

$$\Gamma^{(3,\mathrm{id})} := \begin{array}{cc} & \begin{array}{cccccc} \mu^{-1} & \mu^{-1} & \mu^{-1} & & & \end{array} \\ \begin{array}{c} \mu \\ \mu \\ \\ \end{array} & \begin{pmatrix} 1 & 0 & 1 & x & x & x \\ 0 & 1 & 1 & x & 1 & x \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix} \end{array}$$

Partition of row index set

Let $\mathfrak{p}^{(\Gamma,i)}$ be the finest partition of $\{0,\ldots,k-1\}$ such that

$$\forall\, j \in \{0,\ldots,i-1\}\ \exists p \in \mathfrak{p}^{(\Gamma,i)} : \mathrm{supp}(\Gamma_{*,j}) \subseteq p$$

Example, $i = 4$

The case $i = 4$ is done by Rule 1.

$$\Gamma^{(4,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & x & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix}$$

Example, $i = 5$

Rule 2

Suppose column $i-1$ is linearly dependent from the columns with indices $\{0, \dots, i-2\}$.

For each $p \in \mathfrak{p}^{(\Gamma, i-1)}$ take the maximal index of a nonzero entry $\Gamma_{j,i-1}$ with $j \in p$.

Map those entries to 1.

$$\mathfrak{p}^{(\Gamma,4)} = \{\color{blue}{\{0,1\}}, \color{red}{\{2\}}\}$$

$$\Gamma^{(4,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & \color{blue}{x} & 1 \\ 0 & 0 & 0 & 1 & \color{red}{x} & x \end{pmatrix}$$

Example, $i = 5$

$$\mathfrak{p}^{(\Gamma,4)} = \{\{0,1\},\{2\}\}$$

$$\Gamma^{(4,\mathrm{id})} := \begin{array}{c} \\ x^2 \\ x^2 \\ x^2 \end{array} \begin{array}{c} \begin{array}{cccccc} x & x & x & x & & \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & x & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix} \end{array}$$

Example, $i = 5$

$$\mathfrak{p}^{(\Gamma,4)} = \{\{0,1\},\{2\}\}$$

$$\Gamma^{(4,\mathrm{id})} := \begin{array}{cc} & \begin{array}{cccccc} x & x & x & x & & \end{array} \\ \begin{array}{c} x^2 \\ x^2 \\ x^2 \end{array} & \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & x & 1 \\ 0 & 0 & 0 & 1 & x & x \end{pmatrix} \end{array}$$

$$\Gamma^{(4,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & x^2 & x^2 \\ 0 & 1 & 1 & 0 & 1 & x^2 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Example, $i = 5$

$$\Gamma^{(4,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & x^2 & x^2 \\ 0 & 1 & 1 & 0 & 1 & x^2 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

is not a 5-semicanonical representative!

Application of the field automorphism

There is one more free parameter $\alpha \in Aut(\mathbb{F}_q)$ of the inner group!

Example, $i = 5$

$$\Gamma^{(4,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & x^2 & x^2 \\ 0 & 1 & 1 & 0 & 1 & x^2 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

is not a 5-semicanonical representative!

Application of the field automorphism

There is one more free parameter $\alpha \in Aut(\mathbb{F}_q)$ of the inner group!

Apply the Frobenius automorphism on each entry.

$$\Gamma^{(5,\mathrm{id})} := \begin{pmatrix} 1 & 0 & 1 & 0 & x & x \\ 0 & 1 & 1 & 0 & 1 & x \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Minimization by Field automorphisms

Rule 3

Go from the bottom-up through the column:

Minimize the entries $\Gamma_{j,i-1}$ by the application of the remaining field automorphisms.

Restrict in each step the remaining automorphisms to those which additionally fix $\Gamma_{j,i-1}$.

Pruning III – Partition and Refinement

The search tree is still to huge to get results for larger parameters.

Remember the example with (4096)! possible permutations.

Partition and Refinement

This is a well-known approach also used for example in the program *nauty* (McKay) to calculate a canonical labeling of a graph based on invariants.

Example: Using the Weight Enumerator

Suppose Γ is equal to

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Example: Using the Weight Enumerator

Suppose Γ is equal to

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Calculate the weight enumerator of the punctured codes $C_j,\ j = 0, \ldots, 3$:

$$\begin{array}{c|l} 0 & 1 + x + x^2 + x^3 \\ 1 & 1 + 3x^2 \\ 2 & 1 + x + x^2 + x^3 \\ 3 & 1 + 3x^2 \end{array}$$

Example: Using the Weight Enumerator

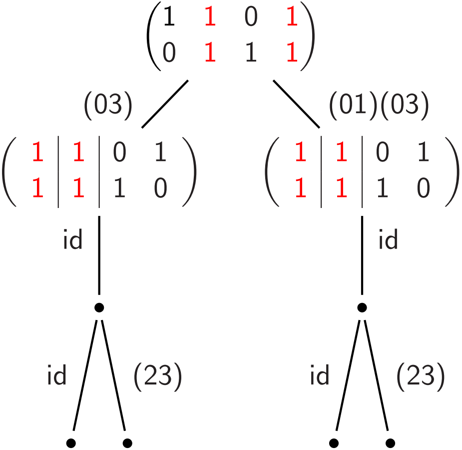
Suppose Γ is equal to

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Calculate the weight enumerator of the punctured codes $C_j,\ j = 0, \ldots, 3$:

$$\begin{array}{c|l} 0 & 1 + x + x^2 + x^3 \\ 1 & 1 + 3x^2 \\ 2 & 1 + x + x^2 + x^3 \\ 3 & 1 + 3x^2 \end{array}$$

Example: Using the Weight Enumerator



Generalization for finite chain rings

Currently Possible

If R is a finite commutative chain ring. We can show that the inner minimization algorithm is still easy to handle.

A first version is already implemented.

Future Work

Show that the inner minimization algorithm is still easy for non-commutative chain rings.

Generalization for finite chain rings

Currently Possible

If R is a finite commutative chain ring. We can show that the inner minimization algorithm is still easy to handle.

A first version is already implemented.

Future Work

Show that the inner minimization algorithm is still easy for non-commutative chain rings.

Conclusion

Test the program online

`http://www.algorithm.uni-bayreuth.de/en/research/Coding_Theory/CanonicalForm/index.html`

Thank you very much for your attention.