Canonization of Linear Codes

Thomas Feulner

University of Bayreuth

July 12, 2010

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 $GL_k(\mathbb{F}_q)\Gamma$.

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 $GL_k(\mathbb{F}_q)\Gamma$.

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 $GL_k(\mathbb{F}_q)\Gamma$.

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 $\varphi \in {\mathbb{F}_q^*}^n$

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.

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.

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.

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.

Goal: Canonization

Tool: Group action on generator matrices

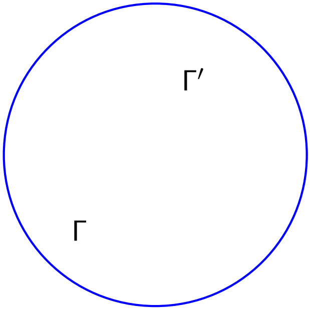
$$\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \left(\mathrm{Aut}(\mathbb{F}_q) \times S_n\right)$$

Goal: Canonization

Tool: Group action on generator matrices

$$\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$$

Let $\Gamma, \Gamma' \in \mathbb{F}_q^{k \times n}$ be equivalent generator matrices



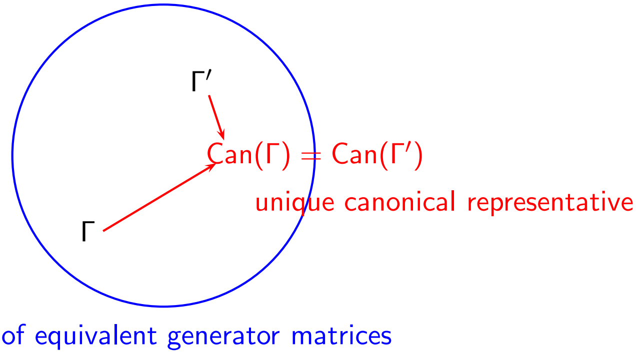
orbit of equivalent generator matrices

Goal: Canonization

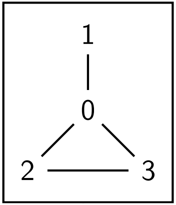
Tool: Group action on generator matrices

$$\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \left(\mathrm{Aut}(\mathbb{F}_q) \times S_n\right)$$

Let $\Gamma, \Gamma' \in \mathbb{F}_q^{k \times n}$ be equivalent generator matrices



The partition and refinement idea



There is a well-known, very fast canonization algorithm for graphs:

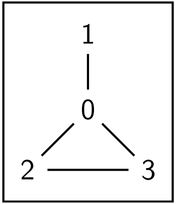
nauty (B. McKay)

based on

Partition & Refinement

The Refinement step

Calculate properties of the vertices, invariant under relabeling!

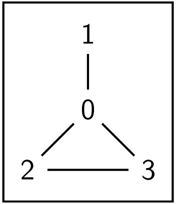


Calculate the degree of the vertices

i	0	1	2	3
degree(i)	3	1	2	2

The Refinement step

Calculate properties of the vertices, invariant under relabeling!



Calculate the degree of the vertices

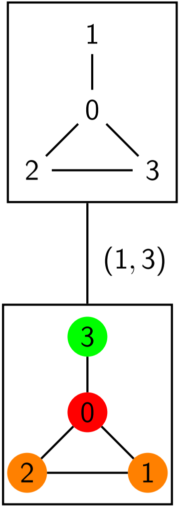
i	0	1	2	3
degree(i)	3	1	2	2

Sort in descending order

i	0	3	2	1
degree(i)	3	2	2	1

The Refinement step

Calculate properties of the vertices, invariant under relabeling!



Calculate the degree of the vertices

i	0	1	2	3
degree(i)	3	1	2	2

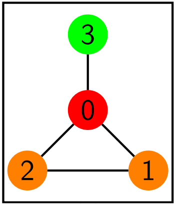
Sort in descending order

i	0	3	2	1
degree(i)	3	2	2	1

Relabel the vertices

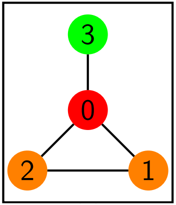
i	0	1	2	3
degree(i)	3	2	2	1

The Partition step



Do a backtracking procedure.

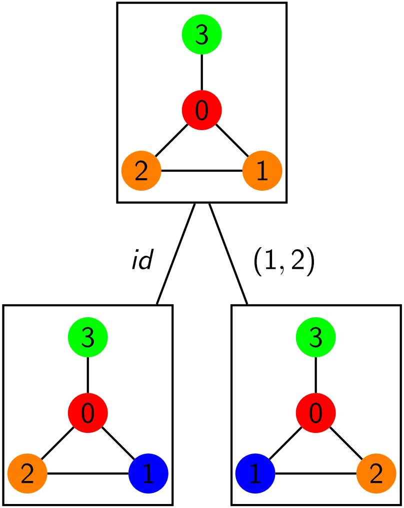
The Partition step



Do a backtracking procedure.

Choose a block of vertices which have the same color.

The Partition step

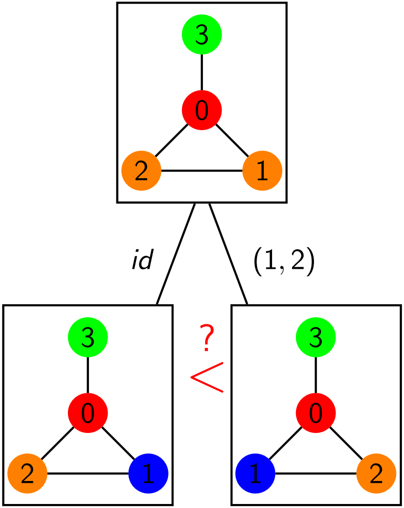


Do a backtracking procedure.

Choose a block of vertices which have the same color.

Investigate all possibilities to color one vertex in this block with a new color and to give it the smallest label.

The Partition step



Do a backtracking procedure.

The comparison of the leaf nodes yields "=":

- $(1,3)$ and $(1,2)(1,3)$ map the graph to its canonical representative
- $(1,3)^{-1}(1,2)(1,3)$ is the only automorphism

Comparison: Graphs and linear Codes

	Graphs	linear Codes
Group Action	$S_n \setminus\!\!\setminus 2^{\binom{n}{2}}$	$\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times {\mathbb{F}_q^*}^n\right) \rtimes \left(\mathrm{Aut}(\mathbb{F}_q) \times S_n\right)\right) \setminus\!\!\setminus \mathbb{F}_q^{k \times n}$

Comparison: Graphs and linear Codes

	Graphs	linear Codes
Group Action	$S_n \backslash\!\backslash 2^{\binom{n}{2}}$	$\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times {\mathbb{F}_q^*}^n\right) \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)\right) \backslash\!\backslash \mathbb{F}_q^{k\times n}$ replace by $S_n \backslash\!\backslash \left[\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \mathrm{Aut}(\mathbb{F}_q)\right) \backslash\!\backslash \mathbb{F}_q^{k\times n}\right]$

Comparison: Graphs and linear Codes

	Graphs	linear Codes
Group Action	$S_n \backslash\!\backslash 2^{\binom{n}{2}}$	$S_n \backslash\!\backslash \left[\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \mathrm{Aut}(\mathbb{F}_q)\right) \backslash\!\backslash \mathbb{F}_q^{k \times n}\right]$

Comparison: Graphs and linear Codes

	Graphs	linear Codes
Group Action	$S_n \backslash\backslash 2^{\binom{n}{2}}$	$S_n \backslash\backslash \left[\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \mathrm{Aut}(\mathbb{F}_q)\right) \backslash\backslash \mathbb{F}_q^{k\times n}\right]$
Refinement	G-homomorphism for some appropriate $G \leq S_n$; $2^{\binom{n}{2}} \to X^n$	

Homomorphism of group actions

Let G act on X, Y.
$f : X \to Y$ is a G-homomorphism if

$$f(gx) = gf(x), \; \forall\, x \in X, g \in G$$

Comparison: Graphs and linear Codes

	Graphs	linear Codes
Group Action	$S_n \backslash\backslash 2^{\binom{n}{2}}$	$S_n \backslash\backslash \left[\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \mathrm{Aut}(\mathbb{F}_q)\right) \backslash\backslash \mathbb{F}_q^{k\times n}\right]$
Re-fine-ment	G-homomorphism for some appropriate $G \leq S_n$	
	$2^{\binom{n}{2}} \to X^n$	$\left(\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n\right) \rtimes \mathrm{Aut}(\mathbb{F}_q)\right) \backslash\backslash \mathbb{F}_q^{k\times n} \to X^n$

Homomorphism of group actions

Let G act on X, Y.
$f : X \to Y$ is a G-homomorphism if

$$f(gx) = gf(x),\ \forall\, x \in X, g \in G$$

An example in the binary case

Canonize the matrix

$$\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 4}$$

An example in the binary case

Canonize the matrix

$$\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 4}$$

Refinement step

Find a S_n-homomorphism

$$f : \left(\mathrm{GL}_3(\mathbb{F}_2)\backslash\backslash \mathbb{F}_2^{3\times 4}\right) \to X^n$$

An example in the binary case: First Refinement

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

Use

$$f\left(\mathrm{GL}_3(\mathbb{F}_2)\cdot\Gamma\right) := \left(\dim(C_0^\Gamma), \ldots, \dim(C_3^\Gamma)\right)$$

C^Γ := the code generated by Γ
C_i := the puncturing of C at postion i

An example in the binary case: First Refinement

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

Use

$$f\left(\mathrm{GL}_3(\mathbb{F}_2) \cdot \Gamma\right) := (3, 2, 3, 3)$$

An example in the binary case: First Refinement

$$\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Use

$$f\left(GL_3(\mathbb{F}_2) \cdot \Gamma\right) := (3, 2, 3, 3) \rightsquigarrow (2, 3, 3, 3)$$

An example in the binary case: Refinement

$$
\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}
\Big\downarrow (0,1)
$$

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

An example in the binary case: Refinement

$$\begin{pmatrix} 0 & {\color{red}1} & 0 & 0 \\ 1 & {\color{red}1} & 1 & 0 \\ 0 & {\color{red}1} & 1 & 1 \end{pmatrix} \Big\downarrow (0,1) \begin{pmatrix} {\color{red}1} & 0 & 0 & 0 \\ {\color{red}0} & 1 & 1 & 0 \\ {\color{red}0} & 0 & 1 & 1 \end{pmatrix}$$

Application of the inner group action:

Minimize the fixed columns.

An example in the binary case: Refinement

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

$$\Big\downarrow (0,1)$$

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

Application of the inner group action:

Minimize the fixed columns.

Further Application of the inner group action:

Use just the stabilizer of the fixed columns for further minimization.

An example in the binary case: Refinement

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

$$\Big\downarrow (0,1)$$

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

Application of the inner group action:

Minimize the fixed columns.

Further Application of the inner group action:

Use just the stabilizer of the fixed columns for further minimization.

Further Refinements:

Restrict to $G \leq S_n$ stabilizing the colors.

An example in the binary case: Backtracking (Partitioning)

$$\begin{tikzcd}[column sep=small]
& \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \arrow[d, "{(0,1)}"] & \\
& \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dl, "(\mathrm{id})"'] \arrow[d, "{(1,2)}"] \arrow[dr, "{(1,3)}"] & \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}
\end{tikzcd}$$

An example in the binary case: Backtracking (Partitioning)

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

$$\big\downarrow (0,1)$$

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

$$\swarrow (\mathrm{id}) \qquad \big\downarrow (1,2) \qquad \searrow (1,3)$$

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

Application of the inner group action:

Minimize the fixed columns

Application of the inner group action:

Prune nodes, whose fixed columns are not minimal.

The whole example

$$
\begin{tikzcd}[column sep=small]
 & & & & & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \arrow[d, "{(0,1)}"] & & & & & \\
 & & & & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dllll, "(\mathrm{id})"'] \arrow[d, "{(1,2)}"] \arrow[drrrr, "{(1,3)}"] & & & & & \\
 & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dl, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dl, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dl, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}
\end{tikzcd}
$$

The whole example

$$
\begin{tikzcd}
 & & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \arrow[d, "{(0,1)}"] & & & \\
 & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[dll, "(\mathrm{id})"'] \arrow[d, "{(1,2)}"] \arrow[drr, "{(1,3)}"] & & & \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \arrow[d, "(\mathrm{id})"'] \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}
\end{tikzcd}
$$

Canonical representative:

A minimal **(including images of the invariants)** leaf node of the pruned tree.

The whole example

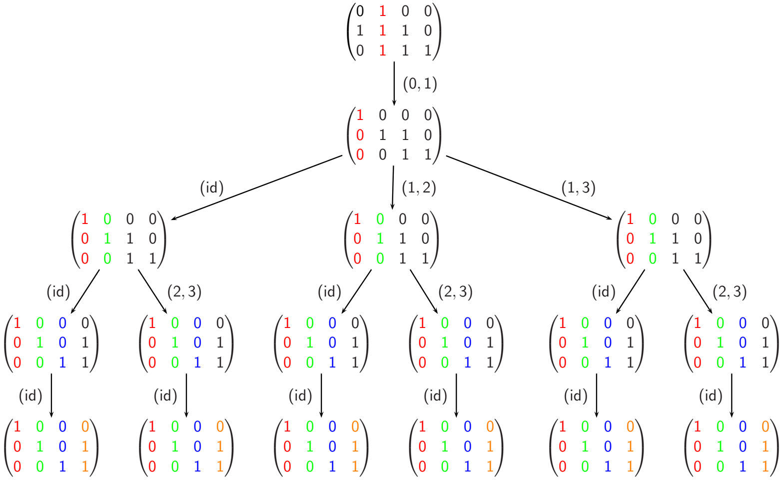
$$
\begin{tikzcd}[column sep=small]
 & & \begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&1&1\end{pmatrix} \arrow[d, "{(0,1)}"] & & & \\
 & & \begin{pmatrix}1&0&0&0\\0&1&1&0\\0&0&1&1\end{pmatrix} \arrow[dll, "(\mathrm{id})"'] \arrow[d, "{(1,2)}"] \arrow[drr, "{(1,3)}"] & & & \\
\begin{pmatrix}1&0&0&0\\0&1&1&0\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & \begin{pmatrix}1&0&0&0\\0&1&1&0\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & & \begin{pmatrix}1&0&0&0\\0&1&1&0\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] \arrow[dr, "{(2,3)}"] & \\
\begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} \arrow[d, "(\mathrm{id})"'] \\
\begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&1\end{pmatrix}
\end{tikzcd}
$$

Automorphisms:

$(\text{root} \to \text{equal leaf node})^{-1} \cdot (\text{root} \to \text{leaf node})$

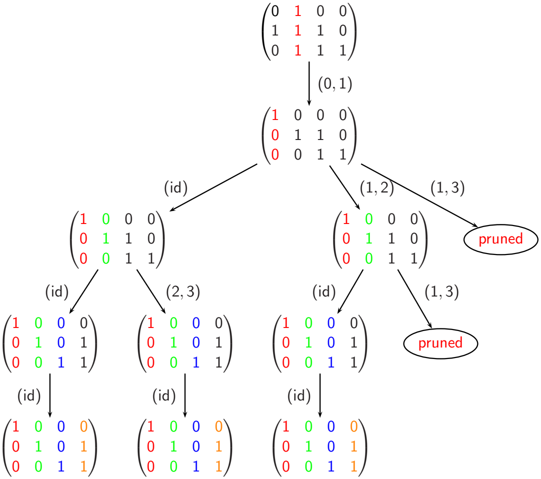
Pruning by Automorphism

Traverse the tree in depth-first-search



Pruning by Automorphism

Traverse the tree in depth-first-search



Application of Automorphisms:

Prune subtrees which carry no new information.

Canonization of APN-Functions

CCZ-Equivalence

CCZ-Equivalence = usual code equivalence

EA-Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & B & C \end{pmatrix}$$

Affine Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & 0 & C \end{pmatrix}$$

Canonization of APN-Functions

CCZ-Equivalence

CCZ-Equivalence = usual code equivalence

EA-Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & B & C \end{pmatrix}$$

Affine Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & 0 & C \end{pmatrix}$$

Canonization of APN-Functions

CCZ-Equivalence

CCZ-Equivalence = usual code equivalence

EA-Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & B & C \end{pmatrix}$$

Affine Equivalence

Restrict the inner group $\mathrm{GL}_k(\mathbb{F}_2)$ to the subgroup

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A & 0 \\ b & 0 & C \end{pmatrix}$$