On canonical forms of ring-linear codes

Thomas Feulner

University of Bayreuth

April 2013
Problem definition

Goal: A canonization algorithm

Let $G$ be a group, which acts on a set $X$.
For some arbitrary element $x \in X$ compute:

Canonical Form A unique representative $\text{CF}_G(x)$ of the orbit $Gx$ of $x$, i.e. $\text{CF}_G(x) = \text{CF}_G(gx)$ for all $g \in G$.

Transporter Element A group element $\text{TR}_G(x) := g \in G$ such that $gx = \text{CF}_G(x)$.

Automorphism Group The stabilizer $\text{Stab}_G(x) := \{g \in G \mid gx = x\}$ of $x$. 

Problem definition

Goal: A canonization algorithm

Let $G$ be a group, which acts on a set $X$.
For some arbitrary element $x \in X$ compute:

**Canonical Form** A unique representative $\text{CF}_G(x)$ of the orbit $Gx$ of $x$, i.e. $\text{CF}_G(x) = \text{CF}_G(gx)$ for all $g \in G$.

**Transporter Element** A group element $\text{TR}_G(x) := g \in G$ such that $gx = \text{CF}_G(x)$.

**Automorphism Group** The stabilizer $\text{Stab}_G(x) := \{g \in G \mid gx = x\}$ of $x$. 
Introduction

Canonization

Adaption

Refinements

Problem definition

Goal: A canonization algorithm

Let $G$ be a group, which acts on a set $X$. For some arbitrary element $x \in X$ compute:

**Canonical Form** A unique representative $CF_G(x)$ of the orbit $Gx$ of $x$, i.e. $CF_G(x) = CF_G(gx)$ for all $g \in G$

**Transporter Element** A group element $TR_G(x) := g \in G$ such that $gx = CF_G(x)$.

**Automorphism Group** The stabilizer $\text{Stab}_G(x) := \{g \in G \mid gx = x\}$ of $x$. 


Problem definition

Goal: A canonization algorithm

Let $G$ be a group, which acts on a set $X$. For some arbitrary element $x \in X$ compute:

**Canonical Form** A unique representative $\text{CF}_G(x)$ of the orbit $Gx$ of $x$, i.e. $\text{CF}_G(x) = \text{CF}_G(gx)$ for all $g \in G$.

**Transporter Element** A group element $\text{TR}_G(x) := g \in G$ such that $gx = \text{CF}_G(x)$.

**Automorphism Group** The stabilizer $\text{Stab}_G(x) := \{g \in G | gx = x\}$ of $x$. 

Section 1

Introduction
Chain Rings

The (finite, associative) ring \( R \) is a chain ring if the set of left ideals forms a chain:

\[
R \triangleright N \triangleright N^2 \triangleright \ldots \triangleright N^m = \{0_R\}
\]

The maximal ideal \( N := R\theta \) is generated by \( \theta \in R \).

\((N^i = R\theta^i = \theta^i R \text{ and } R/N \cong \mathbb{F}_q.)\)

Examples of chain rings

- finite fields \( \mathbb{F}_{p^r} \)
- integers modulo some prime power \( \mathbb{Z}_{p^r} \)
- Galois rings \( GR(q^r, p^r) \)
Chain Rings

The (finite, associative) ring \( R \) is a chain ring if the set of left ideals forms a chain:

\[
R \triangleright N \triangleright N^2 \triangleright \ldots \triangleright N^m = \{0_R\}
\]

The maximal ideal \( N := R\theta \) is generated by \( \theta \in R \).

\((N^i = R\theta^i = \theta^i R \text{ and } R/N \cong \mathbb{F}_q.)\)

Examples of chain rings

- finite fields \( \mathbb{F}_{p^r} \)
- integers modulo some prime power \( \mathbb{Z}_{p^r} \)
- Galois rings \( GR(q^r, p^r) \)
Chain Rings

The (finite, associative) ring $R$ is a *chain ring* if the set of left ideals forms a chain:

$$R \triangleright N \triangleright N^2 \triangleright \ldots \triangleright N^m = \{0_R\}$$

The maximal ideal $N := R\theta$ is generated by $\theta \in R$.

($N^i = R\theta^i = \theta^i R$ and $R/N \simeq \mathbb{F}_q$.)

Examples of chain rings

- finite fields $\mathbb{F}_{p^r}$
- integers modulo some prime power $\mathbb{Z}_{p^r}$
- Galois rings $GR(q^r, p^r)$
The (finite, associative) ring $R$ is a *chain ring* if the set of left ideals forms a chain:

$$R \triangleright N \triangleright N^2 \triangleright \ldots \triangleright N^m = \{0_R\}$$

The maximal ideal $N := R\theta$ is generated by $\theta \in R$. 
($N^i = R\theta^i = \theta^i R$ and $R/N \simeq \mathbb{F}_q$.)

**Examples of chain rings**

- finite fields $\mathbb{F}_{p^r}$
- integers modulo some prime power $\mathbb{Z}_{p^r}$
- Galois rings $GR(q^r, p^r)$
Chain Rings

The (finite, associative) ring $R$ is a *chain ring* if the set of left ideals forms a chain:

$$R \supset N \supset N^2 \supset \ldots \supset N^m = \{0_R\}$$

The maximal ideal $N := R\theta$ is generated by $\theta \in R$. ($N^i = R\theta^i = \theta^i R$ and $R/N \simeq \mathbb{F}_q$.)

Examples of chain rings

- finite fields $\mathbb{F}_{p^r}$
- integers modulo some prime power $\mathbb{Z}_{p^r}$
- Galois rings $GR(q^r, p^r)$
Linear Codes over $R$

**Linear Code**

A linear code $C$ is a $R$-submodule of $R^n$.

**Shape of a linear code**

There exists a unique sequence of integer $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ with $m \geq \lambda_i \geq \lambda_{i+1} \geq 1$ such that

$$C \simeq R/N^{\lambda_0} \oplus \ldots \oplus R/N^{\lambda_{k-1}}.$$ 

- $\text{shp}(C) := \lambda$ is called the shape of $C$.
- $\text{rk}(C) := k$ is called the rank of $C$. 
Linear Codes over $R$

**Linear Code**

A linear code $C$ is a $R$-submodule of $R^n$.

**Shape of a linear code**

There exists a unique sequence of integer $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ with $m \geq \lambda_i \geq \lambda_{i+1} \geq 1$ such that

$$C \simeq R/N^{\lambda_0} \oplus \ldots \oplus R/N^{\lambda_{k-1}}.$$

- $\text{shp}(C) := \lambda$ is called the shape of $C$.
- $\text{rk}(C) := k$ is called the rank of $C$. 
Linear Codes over $R$

**Linear Code**

A linear code $C$ is a $R$-submodule of $R^n$.

**Shape of a linear code**

There exists a unique sequence of integer $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ with $m \geq \lambda_i \geq \lambda_{i+1} \geq 1$ such that

$$C \simeq R/N^{\lambda_0} \oplus \ldots \oplus R/N^{\lambda_{k-1}}.$$ 

- $\text{shp}(C) := \lambda$ is called the shape of $C$.
- $\text{rk}(C) := k$ is called the rank of $C$. 
## Linear Codes over $R$

### Linear Code

A linear code $C$ is a $R$-submodule of $R^n$.

### Shape of a linear code

There exists a unique sequence of integer $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ with $m \geq \lambda_i \geq \lambda_{i+1} \geq 1$ such that

$$C \cong R/N^{\lambda_0} \oplus \ldots \oplus R/N^{\lambda_{k-1}}.$$

- $\text{shp}(C) := \lambda$ is called the shape of $C$.
- $\text{rk}(C) := k$ is called the rank of $C$. 
Generator matrices

**Generator matrix**

A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,\ast} \simeq R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k-1\}$

**Warning**

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(\mathbb{R})$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(\mathbb{R}) \leq \text{GL}_k(\mathbb{R})$ such that $\text{GL}_\lambda(\mathbb{R})\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_\lambda(\mathbb{R}) = \text{GL}_k(\mathbb{R}) \iff \lambda = (m, \ldots, m)$ (the codes are free)
Generator matrices

**Generator matrix**

A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,*} \cong R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k-1\}$

**Warning**

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(R)$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(R) \leq \text{GL}_k(R)$ such that $\text{GL}_\lambda(R)\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_\lambda(R) = \text{GL}_k(R) \iff \lambda = (m, \ldots, m)$ (the codes are free)
Generator matrices

Generator matrix

A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_i, \ast \cong R/ N_{\lambda_i}$ for all $i \in \{0, \ldots, k - 1\}$

Warning

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(\mathbb{R})$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_{\lambda}(\mathbb{R}) \leq \text{GL}_k(\mathbb{R})$ such that $\text{GL}_{\lambda}(\mathbb{R})\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_{\lambda}(\mathbb{R}) = \text{GL}_k(\mathbb{R}) \iff \lambda = (m, \ldots, m)$ (the codes are free)
**Generator matrices**

**Generator matrix**

A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,\ast} \simeq R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k-1\}$

**Warning**

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(\mathbb{R})$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(\mathbb{R}) \leq \text{GL}_k(\mathbb{R})$ such that $\text{GL}_\lambda(\mathbb{R})\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_\lambda(\mathbb{R}) = \text{GL}_k(\mathbb{R}) \iff \lambda = (m, \ldots, m)$ (the codes are free)
A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,*} \cong R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k-1\}$

**Warning**

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(R)$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(R) \leq \text{GL}_k(R)$ such that $\text{GL}_\lambda(R)\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_\lambda(R) = \text{GL}_k(R) \iff \lambda = (m, \ldots, m)$ (the codes are free)
Generator matrices

A matrix $\Gamma \in R^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if
- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,*} \simeq R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k-1\}$

Warning

Let $\Gamma$ be a generator matrix of $C$.
- For some arbitrary $A \in \text{GL}_k(R)$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(R) \leq \text{GL}_k(R)$ such that $\text{GL}_\lambda(R)\Gamma$ is equal to the set of generator matrices of $C$.
  - $\text{GL}_\lambda(R) = \text{GL}_k(R) \iff \lambda = (m, \ldots, m)$ (the codes are free)
## Generator matrices

### Generator matrix

A matrix $\Gamma \in \mathbb{R}^{k \times n}$ is called a generator matrix of a linear code $C$ with $\text{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k-1})$, if

- rows of $\Gamma$ generate the module $C$
- $R\Gamma_{i,*} \cong R/N^{\lambda_i}$ for all $i \in \{0, \ldots, k - 1\}$

### Warning

Let $\Gamma$ be a generator matrix of $C$.

- For some arbitrary $A \in \text{GL}_k(R)$, the matrix $A\Gamma$ must not be a generator matrix of $C$ (but the rows do generate $C$).
- But, there is a subgroup $\text{GL}_\lambda(R) \leq \text{GL}_k(R)$ such that $\text{GL}_\lambda(R)\Gamma$ is equal to the set of generator matrices of $C$.
- $\text{GL}_\lambda(R) = \text{GL}_k(R) \iff \lambda = (m, \ldots, m)$ (the codes are free)
Isometries

Isometry

We will not specify the distance $d$ defined on $R^n$. But the group of linear isometries defined by $d$ should be equal to the monomial group

$$R^*n \rtimes S_n$$

acting on a vector $v \in R^n$ via

$$(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1})$$

Theorem

If $d(0, xr^{-1}) = d(0, x)$ for all $r \in R^*$, $x \in R$ then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- homogeneous distance
Isometries

Isometry

We will not specify the distance $d$ defined on $R^n$. But the group of linear isometries defined by $d$ should be equal to the monomial group

$$R^*n \rtimes S_n$$

acting on a vector $v \in R^n$ via

$$(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)\varphi^{-1}_0}, \ldots, v_{\pi^{-1}(n-1)\varphi^{-1}_{n-1}})$$

Theorem

If $d(0, xr^{-1}) = d(0, x)$ for all $r \in R^*$, $x \in R$ then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- Homogeneous distance
Isometry

We will not specify the distance $d$ defined on $\mathbb{R}^n$. But the group of linear isometries defined by $d$ should be equal to the monomial group

$$R^* \times S_n$$

acting on a vector $v \in \mathbb{R}^n$ via

$$(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1})$$

Theorem

If $d(0, xr^{-1}) = d(0, x)$ for all $r \in R^*$, $x \in R$ then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- Homogeneous distance
Isometries

We will not specify the distance \( d \) defined on \( \mathbb{R}^n \). But the group of linear isometries defined by \( d \) should be equal to the monomial group

\[
\mathbb{R}^* \ltimes S_n
\]

acting on a vector \( v \in \mathbb{R}^n \) via

\[
(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1})
\]

**Theorem**

If \( d(0, xr^{-1}) = d(0, x) \) for all \( r \in \mathbb{R}^*, x \in \mathbb{R} \) then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- Homogeneous distance
Isometries

Isometry

We will not specify the distance $d$ defined on $R^n$. But the group of linear isometries defined by $d$ should be equal to the monomial group

$$R^{*n} \rtimes S_n$$

acting on a vector $v \in R^n$ via

$$(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1})$$

Theorem

If $d(0, xr^{-1}) = d(0, x)$ for all $r \in R^*$, $x \in R$ then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- homogeneous distance
Isometries

Isometry

We will not specify the distance $d$ defined on $R^n$. But the group of linear isometries defined by $d$ should be equal to the monomial group

$$R^*n \rtimes S_n$$

acting on a vector $v \in R^n$ via

$$(\varphi; \pi)(v_0, \ldots, v_{n-1}) := (v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1})$$

Theorem

If $d(0, xr^{-1}) = d(0, x)$ for all $r \in R^*$, $x \in R$ then the linear isometry group is equal to the monomial group. Examples are:

- Hamming distance
- homogeneous distance
**Convention**

The shape and the rank of a linear code is invariant under the action of the monomial group. Hence, we fix \( k \) and \( \lambda = (\lambda_0, \ldots, \lambda_{k-1}) \) for the rest of the talk.

**Set of all generator matrices**

Define

\[
R^{\lambda \times n} := \{ \Gamma \in R^{k \times n} \mid \text{shp}(R\langle \Gamma \rangle) = \lambda \}
\]

**Identification**

We can identify the linear code \( C \) with the orbit \( \text{GL}_\lambda(R)\Gamma \).

There is a natural bijection

\[
\{ C \mid \text{shp}(C) = \lambda \} \rightarrow R^{\lambda \times n} / \text{GL}_\lambda(R)
\]
Introduction

**Canonization**

**Adaption**

**Refinements**

---

**Generator matrices**

### Convention

The shape and the rank of a linear code is invariant under the action of the monomial group. Hence, we fix $k$ and $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ for the rest of the talk.

### Set of all generator matrices

Define

$$R^{\lambda \times n} := \{ \Gamma \in R^{k \times n} | \text{shp}(R\langle \Gamma \rangle) = \lambda \}$$

### Identification

We can identify the linear code $C$ with the orbit $\text{GL}_\lambda(R)\Gamma$. There is a natural bijection

$$\{ C | \text{shp}(C) = \lambda \} \rightarrow R^{\lambda \times n} / \text{GL}_\lambda(R)$$
**Generator matrices**

**Convention**

The shape and the rank of a linear code is invariant under the action of the monomial group. Hence, we fix $k$ and $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ for the rest of the talk.

**Set of all generator matrices**

Define

$$R^{\lambda \times n} := \{ \Gamma \in R^{k \times n} \mid \text{shp}(R\langle \Gamma \rangle) = \lambda \}$$

**Identification**

We can identify the linear code $C$ with the orbit $\text{GL}_\lambda(R)\Gamma$. There is a natural bijection

$$\{ C \mid \text{shp}(C) = \lambda \} \rightarrow R^{\lambda \times n} / \text{GL}_\lambda(R)$$
A group action

Equivalent generator matrices

We call two generator matrices equivalent, if they generate linearly isometric codes.

In terms of a group action

Two generator matrices $\Gamma, \Gamma' \in R^{\lambda \times n}$ are equivalent, if and only if

$$\exists (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n : (A, \varphi; \pi)\Gamma = \Gamma'$$
A group action

Equivalent generator matrices

We call two generator matrices equivalent, if they generate linearly isometric codes.

In terms of a group action

Two generator matrices $\Gamma, \Gamma' \in R^{\lambda \times n}$ are equivalent, if and only if

\[
\exists (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n : (A, \varphi; \pi)\Gamma = \Gamma'
\]
Problem definition

Goal : A canonization algorithm

Let $\Gamma$ be some arbitrary generator matrix of a linear code of shape $\lambda$. Compute:

- **Canonical Form** A unique representative $\text{CF}(\Gamma)$ of $((\text{GL}_\lambda(R) \times R^*^n) \ltimes S_n) \, \Gamma$

- **Transporter Element** A group element $\text{TR}(\Gamma) := (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \times R^*^n) \ltimes S_n$ such that $(A, \varphi; \pi) \, \Gamma = \text{CF}(\Gamma)$.

- **Automorphism Group** The stabilizer $\text{Stab}_{(\text{GL}_\lambda(R) \times R^*^n) \ltimes S_n}(\Gamma)$ of $\Gamma$. 


Goal: A canonization algorithm

Let Γ be some arbitrary generator matrix of a linear code of shape λ. Compute:

**Canonical Form**  A unique representative CF(Γ) of 
\[ ((\text{GL}_\lambda(R) \rtimes R^n) \rtimes S_n) \, \Gamma \]

**Transporter Element**  A group element 
\[ TR(\Gamma) := (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \rtimes R^n) \rtimes S_n \text{ such that } (A, \varphi; \pi)\Gamma = CF(\Gamma). \]

**Automorphism Group**  The stabilizer \( \text{Stab}_{(\text{GL}_\lambda(R) \rtimes R^n) \rtimes S_n}(\Gamma) \) of \( \Gamma \).
Problem definition

Goal: A canonization algorithm

Let $\Gamma$ be some arbitrary generator matrix of a linear code of shape $\lambda$. Compute:

**Canonical Form** A unique representative $\text{CF}(\Gamma)$ of $((\text{GL}_\lambda(R) \times R^*^n) \rtimes S_n)\Gamma$

**Transporter Element** A group element $\text{TR}(\Gamma):= (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \times R^*^n) \rtimes S_n$ such that $(A, \varphi; \pi)\Gamma = \text{CF}(\Gamma)$.

**Automorphism Group** The stabilizer $\text{Stab}_{(\text{GL}_\lambda(R) \times R^*^n) \rtimes S_n}(\Gamma)$ of $\Gamma$. 

Problem definition

Goal : A canonization algorithm

Let $\Gamma$ be some arbitrary generator matrix of a linear code of shape $\lambda$. Compute:

**Canonical Form** A unique representative $\text{CF}(\Gamma)$ of $((\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n) \Gamma$

**Transporter Element** A group element $\text{TR}(\Gamma) := (A, \varphi; \pi) \in (\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n$ such that $(A, \varphi; \pi)\Gamma = \text{CF}(\Gamma)$.

**Automorphism Group** The stabilizer $\text{Stab}_{(\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n}(\Gamma)$ of $\Gamma$. 
Section 2

General Canonization Algorithms
We use the ideas of partition refinements, similar to the computation of a canonical labeling of a graph (McKay, ...). There is a nice description of this idea for a group action of $G$ on $X$ in

Kaski & Östergård : Classification algorithms for codes and designs

**definitions**

\[ \mathcal{L}(G) := \{ H \mid H \leq G \} , \]
\[ \mathcal{C}(G) := \{ Hg \mid H \leq G, g \in G \} \]

**refinement**

\[ r : X \times \mathcal{C}(G) \to \mathcal{C}(G), (x, Hg) \mapsto H' h g \subseteq H g \text{ with} \]
\[ r(g_0 x, H g g_0^{-1}) = r(x, H g) g_0^{-1} \text{ (G-Homomorphism)} \]

**partitioning**

\[ p : X \times \mathcal{C}(G) \to \mathcal{L}(G), (x, Hg) \mapsto H' \leq H \text{ with} \]
\[ p(g_0 x, H g g_0^{-1}) = p(x, H g) \text{ (G-invariant)} \]
A general solution: partition refinement

We use the ideas of partition refinements, similar to the computation of a canonical labeling of a graph (McKay, . . . ). There is a nice description of this idea for a group action of $G$ on $X$ in

Kaski & Östergård: Classification algorithms for codes and designs

**Definitions**

$L(G) := \{ H \mid H \leq G \}$,

$C(G) := \{ Hg \mid H \leq G, g \in G \}$

**Refinement**

$r : X \times C(G) \to C(G), (x, Hg) \mapsto H' hg \subseteq Hg$ with

$r(g_0 x, Hgg_0^{-1}) = r(x, Hg)g_0^{-1}$ (G-Homomorphism)

**Partitioning**

$p : X \times C(G) \to L(G), (x, Hg) \mapsto H' \leq H$ with

$p(g_0 x, Hgg_0^{-1}) = p(x, Hg)$ (G-invariant)
We use the ideas of partition refinements, similar to the computation of a canonical labeling of a graph (McKay, . . . ). There is a nice description of this idea for a group action of $G$ on $X$ in

Kaski & Östergård : Classification algorithms for codes and designs

**definitions**

- $\mathcal{L}(G) := \{H \mid H \leq G\}$,
- $\mathcal{C}(G) := \{Hg \mid H \leq G, g \in G\}$

**refinement**

- $r : X \times \mathcal{C}(G) \to \mathcal{C}(G)$, $(x, Hg) \mapsto H'hg \subseteq Hg$ with $r(g_0x, Hgg_0^{-1}) = r(x, Hg)g_0^{-1}$ (G-Homomorphism)

**partitioning**

- $p : X \times \mathcal{C}(G) \to \mathcal{L}(G)$, $(x, Hg) \mapsto H' \leq H$ with $p(g_0x, Hgg_0^{-1}) = p(x, Hg)$ (G-invariant)
A general solution: partition refinement

We use the ideas of partition refinements, similar to the computation of a canonical labeling of a graph (McKay, ...). There is a nice description of this idea for a group action of $G$ on $X$ in

Kaski & Östergård : Classification algorithms for codes and designs

**definitions**
\[
\mathcal{L}(G) := \{ H \mid H \leq G \}, \\
\mathcal{C}(G) := \{ Hg \mid H \leq G, g \in G \}
\]

**refinement**
\[
r : X \times \mathcal{C}(G) \to \mathcal{C}(G), (x, Hg) \mapsto H' h g \leq Hg \\
r(g_0 x, H g g_0^{-1}) = r(x, H g) g_0^{-1} \quad (G\text{-Homomorphism})
\]

**partitioning**
\[
p : X \times \mathcal{C}(G) \to \mathcal{L}(G), (x, Hg) \mapsto H' \leq H \\
p(g_0 x, H g g_0^{-1}) = p(x, H g) \quad (G\text{-invariant})
\]
Backtrack tree $T(x, G)$ for input $x \in X$:

$$G$$

$$r(x, G) =: Hg$$
Backtrack tree $T(x, G)$ for input $x \in X$:

$H' := p(x, Hg)$

$\{h_0, h_1, h_2\}$ right transversal of $H'$ in $H$
Backtrack tree $T(x, G)$ for input $x \in X$:

$Hg$ refine $G$

- $H'h_0g$
  - $r(x, H'h_0g)$

- $H'h_1g$
  - $r(x, H'h_1g)$

- $H'h_2g$
  - $r(x, H'h_2g)$
Backtrack tree $T(x, G)$ for input $x \in X$:

Continue until $Hg = \{g\}$
Properties

**Theorem**

*Isomorphic inputs define isomorphic search trees.*

\[
G_r(x, G) \quad H_{g} \quad H'_{g}
\]

\[
G_r(x, G)g^{-1} \quad H_{g}g^{-1} \quad H'_{hgg^{-1}} \quad \{\bar{g}g^{-1}\}
\]

\[
T(x, G) \quad T(g_0x, G)
\]
Canonical Form

Define the canonical form by

\[ CF(x) = \min_{\{g\} \text{ leaf in } T(x, G)} g^x \]

Transporter Element

The transporter element \( g \) by one of those leafs \( \{g\} \) leading to the canonical form \( g^x = CF(x) \).

Automorphism Group

The automorphism group is given by

\[ Stab_G(x) = \{ TR(x)^{-1} g \mid \{g\} \text{ leaf in } T(x, G) \text{ and } g^x = CF(x) \} \]
Canonical Form

Define the canonical form by

\[ \text{CF}(x) = \min_{\{g\} \text{ leaf in } T(x, G)} g^x \]

\[ = \min_{\{gg^{-1}\} \text{ leaf in } T(g_0 x, G)} gg_0^{-1} g_0 x = \text{CF}(g_0 x) \]

Transporter Element

The transporter element \( g \) by one of those leaves \( \{g\} \) leading to the canonical form \( g^x = \text{CF}(x) \).

Automorphism Group

The automorphism group is given by

\[ \text{Stab}_G(x) = \{ \text{TR}(x)^{-1} g \mid \{g\} \text{ leaf in } T(x, G) \text{ and } g^x = \text{CF}(x) \} \]
**Canonical Form**

Define the *canonical form* by

\[ CF(x) = \min_{\{g\} \text{ leaf in } T(x, G)} g_x \]

**Transporter Element**

The *transporter element* \( g \) by one of those leafs \( \{g\} \) leading to the canonical form \( g_x = CF(x) \).

**Automorphism Group**

The *automorphism group* is given by

\[ Stab_G(x) = \{ TR(x)^{-1} g | \{g\} \text{ leaf in } T(x, G) \text{ and } g_x = CF(x) \} \]
Define the canonical form by

$$\text{CF}(x) = \min_{\{g\} \text{ leaf in } T(x, G)} g^x$$

The transporter element $g$ by one of those leaves $\{g\}$ leading to the canonical form $g^x = \text{CF}(x)$.

The automorphism group is given by

$$\text{Stab}_G(x) = \{\text{TR}(x)^{-1}g \mid \{g\} \text{ leaf in } T(x, G) \text{ and } g^x = \text{CF}(x)\}$$
Pruning

by known automorphisms

Use the subgroup $A \leq \text{Stab}_G(x)$ of known automorphisms to define pruning mechanisms \( \sim \) traverse the tree in a depth-first search manner.

by refinements

- Let $f_H : X \rightarrow Y$ be an $H$-Homomorphism
- $r(x, Hg) := \text{Stab}_H(\text{CF}_H(f_H(gx))) \cdot \text{TR}_H(f_H(gx))$
- $Hg_1, Hg_2$ two nodes in $T(x, G)$ with $\text{CF}_H(f_H(g_1x)) < \text{CF}_H(f_H(g_2x))$
  \( \Rightarrow \) prune the subtree rooted in $Hg_2$ (Homomorphism Principle)
Pruning

**by known automorphisms**

Use the subgroup $A \leq \text{Stab}_G(x)$ of known automorphisms to define pruning mechanisms $\leadsto$ traverse the tree in a depth-first search manner.

**by refinements**

- Let $f_H : X \rightarrow Y$ be an $H$-Homomorphism.
- $r(x, Hg) := \text{Stab}_H(\text{CF}_H(f_H(gx))) \cdot \text{TR}_H(f_H(gx))$
- $Hg_1, Hg_2$ two nodes in $T(x, G)$ with $\text{CF}_H(f_H(g_1x)) < \text{CF}_H(f_H(g_2x))$
  $\implies$ prune the subtree rooted in $Hg_2$ (Homomorphism Principle)
## Pruning

### by known automorphisms

Use the subgroup $A \leq \text{Stab}_G(x)$ of known automorphisms to define pruning mechanisms $\rightsquigarrow$ traverse the tree in a depth-first search manner.

### by refinements

Let $f_H : X \rightarrow Y$ be an $H$-Homomorphism

$r(x, Hg) := \text{Stab}_H(\text{CF}_H(f_H(gx))) \cdot \text{TR}_H(f_H(gx))$

$Hg_1, Hg_2$ two nodes in $T(x, G)$ with $\text{CF}_H(f_H(g_1x)) < \text{CF}_H(f_H(g_2x))$

$\Rightarrow$ prune the subtree rooted in $Hg_2$ (Homomorphism Principle)
Section 3

Canonization of linear codes
The algorithm for linear codes

An equivalent group action

Instead of

\[(\text{GL}_\lambda(R) \times R^{*n}) \rtimes S_n \]

=: G

acting on \( R^{\lambda \times n} \) we will investigate the group action of \( S_n \) on the set of orbits \( R^{\lambda \times n}/G := \{ G\Gamma \mid \Gamma \in R^{\lambda \times n} \} \).

Motivation

- We can efficiently compute canonical representatives for the orbits \( G\Gamma \).
- Permutation groups are much simpler.
- The algorithm is well-studied for the action of the symmetric group. (efficient data types & prunings)
The algorithm for linear codes

An equivalent group action

Instead of

\[
\left( \text{GL}_\lambda(R) \times R^{\ast n} \right) \rtimes S_n
\]

acting on \( R^{\lambda \times n} \) we will investigate the group action of \( S_n \) on the set of orbits \( R^{\lambda \times n}/G := \{ G\Gamma \mid \Gamma \in R^{\lambda \times n} \} \).

Motivation

- We can efficiently compute canonical representatives for the orbits \( G\Gamma \).
- Permutation groups are much simpler.
- The algorithm is well-studied for the action of the symmetric group. (efficient data types & prunings)
The algorithm for linear codes

An equivalent group action

Instead of

\[
\left( \text{GL}_\lambda(R) \times R^{\ast n} \right) \rtimes S_n =: G
\]

acting on \( R^{\lambda \times n} \) we will investigate the group action of \( S_n \) on the set of orbits \( R^{\lambda \times n}/G := \{ G\Gamma \mid \Gamma \in R^{\lambda \times n} \} \).

Motivation

- We can efficiently compute canonical representatives for the orbits \( G\Gamma \).
- Permutation groups are much simpler.
- The algorithm is well-studied for the action of the symmetric group. (efficient data types & prunings)
The algorithm for linear codes

An equivalent group action

Instead of

\[
\underbrace{\left(\text{GL}_\lambda(R) \times R^{\times n}\right)}_{=: G} \rtimes S_n
\]

acting on \(R^{\lambda \times n}\) we will investigate the group action of \(S_n\) on the set of orbits \(R^{\lambda \times n}/G := \{G\Gamma \mid \Gamma \in R^{\lambda \times n}\}\).

Motivation

- We can efficiently compute canonical representatives for the orbits \(G\Gamma\).
- Permutation groups are much simpler.
- The algorithm is well-studied for the action of the symmetric group. (efficient data types & prunings)
Section 4

A selection of important refinements
Refinements by words of given symmetrized weight (Leon)

Draw colored edges depending on the maximal Ideal containing $c_i = u\Gamma_i$.

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m - 1)$$
Refinements by words of given symmetrized weight (Leon)

Draw colored edges depending on the maximal ideal containing $c_i = u\Gamma_i$.

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m - 1)$$
Refinements by words of given symmetrized weight (Leon)

Distinguish nodes by the number of neighbors of some fixed color connected by an edge of fixed color.

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m)$$

$$w \in R^k : \text{shp}(\langle w \rangle) = (m - 1)$$
Refinements by words of given symmetrized weight (Leon)

Continue process until stable (relabellings might be necessary).

\[ w \in R^k : \text{shp}(\langle w \rangle) = (m) \]

\[ w \in R^k : \text{shp}(\langle w \rangle) = (m) \]

\[ w \in R^k : \text{shp}(\langle w \rangle) = (m - 1) \]
Refinements by words of given symmetrized weight (Leon)

Draw colored edges depending on the maximal Ideal containing $c_i = u \Gamma_i$.

Distinguish nodes by the number of neighbors of some fixed color connected by an edge of fixed color.

Continue process until stable (relabellings might be necessary).

Interpret new coloring as refinement.

$w \in R^k : \text{shp}(\langle w \rangle) = (m)$

$w \in R^k : \text{shp}(\langle w \rangle) = (m)$

$w \in R^k : \text{shp}(\langle w \rangle) = (m - 1)$
A second „refinement“

**Preparation**

For each occurring node $S_p \pi$ in $T(G, \Gamma, S_n)$ choose an injective sequence $F = F(S_p, \pi \Gamma) \subseteq \text{Fix}_{S_p}(\{0, \ldots, n - 1\})$.

**An invariant for pruning subtrees**

At the node $S_p \pi$ of $T(G, \Gamma, S_n)$, compute

$$f_{S_p}(\pi \Gamma) := \text{CF}_G \left( (\Gamma_*, \pi^{-1}(i))_{i \in F} \right)$$

and use the result to potentially prune the subtree rooted in this node. (There will be no refinement of $p$.)
A second „refinement“

Preparation
For each occurring node $s_{p \pi}$ in $T(G\Gamma, S_n)$ choose an injective sequence $F = F(s_{p \pi}, \pi \Gamma) \subseteq \text{Fix}_{s_p} \{0, \ldots, n - 1\}$.

An invariant for pruning subtrees
At the node $s_{p \pi}$ of $T(G\Gamma, S_n)$, compute

$$f_{s_p}(\pi \Gamma) := \text{CF}_G \left( (\Gamma \ast, \pi^{-1}(i))_{i \in F} \right)$$

and use the result to potentially prune the subtree rooted in this node. (There will be no refinement of $p$.)
How can you use it!

Finite fields, $\mathbb{Z}_4$, $\mathbb{F}_2[x]/(x^2)$

An implementation in C++ and an online calculator is available at [http://codes.uni-bayreuth.de/CanonicalForm/index.html](http://codes.uni-bayreuth.de/CanonicalForm/index.html)

Sage

- Finite Fields: [Ticket 13771](http://www.sage.math.washington.edu/trac/ticket/13771) (Reviewers wanted!)
- Finite Chain Rings: hopefully soon available!

Network Codes & $\mathbb{F}_q$-linear codes over $\mathbb{F}_{q^r}$

An implementation in C++ exists. → Write me an email.