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 $\mathrm{CF}_G(x)$ of the orbit Gx of x, i.e. $\mathrm{CF}_G(x) = \mathrm{CF}_G(gx)$ for all $g \in G$

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

Automorphism Group The stabilizer $\mathrm{Stab}_G(x) := \{g \in G \mid gx = x\}$ 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 $\mathrm{CF}_G(x)$ of the orbit Gx of x, i.e. $\mathrm{CF}_G(x) = \mathrm{CF}_G(gx)$ for all $g \in G$

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

Automorphism Group The stabilizer $\mathrm{Stab}_G(x) := \{g \in G \mid gx = x\}$ 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 $\mathrm{CF}_G(x)$ of the orbit Gx of x, i.e. $\mathrm{CF}_G(x) = \mathrm{CF}_G(gx)$ for all $g \in G$

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

Automorphism Group
: The stabilizer $\mathrm{Stab}_G(x) := \{g \in G \mid gx = x\}$ 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 $\mathrm{CF}_G(x)$ of the orbit Gx of x, i.e. $\mathrm{CF}_G(x) = \mathrm{CF}_G(gx)$ for all $g \in G$

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

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

Section 1

Introduction

Chain Rings

Chain Rings

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

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

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

Chain Rings

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

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

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

Chain Rings

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

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

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

Chain Rings

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

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

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

Chain Rings

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

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

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

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

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

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

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

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

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

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

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

Generator matrices

Generator matrix

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

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

Warning

Let Γ be a generator matrix of C.

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

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

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

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

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

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

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*

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*

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

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

Identification

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

$$\{C \mid \mathrm{shp}(C) = \lambda\} \to R^{\lambda \times n} / \mathrm{GL}_\lambda(R)$$

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

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

Identification

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

$$\{C \mid \mathrm{shp}(C) = \lambda\} \to R^{\lambda \times n} / \mathrm{GL}_\lambda(R)$$

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

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

Identification

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

$$\{C \mid \mathrm{shp}(C) = \lambda\} \to R^{\lambda \times n}/\,\mathrm{GL}_\lambda(R)$$

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

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

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

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

Problem definition

Goal : A canonization algorithm

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

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

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

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

Problem definition

Goal : A canonization algorithm

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

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

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

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

Problem definition

Goal : A canonization algorithm

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

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

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

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

Problem definition

Goal : A canonization algorithm

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

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

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

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

Section 2

General Canonization Algorithms

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

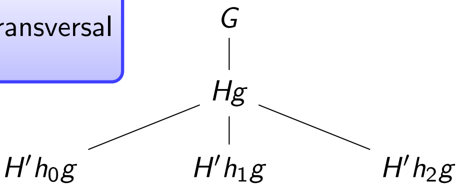
Backtrack tree $T(x, G)$ for input $x \in X$:

$$\begin{array}{c} G \\ | \\ r(x, G) =: Hg \end{array}$$

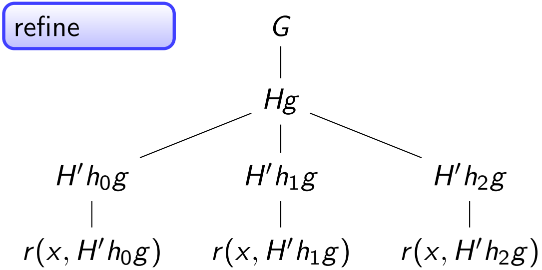
Backtrack tree $T(x, G)$ for input $x \in X$:

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

$\{h_0, h_1, h_2\}$ right transversal of H' in H

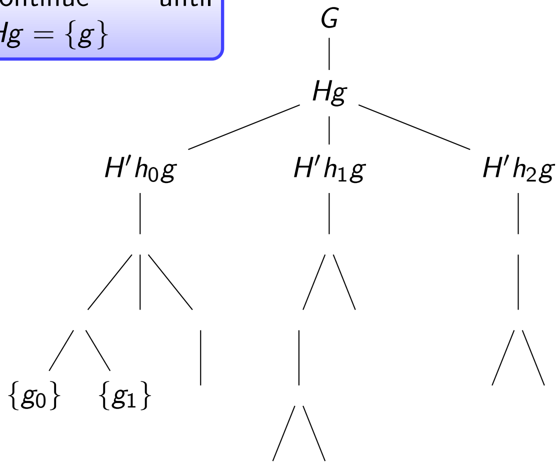


Backtrack tree $T(x, G)$ for input $x \in X$:



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

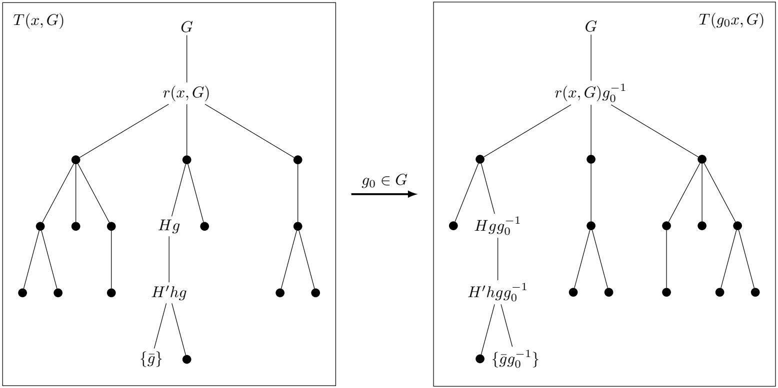
continue until $Hg = \{g\}$



Properties

Theorem

Isomorphic inputs define isomorphic search trees.



Canonical Form

Define the canonical form by

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

Transporter Element

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

Automorphism Group

The automorphism group is given by

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

Canonical Form

Define the canonical form by

$$\begin{aligned}\mathrm{CF}(x) &= \min_{\{g\} \text{ leaf in } T(x,G)} gx \\ &= \min_{\{gg_0^{-1}\} \text{ leaf in } T(g_0x,G)} gg_0^{-1}g_0x = \mathrm{CF}(g_0x)\end{aligned}$$

Transporter Element

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

Automorphism Group

The automorphism group is given by

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

Canonical Form

Define the canonical form by

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

Transporter Element

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

Automorphism Group

The automorphism group is given by

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

Canonical Form

Define the canonical form by

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

Transporter Element

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

Automorphism Group

The automorphism group is given by

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

Pruning

by known automorphisms

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

by refinements

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

Pruning

by known automorphisms

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

by refinements

- Let $f_H : X \to Y$ be an H-Homomorphism
- $r(x, Hg) := \mathrm{Stab}_H(\mathrm{CF}_H(f_H(gx))) \cdot \mathrm{TR}_H(f_H(gx))$
- Hg_1, Hg_2 two nodes in $T(x, G)$ with $\mathrm{CF}_H(f_H(g_1 x)) < \mathrm{CF}_H(f_H(g_2 x))$
  $\Longrightarrow$ prune the subtree rooted in Hg_2 (Homomorphism Principle)

Pruning

by known automorphisms

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

by refinements

- Let $f_H : X \to Y$ be an H-Homomorphism
- $r(x, Hg) := \mathrm{Stab}_H(\mathrm{CF}_H(f_H(gx))) \cdot \mathrm{TR}_H(f_H(gx))$
- Hg_1, Hg_2 two nodes in $T(x, G)$ with $\mathrm{CF}_H(f_H(g_1x)) < \mathrm{CF}_H(f_H(g_2x))$
  $\Longrightarrow$ 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

$$\underbrace{(\mathrm{GL}_\lambda(R) \times R^{*n})}_{=:G} \rtimes S_n$$

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)

The algorithm for linear codes

An equivalent group action

Instead of

$$\underbrace{(\mathrm{GL}_\lambda(R) \times R^{*n})}_{=:G} \rtimes S_n$$

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)

The algorithm for linear codes

An equivalent group action

Instead of

$$\underbrace{(\mathrm{GL}_\lambda(R) \times R^{*n})}_{=:G} \rtimes S_n$$

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)

The algorithm for linear codes

An equivalent group action

Instead of

$$\underbrace{(\mathrm{GL}_\lambda(R) \times R^{*n})}_{=:G} \rtimes S_n$$

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

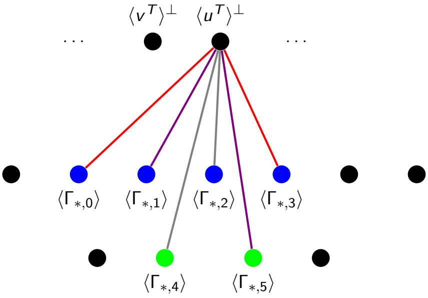
- We can efficiently compute canonical representatives for the orbits $G\Gamma$.
- Permutation groups are much simpler.
- The algorithm is well-studied for the action of the symmetric group. (efficient data types & prunings)

Section 4

A selection of important refinements

Refinements by words of given symmetrized weight (Leon)

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



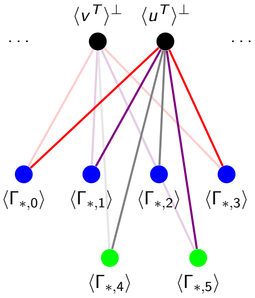
$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m)$

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

$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m-1)$

Refinements by words of given symmetrized weight (Leon)

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

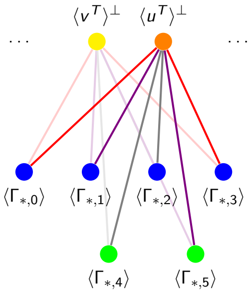


$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m)$

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

$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m-1)$

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.

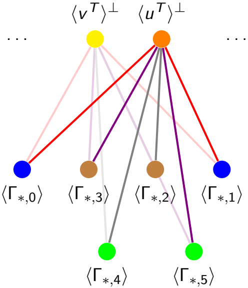
$$w \in R^k : \mathrm{shp}(\langle w\rangle) = (m)$$

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

$$w \in R^k : \mathrm{shp}(\langle w\rangle) = (m-1)$$

Refinements by words of given symmetrized weight (Leon)

Continue process until stable (relabellings might be necessary).



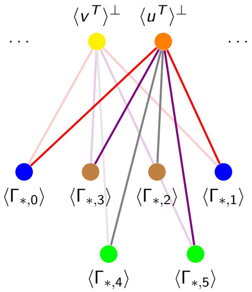
$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m)$

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

$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m-1)$

Refinements by words of given symmetrized weight (Leon)

Interpret new coloring as refinement.



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

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

$w \in R^k : \mathrm{shp}(\langle w \rangle) = (m-1)$

A second „refinement“

Preparation

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

An invariant for pruning subtrees

At the node $S_{\mathfrak{p}}\pi$ of $T(G\Gamma, S_n)$, compute

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

and use the result to potentially prune the subtree rooted in this node. (There will be no refinement of $\mathfrak{p}$.)

A second „refinement“

Preparation

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

An invariant for pruning subtrees

At the node $S_{\mathfrak{p}}\pi$ of $T(G\Gamma, S_n)$, compute

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

and use the result to potentially prune the subtree rooted in this node. (There will be no refinement of $\mathfrak{p}$.)

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`

Sage

- Finite Fields: 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. $\rightarrow$ Write me an email.