Chapter 1

Invariant Theory

This chapter is co-authored by Francesca Gandini, Sumner Strom, Al Ashir Intisar

1.1 Invariant Rings Theory

1.1.1 Finite Matrix Groups

Example: Consider

$$M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and the vector $\bar{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ This gives $M\bar{x} = \begin{bmatrix} x \\ -y \end{bmatrix}$. Thus for the polynomial

$$f(\bar{x}) = f(\begin{bmatrix} x \\ y \end{bmatrix}) = x + y$$

and we have,

$$f(M\bar{x}) = f(\begin{bmatrix} x \\ -y \end{bmatrix}) = x - y.$$

Definition 1.1.1 $G \leq GL_m(\mathbb{K}), |G| < \infty$, then G is a finite matrix group. $\Diamond$

NOTE: An action of a finite group $G \curvearrowright \mathbb{K}^n$ given a realization of G as a finite matrix group.

Example:

$$\langle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \rangle = \{ \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \} \cong C_2$$

1.1.2 Invariant Rings

Notation $\bar{x} = (x_1, x_2, ..., x_n)$, with $R = \mathbb{K}[x_1, x_2, ..., x_n]$

Definition 1.1.2 G is a finite matrix group within $GL_m(\mathbb{K})$ when $f \in \mathbb{K}[x_1, x_2, ..., x_n]$ is invariant under the action of G if and only if $f(A\bar{x}) = f(\bar{x})$, $\forall A \in G$. $\Diamond$

Ex. $f(\bar{x}) = x$ and $f(\bar{x}) = x + y^2$ in $\mathbb{K}[x_1, x_2, ..., x_n]$ is invariant under

$$C_2 = \langle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \rangle$$

However $f(\bar{x}) = x + y$ is not.

Definition 1.1.3 $R^G := \{f \in R \,|\, f(A\bar{x}) = f(\bar{x}), \forall A \in G\} \subseteq R$ is the invariant ring for the action of G $\diamondsuit$

1.1.3 Reynolds Operator

Idea: "Averaging" over the action of G we get an invariant

Definition 1.1.4 $R_G : R \to R^G$

$$R_G(f) = \frac{1}{|G|} \sum_{A \in G} f(A\bar{x})$$

$\diamondsuit$

Example for the Group action $C_2 = \langle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \rangle$:

$$R_G(x + y) = \frac{1}{2}((x + y) + (x - y)) = x \in R^G$$

1.1.4 Nöether Degree Bound(NDB)

Theorem 1.1.5 *(Noether):*

$$R^G = \mathbb{K}[R_G(\bar{x}^{\bar{\beta}}) |\, |\bar{\beta}| \leq |G|]$$

$\implies$ *NDB : The ring of invariants is generated in degrees* $\leq |G|$

Note: This is a computational tool! We can apply R_G to all the finitely many monomials in degrees $\leq |G|$ to get generators for R^G.

1.1.5 Hilbert Ideal

Note: In general for $\{f_1, ..., f_s\} \subseteq \mathbb{R}$, $\{f_1, ...f_s\}$ and $\mathbb{R}$ can be quite different objects

Theorem 1.1.6 *Let* $J_G = R(R^G)_t$*, ideal generated by all positive degree invariants. If* $J_G = (f_1, ..., f_s)$ *and* $f_i \in R^G$, $\forall i$ *(apply* R^G *if it is not), then* $R^G = \mathbb{K}[f_1, ...f_s]$

1.1.6 Presentations

Definition 1.1.7 Definition: Let $S = \mathbb{K}[f_1, ...f_s] \subset R$. A presentation of S is a map,

$$T =: \mathbb{K}[u_1, ...u_s] \xrightarrow{\phi} S$$

such that $\frac{T}{\ker(\phi)} \cong S$ With the syzygies of f_i's giving the presentation ideal. $\diamondsuit$

Proposition 1.1.8 *(Elimination Theory): In* $S \bigotimes \mathbb{K}[u_1, ..., u_s] = \mathbb{K}[x_1, ..., x_n, u_1, ...u_s]$ *consider the ideal,*

$$I = (u_i - f_x(\bar{x}) |\, \langle f_i \rangle = S$$

Then,

$$ker(\phi) = I \cap \mathbb{K}[u_1, ..., u_s]$$

Algorithm 1.1.9 *Compute a Groebner Basis G for I with elimination order for the x's. Then, $G \cap \mathbb{K}[y_1, ...y_s]$ is the Groebner Basis for $ker\phi$*

1.1.7 Graph of Linear Actions

Definition 1.1.10 Let $G \leq GL_n(\mathbb{K})$, $G \curvearrowright \mathbb{K}^n =: V$, $|G|\infty$. For $A \in G$ consider,

$$V_A = \{(\bar{v}, A\bar{v})|\ v \in V\} \subseteq V \bigotimes V$$

Then $A_G = \cup_{A \in G} V_A$ is the subspace arrangement associated to the action of G. $\Diamond$

Note: V_A is a linear subspace, $\mathbb{I}(V_A) :=$ set of polynomials vanishing on $\mathbb{V}_A$ is a linear ideal. Example:

$$V_{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}} = \{(x_1, x_2, x_1, -x_2)\} = V(y_1, -x_1, y_2 + x_2)$$

1.1.8 Subspace Arrangement Approach

Theorem 1.1.11 *(Dekseu): Let $I_G = \mathbb{I}(A_G) = \cap_{A \in G} \mathbb{I}(V_A) \subseteq \mathbb{K}[x_1, ...x_n, y_1, ...y_n]$. Then*

$$(I_G + (y_1, ..., y_n)) \cap \mathbb{R} = J_G$$

This uses elimination theory and the Hilbert ideal.

Note: The same approach works in the exterior algebra!

Theorem 1.1.12 *Let $I_G^{'} = \cap_{A \in G} \mathbb{I}(V_A) \subseteq \Lambda(\bar{x}, \bar{y})$. Then*

$$(I_G^{'} + (y_1, ...y_n)) \cap \Lambda(x_1, ..., x_n) = J_G^{'} := \Lambda(\bar{x})(\Lambda(\bar{x})^G)_+$$

Note: This approach is slow for polynomials, but might be fast for skew polynomials.

1.1.9 Abelian GPS and Weight Matrices

Let $G \cong \mathbb{Z}_d, \bigoplus .... \bigoplus \mathbb{Z}_{dr}, \quad d_i | d_{i+1}$ for $1 \leq i \leq r - 1$

$$\langle g_1 \rangle \bigoplus ... \bigoplus \langle g_r \rangle, \quad |g_i| = d_i$$

A diagonal action of G on R is given by

$$g_i \cdot x_j = \mu_i^{\omega ij} x_j$$

for $\mu_i : d_i^{th}$ primitive root of unity and $i \in [x]$,$j \in [n]$. And encoded in the weight matrix $W = (\omega_{ij})_{ij} = \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \\ x_n & & \end{bmatrix}$

Theorem 1.1.13 $\bar{x}^{\bar{\beta}} \in R^G \iff W_{\bar{\beta}} \cong (0, ..., 0)$ *for zeros being the weight of g_1 acting on $\bar{x}^{\bar{\beta}}$ and being modulo d_i.*

Note: We can examine all monomials $|\bar{\beta}| \leq |G|$ and sort them by their weight $W\bar{\beta}$. The ones with weight $\bar{0}$ will be invariant!

1.1.10 Orbit Sums

Say the symmetric group Σ_n acts on $\{1, ..., n\}$ by permuting its elements. Then the representation of Σ_n is $V = \mathbb{F}^n$ with a set of basis vectors $\{e_1, ..., e_n\}$. This means that Σ_n acts on V by permuting its basis vectors, $\{\sigma(e_1), ..., \sigma(e_n)\}$ we have a permutation representation.

Example of left acting matrix on the basis:

$$(1\,2\,3\,4) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

and then we have it acting,

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} v_4 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

These are useful tools for calculating invariants because we simplify to Linear Algebra! An example of invariants are symmetric polynomials which we can use permutation representations on.

Definition 1.1.14 Symmetric Polynomial: For $f \in R[x_1, x_2, ..., x_n]$ a polynomial is a Symmetric Polynomial if $f(x_1, x_2, ..., x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$ for all permutations of $\sigma \in S_n$ $\Diamond$

Definition 1.1.15 Elementary Symmetric Polynomials: $e_0, e_1, ..., e_n$ in $R[x_1, ..., x_n]$ are defined by

$$e_m = \sum x_{j_1} x_{j_2} ... x_{j_m}$$

$\Diamond$

We now introduce some tools for calculating properties of these permutation groups. Such as Orbit sums and Special Monomials.

Definition 1.1.16 Orbit Sums are the sum of all orbit elements. An orbit is A $G - orbit$ for the left acting group on an element x_0 is

$$G - orbit = \{gx_0 | g \in G\}$$

$\Diamond$

Proposition 1.1.17 *The orbit sums of any given degree d form a basis for the vector space* $\mathbb{F}(V)^G$

Definition 1.1.18 A monomial is special within $\mathbb{F}[x_1, ..., x_n]$ if the partitions satisfy $x_1^n x_2^{n-1} ....x_n^0$. $\Diamond$

Statement: $x_1^n x_2^{n-1} ....x_n^0$ would not be special within $\mathbb{F}[x_1, ..., x_{n-1}]$.

Algorithmically, we can reduce any monomial to special by reducing the upper degrees repeatedly until the monomial is special. Example: within $\mathbb{F}[x_1, x_2, x_3]$

$$x_1^4 x_2^2 x_3 \mapsto x_1^2 x_2$$

This all leads to a theorem.

Theorem 1.1.19 *(Göbel): Let* $\phi : G \mapsto GL(n, \mathbb{F})$ *be a permutation representation of a finite group for* $\mathbb{F} = \mathbb{F}[x_1, ..., x_n]$. *Then the ring of invariants* $\mathbb{F}[V]^G$ *is generated as an algebra by the top elementary symmetric function* $s_n = x_1 ... x_n$

and the orbit sum of monomials.

This theorem is a possibly important tool for reducing computational need to generate these algebra.

1.2 InvariantRings package

1.3 InvariantRing Library Demos

The InvariantRing package in Macaulay2 provides tools to study and compute invariant rings of group actions. To get started, install the package:

```
installPackage "InvariantRing"
```

1.4 SL Actions on C^2 and Variants

A classical example: the standard action of SL on ^{2}. The ring R carries a linearly reductive action from SL defined via the matrix SL2std. The invariants and Hilbert ideal are then computed:

```
restart
needsPackage "InvariantRing"
B = QQ[a,b,c,d]
A = ideal(a*d - b*c - 1)
SL2std = matrix{{a,b},{c,d}}
R = QQ[x_1..x_2]
V = linearlyReductiveAction(A,SL2std,R)
invariants V
elapsedTime hilbertIdeal V
```

1.5 Diagonal Actions of Abelian Groups

This example demonstrates a diagonal action of the abelian group $\times$ on a polynomial ring. After defining the diagonal weights, we compute the invariant ring and its Hilbert series:

```
restart
needsPackage "InvariantRing"
R = QQ[x_1..x_3]
W = matrix{{1,0,1},{0,1,1}}
L = {3,3}
T = diagonalAction(W,L,R)
S = R^T
invariantRing T
I = definingIdeal S
Q = ring I
F = res I
hilbertSeries S
equivariantHilbertSeries T
```

1.6 Linearly Reductive Actions: Permutations and Binary Forms

Here's how the symmetric group S acts via a matrix of projection operators. We identify which polynomials are invariant under the group action:

```
restart
needsPackage "InvariantRing"
S = QQ[z]
A = ideal(z^2 - 1)
M = matrix{{(1+z)/2, (1-z)/2},{(1-z)/2,(1+z)/2}}
R = QQ[a,b]
X = linearlyReductiveAction(A,M,R)
isInvariant(a,X)
invariants X
```

Now we compute the invariants of binary quadratics and quartics using SL actions. These involve basis substitutions in a ring of forms and are more computationally demanding:

```
restart
needsPackage "InvariantRing"
S = QQ[a,b,c,d]
I = ideal(a*d - b*c - 1)
A = S[u,v]
M = transpose (map(S,A)) last coefficients
    sub(basis(2,A),{u=>a*u+b*v,v=>c*u+d*v})
R = QQ[x_1..x_3]
L = linearlyReductiveAction(I,M,R)
hilbertIdeal L
invariants L
invariants(L,4)
invariants(L,5)
```

```
restart
needsPackage "InvariantRing"
S = QQ[a,b,c,d]
I = ideal(a*d - b*c - 1)
A = S[u,v]
M4 = transpose (map(S,A)) last coefficients
    sub(basis(4,A),{u=>a*u+b*v,v=>c*u+d*v})
R4 = QQ[x_1..x_5]
L4 = linearlyReductiveAction(I,M4,R4)
elapsedTime hilbertIdeal L4
elapsedTime X = invariants L4
g2 = X_0/12
g3 = -X_1/216
256*(g2^3 - 27*g3^2)
1728*(g2^3)/(g2^3 - 27*g3^2)
```

1.7 Matrix Invariants and Conjugation Actions

We define SL actions on 2×2 and 3×3 matrices of binary or ternary forms. The conjugation action creates sophisticated invariants under change of basis:

```
restart
needsPackage "InvariantRing"
S = QQ[g_(1,1)..g_(2,2),t]
I = ideal((det genericMatrix(S,2,2))*t-1)
Q = S/I
A = Q[y_(1,1)..y_(2,2)]
Y = transpose genericMatrix(A,2,2)
g = promote(genericMatrix(S,2,2),A)
G = reshape(A^1,A^4,g*Y*inverse(g)) // (vars A)
G = lift(map(A^4,A^4,G),S)
R = QQ[x_(1,1)..x_(2,2)]
L = linearlyReductiveAction(I,G,R)
elapsedTime H=hilbertIdeal(L)
elapsedTime invariants L
```

The same process is repeated for 3×3 matrices. This involves 9-dimensional vector spaces and is more computationally demanding:

```
restart
needsPackage "InvariantRing"
S = QQ[g_(1,1)..g_(3,3),t]
I = ideal((det genericMatrix(S,3,3))*t-1)
Q = S/I
A = Q[y_(1,1)..y_(3,3)]
Y = transpose genericMatrix(A,3,3)
g = promote(genericMatrix(S,3,3),A)
G = reshape(A^1,A^9,g*Y*inverse(g)) // (vars A)
G = lift(map(A^9,A^9,G),S)
R = QQ[x_(1,1)..x_(3,3)]
L = linearlyReductiveAction(I,G,R)
elapsedTime H=hilbertIdeal(L)
elapsedTime invariants(L,1)
elapsedTime invariants(L,2)
elapsedTime invariants(L,3)
```

1.8 Finite Group Actions: S Example

Finally, we examine the symmetric group S acting on 4 variables. We use both King's algorithm and a slower linear algebra method to compute primary and secondary invariants:

```
restart
needsPackage "InvariantRing"
R = QQ[x_1..x_4]
L = apply({[2,1,3,4],[2,3,4,1]},permutationMatrix);
S4 = finiteAction(L,R)
elapsedTime invariants S4
elapsedTime invariants(S4,Strategy=>"LinearAlgebra")
elapsedTime p=primaryInvariants S4
elapsedTime secondaryInvariants(p,S4)
elapsedTime hironakaDecomposition(S4)
```