Moscow Institute of Physics and Technology (MIPT)

Department of Discrete Mathematics

Master Thesis

Theme:
Weak Faddeev-Takhtajan-Volkov algebras; Lattice W_n-algebras

Author:
[1] Farrokh Razavinia[*1]
[1] f.razavinia@phystech.edu
Supervisor:
[2] Professor Alexei Kanel-Belov[†2]
[2] kanelster@gmail.com

[3][1] [4] [5]

[*] Farrokh Razavinia

[†] Professor Alexei Kanel-Belov

[3] The author would like to thank Professor Yaroslav Pugai for his helpful discussion during the preparation for this thesis.

[4] subjclass[2000]: Primary 16D10. 17B37. 81R50. ; Secondary 16W30. 20G42.

[5] keywords: Lattice W algebras, quantum groups, Feigin's homomorphisms, Boris Feigin, Yaroslav Pugai.

Contents

Abstract

In this thesis, we will introduce a new and a weaker version of the famous Faddeev-Takhtajan-Volkov algebra in Sl_2 case and a complete calculation toward it by getting help of a new defined Poisson bracket just by using the Cartan matrix A_2 and then by employing this structure we will extend it to the Sl_3 case and so on to Sl_n case by using the Cartan matrix A_n.

These newly defined structures will also help us to define Lattice W_n- algebras as an extension of Lattice Virasoro algebras in a very simple way which has known till now by employing Feigin's homomorphisms and screening operators and also our newly defined Poisson brackets; just based on the generators of the invariant space of the nilpotent part of $U_q(Sl_n)$.
Also, our approach gives an efficient way of solving system of partial differential equations in higher dimension with coefficients as functions consist of n independent variables $X_1, \cdots, X_n$ such that do not contain the unknown function f as a solution, by transferring some results and ideas from "Chapter V, Section IV of Goursat's Differential Equations" in the language of Mathematica.

6

1. Introduction and some historical flashback

This is an old project which has been considered and introduced by Boris Feigin in 1992. It has born in its new formulation on quantum Gelfand-Kirillov conjecture in a public conference at RIMS in 1992 based on the nilpotent part of $U_q(g)$ i.e. $U_q(\mathfrak{n})$ for g a simple Lie algebra.
Now, this problem is known as "Feigin's Conjecture".
In the mentioned talk, Feigin proposed the existence of a certain family of homomorphisms on quantized enveloping algebra $U_q(g)$ to the ring of skew-polynomials which will led us to a deffinition of lattice $W-$algebras. These "homomorphisms" has been turned to a very useful tool for to study the fraction field of quantized enveloping algebras. [8]
There been many attempt for to construct lattice $W-$ algebras in Feigin's sence, which ensures the simplicity of the construction process of lattice $W-$algebra; for example the best known articles in the subject has been written by Kazuhiro Hikami and Rei Inoue who tried to obtain the algebra structure by using lax operators and generalized R matrices. [9] [10]
Or Alexander Belov and Alexander Antonov and Karen Chaltikian, who first tried to follow Feigin's construction but finaly they also solved part of the conjecture by getting help of lax operators, and it made very difficult to follow their publication.[11] [12]
But here in this article we will proceed and will introduce the most simplest way of constructing such kind of algebras by just employing Feigin's homomorphisms and screening operators by defining a Poisson bracket on our variables just based on our Cartan matrix. [1] [4]
In [4], Yaroslav Pugai has constructed lattice W_3 algebras already, but here we will introduce its weaker version based on a Poisson bracket as mentioned before, constructed on just Cartan matrix A_n, which will make our job more easier and more elegant.
For to do this, let us set C an arbitrary symmetrizable Cartan matrix of rank r and let $n = n_+$ be the standard maximal nilpotent sub-algebra of the Kac-Moody algebra associated with C.

[6] These results are already published in arXiv with characteristic numbers "1610.09443" and "1702.07402" and already have been sent to the Journals " Theoretical and Mathematical Physics " and " Mathematical Notes ", respectively. We still are waiting for referees.

So n is generated by elements $E_1, \cdots, E_r$ which are satisfying in Serre relations. [13] Where r stands by rank(C).

In [1], we proved that screening operators $S_{X_i^{ji}} = \sum_{\substack{j\in\mathbb{Z} \\ \text{for } i \text{ fixed}}}^{n} X_i^{ji}$; for X_i^{ji} generators of the $q-$commutative ring $\mathbb{C}_q[X_i^{ji}] := \frac{\mathbb{C}[X_i^{ji}]}{X_i^{ji}X_k^{jk} - q^{<\alpha_i,\alpha_j>}X_k^{jk}X_i^{ji}}$ and for $<\alpha_i, \alpha_j>= a_{ij}$ the ij's components of our Cartan matrix C; are satisfying in quantum Serre relations $\mathrm{ad}_q(X_i)^{1-a_{ij}}(X_j)$ for adjoint action $\mathrm{ad}_q(X_i)(X_j) = X_iX_j - q^{a_{ij}}X_jX_i$ and $X_i \in (U_q)_\alpha$, $X_j \in (U_q)_\beta$. [5]

Where $(U_q)_\alpha = \{u \in U_q(g) | q^{\mathfrak{h}}uq^{-\mathfrak{h}} = q^{\alpha(\mathfrak{h})}u \text{ for all } \mathfrak{h} \in \overset{\vee}{P}\}$ and $U_q(g) = \underset{\alpha\ in\ Q}{\oplus}(U_q)_\alpha$, for $Q = \underset{i\in I}{\oplus}\mathbb{Z}_{\alpha_i}$ the root lattice and for $\overset{\vee}{P}$ a free abelian group of rank $2|I| - \mathrm{rank}A$ with $\mathbb{Z}-$basis $\{h_i | i \in I\} \cup \{d_s | s = 1, \cdots, |I| - \mathrm{rank}A\}$ and $\mathfrak{h} = \mathbb{F} \otimes_{\mathbb{Z}} \overset{\vee}{P}$ be the $\mathbb{F}-$linear space spanned by $\overset{\vee}{P}$. [5] $\overset{\vee}{P}$ will be called dual weight lattice and $\mathfrak{h}$ the Cartan subalgebra. And $\mathbb{F}$ will stand for our ground field.[5]

Here for our Cartan matrix C, the quantum Serre relation will be

$$\begin{aligned}\mathrm{ad}_q(X_i)^{1-(-1)}(X_j) &= \mathrm{ad}_q^2(X_i)(X_j)\\ &= X_i^2X_j - [2]_qX_iX_jX_i + X_jX_i^2\\ &= X_i^2X_j - (q+q^{-1})X_iX_jX_i + X_jX_i^2\end{aligned}$$

Where $[2]_q$ stands for quantum number $[n]_q = \frac{q^n - q^{-n}}{q-q^{-1}}$ in general.

And again as what we had in [1], we can define

$$U_q(n) := \left\langle S_{X_i^{ji}}, S_{X_k^{jk}} \mid (\mathrm{ad}_q(S_{X_i^{ji}}))^2(S_{X_k^{jk}}) = 0\right\rangle$$

and for $\mathbb{C}_q[X]$ the quantum polynomial ring in one variable and twisted tensor product $\bar{\otimes}$, we can define

$$\begin{aligned}U_q(n)\bar{\otimes}\mathbb{C}_q[X_l^{jl}] := \big\langle & S_{X_i^{ji}}, S_{X_k^{jk}}, X_l^{jl} \mid (\mathrm{ad}_q(S_{X_i^{ji}}))^2(S_{X_k^{jk}}) = 0\\ & , S_{X_i^{ji}}X_l^{jl} = q^2X_l^{jl}S_{X_i^{ji}}, S_{X_k^{jk}}X_l^{jl} = q^{-1}X_l^{jl}S_{X_k^{jk}}\big\rangle\end{aligned}$$

such that we have the following embeding

$$U_q(n) \hookrightarrow U_q(n)\bar{\otimes}\mathbb{C}_q[X_l^{jl}] \hookrightarrow U_q(n)\bar{\otimes}\mathbb{C}_q[X_l^{jl}]\bar{\otimes}\mathbb{C}_q[X_m^{jm}]$$

where $\mathbb{C}_q[X_l^{jl}]\bar{\otimes}\mathbb{C}_q[X_m^{jm}] = \mathbb{C}\langle X_l^{jl}, X_m^{jm} \mid X_l^{jl}X_m^{jm} = q^{a_{lm}}X_m^{jm}X_l^{jl}$.[1]

Which will ensure the well definedness of our definition of lattice $W-$algebras.

2. Preliminary

In this section we will recover some preliminary definitions and examples on Lie algebras and quantum groups.

2.1. Lie algebras.

Definition 2.1. Lie algebra $\mathfrak{g}$ is a vector space over $\mathbb{C}$ with a bilinear map

$$[,] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g} : (x, y) \mapsto [x, y] \tag{2.1}$$

that is called bracket and satisfying:

(1) $[x, y] = -[y, x] \quad \forall x, y \in \mathfrak{g}$; skew symmetric.

(2) $[x,[y,z]]+[y,[z,x]]+[z,[x,y]]=0\ \forall x,y,z\in\mathfrak{g}$;   Jacobi Identitiy.

Example 2.2. **(1)** $\mathsf{gl_n} := \mathrm{Mat}_{n\times n}(\mathbb{C})$; $n\times n-$ matrices over $\mathbb{C}$ with $[x,y] := x\cdot y - y\cdot x$, where $\cdot$ is the product in $\mathbf{Mat}_{n\times n}(\mathbb{C})$.
(2) For V a vector space over $\mathfrak{C}$
$\mathsf{gl}(V) := \{\text{linear map} V\to V\} = \mathrm{End}(V)$
with $[x,y]=x\circ y - y\circ x$; where $\circ$ stands for the composition of maps.

Remark 2.3. When $V=\mathbb{C}^n$ then we have $\mathsf{gl}(V)=\mathsf{gl_n}$

(3) $\mathsf{sl_n} := \{X\in\mathsf{gl_n}\mid \mathrm{trac}X=0\}\subset\mathsf{gl_n}$. And is a subalgebra with the same bracket $[x,y]=x\circ y-y\circ x$.
(4) $\mathfrak{G}$ a Lie group, then
$\mathfrak{g} := \{\text{vector fields on } G \text{invariant under } G-\text{action}\}$
is a Lie algebra.
(5) Any associative algebra is a Lie algebra with bracket

$$[x,y] := xy-yx \tag{2.2}$$

and will be called underlying Lie algebra.
(6) $\mathfrak{g}$ a Lie algebra. We define $\tilde{\mathfrak{g}} := \mathfrak{g}\otimes_{\mathbb{C}}\mathbb{C}[t,t^{-1}]$ with bracket $[x\otimes f, y\otimes g] := [x,y]\otimes fg\ \ \forall x,y\in\mathfrak{g}$ and $f,g\in\mathbb{C}[t,t^{-1}]$.
$\tilde{\mathfrak{g}}$ is an infinite dimentional Lie algebra and is called Loop (Lie) algebra.
(7) For $\mathfrak{g}=sl_n$, we define
$\hat{\mathfrak{g}} = \hat{sl_n} := \mathfrak{g}\otimes\mathbb{C}[t,t^{-1}]\oplus\mathbb{C}c\oplus\mathbb{C}d$ with a bracket which satisfies in the following relations
(1) $[X\otimes t^m, Y\otimes t^n] := [X,Y]\otimes t^{m+n} + n(trXY)\delta_{m+n}c$;
(2) $[c, X\otimes t^m]=[c,d]=0$;
(3) $[d, X\otimes t^m] = m\cdot X\otimes t^m$;
for all $X,Y\in\mathfrak{g}$ and $m,n\in\mathbb{Z}$.

Exercise 2.4. Check that $\hat{\mathfrak{g}}$ is a Lie algebra.
$\hat{\mathfrak{g}}$ is called affine Lie algebra.

Definition 2.5. For $\mathfrak{g}$ a Lie algebra and subspace $\mathfrak{h}\subset\mathfrak{g}$, if
(1) $\mathfrak{h}$ is closed under bracket of $\mathfrak{g}$, i.e. $\forall x,y\in\mathfrak{h}\Rightarrow[x,y]\in\mathfrak{h}$. then $\mathfrak{h}$ is called Lie subalgebra.
(2) If we have $[x,y]\in\mathfrak{h}$ for every $x\in\mathfrak{h}$ and $y\in\mathfrak{g}$, then we call $\mathfrak{h}$ an ideal of $\mathfrak{g}$.
(3) $\mathfrak{g}$ is called abelian or commutative if $[\mathfrak{g},\mathfrak{g}]=0$.
(4) $\mathfrak{g}$ is called simple if for all ideal $\mathfrak{h}\subset\mathfrak{g}$, we have $\mathfrak{h}=0$ or $\mathfrak{h}=\mathfrak{g}$.

Exercise 2.6. Show that sl_n is simple.
(hint: $[E_{ii}-E_{jj}, E_{kl}] = (\delta_{ik}-\delta_{jk})E_{kl}$)

Definition 2.7. For Lie algebras $\mathfrak{h},\mathfrak{g}$, a linear map $\varphi:\mathfrak{g}\to\mathfrak{h}$ is called a Lie algebra homomorphism if $\varphi([x,y])=[\varphi(x),\varphi(y)]$.

Definition 2.8. For V a vector space over $\mathbb{C}$, the map $\varphi:\mathfrak{g}\to\mathsf{gl}(V)$ is a Lie algebra homomorphism and the pair (V,φ) is called a representation of $\mathfrak{g}$ or a $\mathfrak{g}-$ module.

Remark 2.9. We sometimes say " V is a $\mathfrak{g}-$module or representation of $\mathfrak{g}$" instead of "(V,φ) is a $\mathfrak{g}-$module or representation"

2.2. **q-Numbers and q-Factorials.** For any nonzero complex number number q, the $q-$number $[a]_q$, $a \in \mathfrak{C}$ is defined by

$$\begin{aligned} [a]_q \equiv [a] &:= \tfrac{q^a - q^{-a}}{q - q^{-1}} \\ &= \tfrac{e^{a\mathfrak{h}} - e^{-a\mathfrak{h}}}{e^{\mathfrak{h}} - e^{-\mathfrak{h}}} \\ &= \tfrac{\sinh a\mathfrak{h}}{\sinh \mathfrak{h}} \\ &= q^{a-1} + q^{a-3} + \cdots + q^{-(a-1)} \end{aligned}$$

where $q = e^{\mathfrak{h}}$. Clearly $\lim_{q \to 1} [a]_q \to a$ and for $m, n \in \mathfrak{N}$ and $a, b, c \in \mathbb{C}$, we have:

$$\begin{aligned} [m+n] &= q^n[m] + q^{-m}[n] \\ &= q^{-n}[m] + q^m[n] \\ &= \tfrac{q^{m+n} - q^{-(m+n)}}{q - q^{-1}} \\ &= q^{m+n-1} + q^{m+n-3} + \cdots + q^{-(n+m-1)} \end{aligned}$$

and

$$\begin{aligned} [m-n] &= q^n[m] - q^m[n] \\ &= q^{-n}[m] - q^{-m}[n] \\ 0 &= [a][b-c] + [b][c-a] + [c][a-b] \\ [n] &= [2][n-1] - [n-2] \end{aligned}$$

If $m \in \mathfrak{N}$, then we introduce the $q-$factorial

$$[m]_q! \equiv [m]!$$

by setting

(2.3) $$[m]! := [1][2] \cdots [m] \text{ and } [0]! := 1$$

By defining the expression

(2.4) $$(a : q)_n := (1-a)(1-aq)(1-aq^2) \cdots (1-aq^{n-1}) \text{ and } (a; q)_0 = 1$$

we can redefine our $q-$factorial as follows:

(2.5) $$[m]! := \frac{q^{\frac{-m(m-1)}{2}}}{(1-q^2)^m}(q^2; q^2)_m$$

2.3. **Graded rings and the quantum enveloping algebra $U_q(\mathfrak{g})$.**

Proposition 2.10. There exist an uniqe associative algebra $U(\mathfrak{g})$ with a Lie algebra homomorphism $\mathfrak{i} : \mathfrak{g} \to U(\mathfrak{g})$ satisfying the following universal property:
For any associative algebra A and a Lie algebra homomorphism $\varphi : \mathfrak{g} \to A$, there exist uniquely an algebra homomorphism $\tilde{\varphi} : U(\mathfrak{g}) \to A$ s.t. $\tilde{\varphi} \circ \mathfrak{i} = \varphi$

$$\begin{tikzcd} & U(\mathfrak{g}) \arrow[dr, "\exists!\tilde{\varphi}"] & \\ \mathfrak{g} \arrow[ur, "i"] \arrow[rr, "\forall\varphi"] & & A \end{tikzcd}$$

We call $U(\mathfrak{g})$ the universal enveloping algebra of $\mathfrak{g}$.

Proof. **Uniqueness:**$\Rightarrow$
Assume $(U, \mathfrak{i}), (U', \mathfrak{i}')$ satisfy the above proposition,

$$\begin{tikzcd} & U'(\mathfrak{g}) \arrow[dr, "\tilde{\varphi}"] & \\ \mathfrak{g} \arrow[ur, "i'"] \arrow[rr, "\varphi"] & & A \end{tikzcd} \qquad \begin{tikzcd} & U(\mathfrak{g}) \arrow[dr, "\tilde{\varphi}"] & \\ \mathfrak{g} \arrow[ur, "i"] \arrow[rr, "\varphi"] & & A \end{tikzcd}$$

$$\begin{tikzcd} & U(\mathfrak{g}) \arrow[dr, "\exists!\tilde{i}"] & \\ \mathfrak{g} \arrow[ur, "i"] \arrow[rr, "\varphi"] & & U'(\mathfrak{g}) \end{tikzcd} \qquad \begin{tikzcd} & U'(\mathfrak{g}) \arrow[dr, "\exists!\tilde{i}"] & \\ \mathfrak{g} \arrow[ur, "i'"] \arrow[rr, "\varphi"] & & U(\mathfrak{g}) \end{tikzcd}$$

$\Rightarrow \exists!\tilde{\mathrm{i}} U(\mathfrak{g}) \to U'(\mathfrak{g})$
$\exists!\tilde{\mathrm{i}}' U'(\mathfrak{g}) \to U(\mathfrak{g})$
but on the other hand we have that $\psi : U(\mathfrak{g}) \to U(\mathfrak{g})$ such that $\psi \circ i = i$ is unique and by the assumption and $\psi = id_V$ so we have $\tilde{i} \circ \tilde{i'}$ are such homomorphism and we have $\tilde{i} \circ \tilde{i'} = id_{U(\mathfrak{g})}$ and similarly $\tilde{i'} \circ \tilde{i} = id_{U'(\mathfrak{g})}$. Thus $U(\mathfrak{g}) \cong U'(\mathfrak{g})$.
Existence: $\Rightarrow$
Set $U(\mathfrak{g}) := \frac{T(\mathfrak{g})}{<X \otimes Y - Y \otimes X - [X,Y]; X, Y \in \mathfrak{g}}$, where $T(\mathfrak{g}) := \oplus_{n \geq 0} \mathfrak{g}^{\otimes n}$ is the tensor algebra of the vector space $\mathfrak{g}$.

Remark 2.11. First note that $T(\mathfrak{g})$ has the following universal property:
For any associative algebra A and a linear map $\tilde{\varphi} : T(\mathfrak{g}) \to A$, an algebra homomorphism such that $\tilde{\varphi} \circ i = \emptyset$. (costruction of $\tilde{\varphi} : T(\mathfrak{g}) \to A$ is given by $\tilde{\varphi}(X_1 \otimes X_2 \otimes \cdots \otimes X_n) = \varphi(X_1) \cdots \varphi(X_n)$ for $X_i \in \mathfrak{g}$. And since $\varphi : \mathfrak{g} \to A$ satisfies

$$\varphi(X)\varphi(Y) - \varphi(Y)\varphi(X) - \varphi([X,Y]) = 0 \tag{2.6}$$

$$\begin{tikzcd} & T(\mathfrak{g}) \arrow[dr, "\tilde{\varphi}"] & \\ \mathfrak{g} \arrow[ur, "i"] \arrow[rr, "\varphi"] & & A \end{tikzcd}$$

thus $\varphi(X \otimes Y) = \varphi(X)\varphi(Y)$ and $\varphi(Y \otimes X) = \varphi(Y)\varphi(X)$.
And as A is associative algebra then A have underlying Lie algebra structure with bracket $[X', Y'] = X'Y' - Y'X'$ such that for $\varphi : \mathfrak{g} \to A$ we have $\varphi([X,Y]) = [\varphi(X), \varphi(Y)] = \varphi(X)\varphi(Y) - \varphi(Y)\varphi(X)$, and so the icentity (3.0.3) works. So the homomorphism $\tilde{\varphi} : T(\mathfrak{g}) \to A$ with property $\tilde{\varphi} \circ i = \emptyset$ factors through $\tilde{\varphi} : U(\mathfrak{g}) \to A$. Meanly $\tilde{\varphi} \circ i([X,Y] - X \otimes Y - Y \otimes X) = \varphi([X,Y] - X \otimes Y - Y \otimes X) = 0$. Then $\tilde{\varphi}([X,Y] - X \otimes Y - Y \otimes X) = 0$.
So we can write $\tilde{\varphi} : \frac{T(\mathfrak{g})}{<[X,Y] - X \otimes Y - Y \otimes X>} \to A$, or $\tilde{\varphi} : U(\mathfrak{g}) \to A$)

□

Corollary 2.12. For any $\mathfrak{g}-$module (V, φ), we have a $U(\mathfrak{g})-$module $\tilde{\varphi} : U(\mathfrak{g}) \to End(V)$ such that $\tilde{\varphi} \circ i = \emptyset$

By this corollary we identify $\mathfrak{g}-$modules with $U(\mathfrak{g})-$modules and this will give us the following fact:

$$\begin{gathered}\text{Representation theory of Lie algebra } \mathfrak{g} \\ = \\ \text{Representation theory of associative algebra } U(\mathfrak{g})\end{gathered}$$

For $R \subset \mathbb{Z}$, let $\{X_\alpha \mid \alpha \in R\}$ be a basis of $\mathfrak{g}$, then the universal enveloping algebra $U(\mathfrak{g})$ has a basis $\{x_{\alpha_1} x_{\alpha_2} \cdots x_{\alpha_n} \mid \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n,\ \alpha_i \in R\}$ i.e. as a vector space we have $U(\mathfrak{g}) \cong_{\mathbb{C}} \mathrm{Sym}(\mathfrak{g})$.

Definition 2.13. A ring A together with a set of subgroups A_d, for $d \geq 0$ such that $A = \oplus_{d \geq 0} A_d$ as an abelian group and $st \in A_{d+e}$ for all $s \in A_d$ and $t \in A_e$. And we also require $1 \in A$.

Remark 2.14. Now let A be a graded ring. We call elements of A_d homogenous of degree d. And for $f \in A$ we can write f uniquely as a sum of homogenous elements and we denote by f_d the degree d component.
A two sided ideal I in A is homogenous if whenever $f \in I$ we have $f_d \in I$ for all $d \geq 0$.
Let $S \subset A$ be a subset (possibly empty). Then the two sided ideal generated by S is the intersection of all two sided ideals containing S. This is a two sided ideal denoted by $< S >$.
In fact $< S >$ consist of sums of elements of the form asb with $s \in S$ and $a \cdot b \in A$ and we say a two sided ideal I is generated by a subset $S \subset I$ if $I =< S >$. And also for $f, g \in A$ we have $(f+g)_d = f_d + g_d$ and $(fg)_d = \sum\limits_{i+j=d} f_i g_j$

Lemma 2.15. Let A be a graded ring. A two sided ideal I is homogenous if and only if it is generated by homogenous elements.

Proof. It is clear that if I is homogenous, it is generated by the homogenous component of all its elements.
Conversely, if I is generated by a set of homogenous elements $\{f_i\}_{i \in I}$ with $f_i \in A_{n_i}$. Then suppose $g \in I$, say $g = a_1 f_1 b_1 + \cdots + a_n f_n b_n$. Then $g_d = (a_1 f_1 b_1)_d + \cdots + (a_n f_n b_n)_d$ so it suffices to consider afb for some f homogenous of degree n.
We have

$$(afb)_d = \sum_{i+j=d} a_i (fb)_j = \sum_{i+j=d} a_i f b_{j-n}$$

with the convention that $b_k = 0$ for $k < 0$. So clearly $(afb)_d \in I$ as required.
Now suppose $I < A = \oplus_{d \geq 0} A_d$; then
I is two sided homogenous ideal. This means that if $f \in I$ then $f_d \in I$ and f_d is a homogenous element, so this means that I is generated by homogenous elements.
Conversly, suppose I generated by homogenous elements then we want to prove that I is homogenous.
For this mention that $A = \oplus A_d$ and $f_i \in A_{d_i}$ and we suppose f_i is homogenous component and $I =< \{f_i\} >= \{< f_i >\}$. And suppose $g \in I$ then $g = a_1 f_1 b_1 + \cdots + a_n f_n b_n$ and the component g_d of g is $(a_1 f_1 b_1)_d + \cdots + (a_n f_n b_n)_d$.
Now for some homogenous elements f of degree n, we write

$$(afb)_d = \sum_{i+j=d} a_i (fb)_j$$

$$= \sum_{i+j=d} a_i \sum_{n+i=j} f_n b_i$$

$$\sum_{i+j=d\ ,\ j-i=n} a_i f b_{j-n}$$

So we see that $(afb)_d \in I$ and g is linearly generated by such this elements. □

Remark 2.16. Let A be a ring and I a two-sided ideal. Then the quotient $\frac{A}{I}$ is a ring and $A \to \frac{A}{I}$ is a ring morphism. If A is graded and I homogenous, then $\frac{A}{I}$ is a graded ring and where we define

$$(\frac{A}{I})_d = \{f + I \mid f \in A_d\}$$

So by using these facts we can define

$$U(\mathfrak{g}) = \frac{T(\mathfrak{g})}{< [X,Y] - X \otimes Y - Y \otimes X >}$$

as a graded algebra.

Definition 2.17. $U_q(\mathfrak{g})$ is a $q-$deformation of universal enveloping algebra $U(\mathfrak{g})$, i.e. $U_q(\mathfrak{g})$ is an associative algebra over $\mathbb{K} = \mathbb{C}(\mathfrak{g})$ or $\mathbb{K} = \mathbb{Q}(\mathfrak{g})$ such that for $q = 1$ we get the usual universal enveloping algebra $U(\mathfrak{g})$ for semisimple Lie algebra $\mathfrak{g}$.

3. Representations of semisimple Lie algebras

Definition 3.1. A Lie algebra L is called semisimple if its radical is 0.

As we know from linear algebra, every matrix X can be written uniquely as $X = X_s + X_n$, for X_s the diagonalizable part and X_n the nilpotent part such that X_s and X_n commute.
Now we want to do something similar with Lie algebras, i.e. to find their nilpotent and semisimple elements.
If $X \in \mathrm{gl}(V)$ is a nilpotent matrix, i.e. $X^n = 0$, then for any $Y \in \mathrm{gl}(V)$ we have

$$\begin{aligned}
(\mathrm{ad}X)(Y) &= [X,Y] = XY - YX \\
(\mathrm{ad}X)(\mathrm{ad}X)(Y) &= \mathrm{ad}X([X,Y]) \\
&= [X,[X,Y]] \\
&= [X(XY - YX)] \\
&= [X^2Y - XYX] \\
&= [X^2Y] - [XYX] \\
&= XXY - YX^2 - XYX + X^2Y
\end{aligned}$$

So we have $(\mathrm{ad}X)^2(Y) = \sum_{i=0}^{2} a_i X^i Y X^{2-i}$ for $a_1 = -1$, $a_2 = -1$ and $a_3 = 2$.

$$\begin{aligned}
(\mathrm{ad}X)^3(Y) &= (\mathrm{ad}X)(\mathrm{ad}X)(\mathrm{ad}X)(Y) \\
&= (\mathrm{ad}X)(\mathrm{ad}X)([X,Y]) \\
&= (\mathrm{ad}X)([X,[X,Y]]) \\
&= [X,[X,[X,Y]]] \\
&= [X,(X(XY - YX) - (XY - YX)X)] \\
&= [X, X^2Y - XYX - XYX + YX^2]
\end{aligned}$$

$$
\begin{aligned}
&= X^3Y - 2X^2YX + XYX^2 - X^2YX \\
&\quad - 2XYX^2 + YX^3 \\
&= X^3Y - 3X^2YX - XYX^2 + YX^3 \\
&= \sum_{i=0}^{3} a_i X^i Y X^{3-i}.
\end{aligned}
$$

For $a_0 = 1$, $a_1 = -1$, $a_2 = -3$ and $a_3 = 1$. And it means that

$$
(\mathrm{ad}X)^k(Y) = \sum_{i=0}^{k} a_i X^i Y X^{k-i} \tag{3.1}
$$

and by induction on k we can see that for some $a_i \in \mathbb{F}$ and $k \geq 2n$ we have either i or $k - i \geq n$, so $X^i = 0$ or $X^{k-i} = 0$ and ultimately $(\mathrm{ad}X)^k(Y) = 0$. So $(\mathrm{ad}X)$ is nilpotent and $\mathrm{ad}X = \mathrm{ad}X_s + \mathrm{ad}X_n$

3.1. **Representation of Lie algebra** $Sl(2, \mathbb{F})$**.** Special linear Lie algebra of order n (denoted by $Sl_n(\mathbb{F})$) is the Lie algebra of $n \times n$ matrices with trace zero and with Lie bracket $[X, Y] := XY - YX$.
The simplest non-trivial Lie algebra is $Sl_2(\mathbb{C})$, consisting of 2×2 matrices with zero trace.

$$
\mathfrak{g} = Sl_2 = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \middle| \, a + d = 0 \right\}
$$

There are three basis elements $F = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, $E = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ with $\mathfrak{g} = \mathbb{C}F \oplus \mathbb{C}H \oplus \mathbb{C}E$.
So we have

$$
\begin{aligned}
[H, E] &= HE - EH = 2E \\
[H, F] &= HF - FH = -2F \\
[E, F] &= EF - FE = H
\end{aligned}
$$

3.2. **The quantum group** $U_q(Sl_2)$**.**

Definition 3.2. Let q be a fixed complex number such that $q \neq 0$ and $q^2 \neq 1$. We denote by $U_q(Sl_2)$ the algebra (associative algebra with unit) over $\mathbb{C}$ with four generators E, F, K and K^{-1} satisfying the following defining relations by Klimyk:

$$
\begin{cases}
KK^{-1} = K^{-1}K = 1; \\
KEK^{-1} = q^2E; \\
KFK^{-1} = q^{-2}F; \\
[E, F] = \frac{K - K^{-1}}{q - q^{-1}}
\end{cases}
$$

and also let us define an another set of generators $f, t = q^H, t^{-1}$ and e as follows for our future usage:

$$
\begin{cases}
tt^{-1} = t^{-1}t = 1; \\
tet^{-1} = q^2e; \\
tft^{-1} = q^{-2}f; \\
[e, f] = ef - fe = \frac{K - K^{-1}}{q - q^{-1}}
\end{cases}
$$

Proposition 3.3. The set $\{F^l K^m E^n \mid m \in \mathbb{Z}, l, n \in \mathbb{N}_0\}$, as well as the set $\{E^n K^m F^l \mid m \in \mathbb{Z}, l, n \in \mathbb{N}_0\}$, is a vector space basis of $U_q(Sl_2)$.

Definition 3.4.

$$\begin{cases} U_q^+(Sl_2) := \mathbb{K}[e]; \\ U_q^-(Sl_2) := \mathbb{K}[f]; \\ U_q(Sl_2) := \mathbb{K}[t^{\pm}]; \\ U_q^{\geq 0}(Sl_2) := \mathbb{K} < t^{\pm}, e >; \\ U_q^{\leq 0}(Sl_2) := \mathbb{K} < t^{\pm}, f > \end{cases}$$

3.3. **Nilpotent Lie algebra.**

Definition 3.5. A Lie algebra is called nilpotent if there exists a decreasing finite sequence $(g_i)_{i\in[0,k]}$ of ideals such that $g_0 = g, g_k = 0$ and $[g, g_i] \subset g_{i+1}$ for all $i \in [0, k-1]$.

Definition 3.6. Let $\mathfrak{g}$ be a finite dimensional Lie algebra over field $\mathbb{K}$. A subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ is called a Cartan subalgebra if it is nilpotent and equal to its normalizer, which is the set of those elements $x \in \mathfrak{g}$ such that $[x, \mathfrak{h}] \subset \mathfrak{h}$.

From this definition we can find that if $\mathfrak{g}$ is nilpotent then $\mathfrak{g}$ is a Cartan subalgebra of itself.
Now let $\mathfrak{g}$ be the Lie algebra of all endomorphisms of $\mathbb{K}^n$ with $[f, g] = f \circ g - g \circ f$, then the set of all endomorphisms f of $\mathbb{K}^n$ of the form $f(x_1, \cdots, x_n) = (\lambda_1 x_1, \cdots, \lambda_n x_n)$ is a Cartan subalgebra of $\mathfrak{g}$.
For to see this, let us consider the set of all endomorphisms f with $L := \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f((x_i)_1^n) = (\lambda_i x_i)_1^n\}$. It is clear to see that $L \leq \text{End}(\mathfrak{K}^n)$ and for to see that L is a nilpotent subalgebra of $\mathfrak{g}$ we have:

$$\begin{aligned}
&g_0 := L = \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(x_1, \cdots, x_n) := (\lambda_1 x_1, \cdots, \lambda_n x_n)\}; \\
&g_1 := L = \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(x_1^{(1)}, \cdots, x_n^{(1)}) := (\lambda_1^{(1)} x_1^{(1)}, \cdots, \lambda_{n-1}^{(1)} x_{n-1}^{(1)}, 0)\}; \\
&g_2 := L = \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(x_1^{(2)}, \cdots, x_n^{(2)}) := (\lambda_1^{(2)} x_1^{(2)}, \cdots, \lambda_{n-2}^{(2)} x_{n-2}^{(2)}, 0, 0)\}; \\
&\quad \vdots \qquad\qquad\qquad\qquad \vdots \\
&g_{n-1} := L = \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(x_1^{(n-1)}, \cdots, x_n^{(n-1)}) := (\lambda_1^{(n-1)} x_1^{(n-1)}, 0, \cdots, 0)\}; \\
&g_n := L = \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(x_1^{(n)}, \cdots, x_n^{(n)}) := (0, \cdots, 0)\};
\end{aligned}$$

And on the other hand we have that for $s \in \{0, \cdots, n-i\}$, $j \in \{1, \cdots, n\}$ and $i + s = j$:

$$\begin{aligned}
[g_0, g_1] = g_0 g_1 - g_1 g_0 &= \{f : \mathfrak{K}^n \to \mathfrak{K}^n \mid f(y_1, \cdots, y_n) \\
&= (\lambda_i \lambda_{i+s}^{(1)} x_i x_{i+s}^{(1)})_{i=1}^{n-s} \\
&= ((\lambda_1 \lambda_{1+s}^{(1)} x_1 x_{1+s}^{(1)})_{s=0}^{n-s}, \lambda_2 \lambda_{2+s}^{(1)} x_2 x_{2+s}^{(1)} \cdots \lambda_n \lambda_{n+s}^{(1)} x_n x_{n+s}^{(1)}) \\
&= (\lambda_1 \lambda_1^{(1)} x_1 x_1^{(1)}, \lambda_1 \lambda_2^{(1)} x_1 x_2^{(1)}, \lambda_1 \lambda_3^{(1)} x_1 x_3^{(1)} \cdots, \lambda_n \lambda_n^{(1)} \\
&\quad x_n x_n^{(1)}, \lambda_2 \lambda_2^{(1)} x_2 x_2^{(1)}, \lambda_2 \lambda_2^{(1)} x_2 x_3^{(1)}, \cdots, \lambda_2 \lambda_n^{(1)} x_2 x_n^{(1)}, \cdots, \\
&\quad \lambda_n \lambda_n^{(1)} x_n x_n^{(1)}, \lambda_n \lambda_{n+1}^{(1)} x_n x_{n+1}^{(1)}, \cdots, \lambda_n \lambda_{2n-1}^{(1)} x_n x_{2n-1}^{(1)})\} \\
&= (\lambda_1^{(2)} x_1^{(2)}, \cdots, \lambda_{n-2}^{(2)} x_{n-2}^{(2)}, 0, 0) \\
&= g_2
\end{aligned}$$

So it shows that L is a nilpotent subalgebra of $\mathfrak{g}$ and obviously is a Cartan subalgebra of $\mathfrak{g}$.

Example 3.7. For nilpotent Lie algebras we have an obvious example:
The lie algebra of strictly upper triangular matrices:

$$g = g_0 = \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & 0 & a_{23} & a_{24} & a_{25} \\ 0 & 0 & 0 & a_{34} & a_{35} \\ 0 & 0 & 0 & 0 & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_1 = \begin{bmatrix} 0 & 0 & a_{13} & a_{14} & a_{15} \\ 0 & 0 & 0 & a_{24} & a_{25} \\ 0 & 0 & 0 & 0 & a_{35} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_2 = \begin{bmatrix} 0 & 0 & 0 & a_{14} & a_{15} \\ 0 & 0 & 0 & 0 & a_{25} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & a_{15} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \Rightarrow [g, g_3] \subset g_4$$

3.4. **The nilpotent part of $U_q(\mathfrak{g})$.**

3.5. **Space of roots.** In the study of semisimple Lie algebras over an algebraically closed field $\mathbb{K}$, an important role is played by the roots of a semisimple Lie algebra, which are defined as follows:
Let $\mathfrak{h}$ be a Cartan subalgebra of $\mathfrak{g}$. For a non zero linear function $\alpha \in \mathfrak{h}$, let g_α denote the linear subspace of $\mathfrak{g}$ given by the condition

$$g_\alpha = \{X \in \mathfrak{g} \mid [H, X] = \alpha(H)X,\ H \in \mathfrak{h}\} \tag{3.2}$$

if $g_\alpha \neq 0$ then α is called a root of $\mathfrak{g}$ with respect to $\mathfrak{h}$.
The set σ of all non-zero roots is called the root system or system of roots of $\mathfrak{g}$.
One has the root decomposition

$$\mathfrak{g} = \mathfrak{h} + \sum_{\alpha \in \sigma} g_\alpha \tag{3.3}$$

The root system and the root decomposition of a semi-simple Lie algebra have the following properties:
a) σ generate $\mathfrak{h}$ and is a reduced root system in the abstract sense (the system σ is irreducible if and only if $\mathfrak{g}$ is simple.).
b) For any $\alpha \in \sigma$ we have

$$\text{dim } g_\alpha = \text{dim } [g_\alpha, g_{-\alpha}] = 1$$

there is a unique $H_\alpha \in [g_\alpha, g_{-\alpha}]$ such that $\alpha(H_{alpha}) = 2$.
c) For every non zero $X_{alpha} \in g_\alpha$ there is a unique $Y_\alpha \in g_{-\alpha}$ such that $[X_\alpha, Y_\alpha] = H_\alpha$ and $[H_\alpha, X_\alpha] = 2X_\alpha$ and $[H_\alpha, Y_\alpha] = -2Y_\alpha$. Moreover

$$\beta(H_\alpha) = \frac{2(\alpha, \beta)}{(\alpha, \alpha)} \ \text{for} \alpha, \beta \in \sigma$$

Where $(,)$ is the scalar product induced by the killing form.

Remark 3.8. The killing form is an inner product on a finite dimensional Lie algebra $\mathfrak{g}$ defined by

$$\beta(X, Y) = \mathrm{Tr}(\mathrm{ad(X)}\ \mathrm{ad(Y)})$$

in the adjoint representation $\mathrm{ad}(X)(Y) = [X, Y]$.
The adjoint representation is a Lie algebra representation:

$$\begin{aligned}
[\mathrm{ad}(X_1), \mathrm{ad}(X_2)](Y) &= (\mathrm{ad}(X_1)\mathrm{ad}(X_2) - \mathrm{ad}(X_2)\mathrm{ad}(X_1))(Y) \\
&= \mathrm{ad}(X_1)[X_2, Y] - \mathrm{ad}(X_2)[X_1, Y] \\
&= [X_1, [X_2, Y]] - [X_2, [X_1, Y]] \\
&= X_1[X_2, Y] - [X_2, Y]X_1 - X_2[X_1, Y] \\
&\quad +[X_1, Y]X_2 \\
&= X_1(X_2Y) - X_1(YX_2) - (X_2Y)X_1 + (YX_2)X_1 \\
&\quad -X_2(X_1Y) + X_2(YX_1) + (X_1Y)X_2 - (YX_1)X_2 \\
&= (X_1X_2)Y - (X_2X_1)Y - Y(X_1X_2) - Y(X_2X_1) \\
&= [X_1X_2 - X_2X_1, Y] \\
&= [[X_1, X_2], Y] \\
&= \mathrm{ad}([X_1, X_2])(Y)
\end{aligned}$$

And also we have

$$\beta(\mathrm{ad}(X)Y, Z) = -\beta(Y, \mathrm{ad}(X)Z)$$

where $\mathfrak{g}$ is a semisimple Lie algebra, the killing form is nondegenerate.
d) If $\alpha, \beta \in \sigma$ and $\alpha + \beta \neq 0$, then g_α and g_β are orthogonal with respect to the killing form and $[g_\alpha, g_\beta] = g_{\alpha+\beta}$.

A basis $\{\alpha_1, \cdots, \alpha_n\}$ of the root system σ is also called a system of simple roots of the algebra $\mathfrak{g}$.
Let σ_+ be the system of positive roots with respect to the given basis and let $X_{-\alpha} = Y_\alpha$ for $\alpha \in \sigma_+$. Then the elements $H_{\alpha_1}, \cdots, H_{\alpha_k}, X_\alpha$ for $\alpha \in \sigma$ form a basis of $\mathfrak{g}$, called a Cartan basis.
On the other hand, the elements $X_{\alpha_i}$and$X_{-\alpha_i}$ for $i \in \{1, \cdots, n\}$, form a system of generators of $\mathfrak{g}$, and the defining relations have the following form:

$$\begin{cases} [[X_{\alpha_i}, X_{-\alpha_i}], X_{\alpha_j}] = n(i,j)X_{\alpha_j} \\ (\mathrm{ad}X_{\alpha_i})^{1-n(i,j)}X_{\alpha_j} = 0 \\ (\mathrm{ad}X_{-\alpha_i})^{1-n(i,j)}X_{-\alpha_j} = 0 \end{cases}$$

And also $[[X_{\alpha_i}, X_{-\alpha_i}], X_{-\alpha_j}] = -n(i,j)X_{\alpha_j}$ for $i, j \in \{1, \cdots, n\}$ and

$$n(i,j) = \alpha_j(H_i) = \frac{2(\alpha_i, \alpha_j)}{(\alpha_j, \alpha_i)}$$

property **d)** also implies that

$$[X_\alpha, X_\beta] = \begin{cases} N_{\alpha,\beta}X_{\alpha+\beta} & \text{if } \alpha + \beta \in \sigma \\ 0 & \text{if } \alpha + \beta \notin \sigma \end{cases}$$

Where $N_{\alpha,\beta} \in \mathbb{K}$.

Example 3.9. For $\alpha \in \mathbb{C}$ we have

$$(U_q(Sl_2))_\alpha := \{x \in U_q(Sl_2) \mid t \cdot x \cdot t^{-1} = q^\alpha x\} \tag{3.4}$$

Then we have

$$(U_q(Sl_2))_\alpha = 0 \Leftrightarrow \alpha \in 2\mathbb{Z}$$

and the root decomposition will be $(U_q(Sl_2))_\alpha = \oplus_{\alpha \in 2\mathbb{Z}}(U_q(Sl_2))_\alpha$ and $\alpha \in 2\mathbb{Z}$ will be called a root and $(U_q(Sl_2))_\alpha$ will be called the root space.

4. Feigin's homomorphisms and quantum groups

In this section we will introduce Feigin's homomorphisms and will see that how they will help us to prove our main and fundamental theorems on screening operators.
"Feigin's homomorphisms" was born in his new formulation on quantum Gelfand-Kirillov conjecture, which came on a public view at RIMS in 1992 for the nilpotent part $U_q(\mathfrak{n})$, that are now known as "Feigin's Conjecture". In that mentioned talk, Feigin proposed the existence of a family of homomorphisms from a quantized enveloping algebra to rings of skew-polynomials. These "homomorphisms" are became very useful tools for to study the fraction field of quantized enveloping algebra. [10]

Feigin's homomorphisms on $U_q(\mathfrak{n})$. Here we will briefly try to show that what are Feigin's homomorphisms and how they will guide us to reach and to prove that the screening operators are satisfying in quantum Serre relations.

Set C as an arbitrary symmetrizable Cartan matrix of rank r, and $\mathfrak{n} = \mathfrak{n}_+$ the standard maximal nilpotent sub-algebra in the Kac-Moody algebra associated with C (thus, n is generated by the elements $E_1, ..., E_r$ satisfying in the Serre relations). As always $U_q(n)$ is the quantized enveloping algebra of n. And $A = (A_{ij}) = (d_i c_{ij})$ is the symmetric matrix corresponding to C for non-zero relatively prime integers $d_1, ..., d_n$ such that $d_i a_{ij} = d_j a_{ji}$ for all i, j. And set g as a Kac-Moody Lie algebra attached to A, on generators $E_i, F_i, H_i, 1 \leq i \leq n$.[14] Let us to mention some of the structures related to g that we will use them here:

the triangular decomposition $g = \mathfrak{n}_- \oplus \mathfrak{h} \oplus \mathfrak{n}_+$;

the dual space $\mathfrak{h}^*$; elements of $\mathfrak{h}^*$ will be referred to as weights;

the root space decomposition $\mathfrak{n}_\pm = \oplus_{\alpha \in \Delta_\pm} g_\alpha, g_{\alpha_i} = \mathbb{C}E_i$;

the root lattice $\Lambda \in \mathfrak{h}^*, \{\alpha_1, \cdots, \alpha_n\} \subset \Delta_+ \subset \mathfrak{h}^*$ being the set of simple roots;

the invariant bilinear form $\Lambda \times \Lambda \to \mathbb{Z}$ defined by $<\alpha_i, \alpha_j>= d_i a_{ij}$. [14]

Set A_1 and A_2 as a $\Lambda-$ graded associative algebras and define a $q-$ twisted tensor product as the algebra $A_1 \bar{\otimes} A_2$ isomorphic with $A_1 \otimes A_2$ as a linear space with multiplication given by $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') := q^{<\alpha_1', \alpha_2>} a_1 a_1' \otimes a_2 a_2'$, where $\alpha_1' = deg(a_1')$ and $\alpha_2 = deg(a_2)$. And by this definition $A_1 \bar{\otimes} A_2$ become a $\Lambda-$ graded algebra.

Proposition 4.1. Set g an arbitrary Kac-Moody algebra, then the map

$$\bar{\Delta} : U_q^{\pm}(g) \to U_q^{\pm}(g) \bar{\otimes} U_q^{\pm}(g) \tag{4.1}$$

Such that

$$\begin{cases} \bar{\Delta}(1) := 1 \otimes 1 \\ \bar{\Delta}(E_i) := E_i \otimes 1 + 1 \otimes E_i \\ \bar{\Delta}(F_i) := F_i \otimes 1 + 1 \otimes F_i \end{cases}$$

for $1 \leqslant i \leqslant n,$ is a homomorphism of associative algebras. [15][10]

Remark 4.2. there is no such map as $U_q^{\pm}(g) \to U_q^{\pm}(g) \bar{\otimes} U_q^{\pm}(g)$ in the case that g is an associative algebra. [15]

And as always after defining a co-multiplication, $\bar{\Delta}$, then we can extend it by a iteration as a sequence of maps [16]

$$\bar{\Delta}^n : U_q^{-}(g) \to U_q^{-}(g)^{\otimes n},\ n = 2, 3, ... \tag{4.2}$$

determined by $\bar{\Delta}^2 = \bar{\Delta},\ \bar{\Delta}^m = (\bar{\Delta} \otimes id) \circ \bar{\Delta}^{n-1}.$

Now set $\mathbb{C}[X_i]$ as a ring of polynomials in one variable and by equipping it by grading structure $degX_i = \alpha_i$ for any simple root α_i, we can regard it as a $\Lambda-$ graded.
By this grading there will be a morphism of $\Lambda-$ graded associative algebras

$$\phi_i : U_q^{-}(g) \to \mathbb{C}[X_i] : F_j \mapsto \delta_{ij} x_i \tag{4.3}$$

By following this construction for any sequence of simple roots $\beta_{i_1}, \cdots, \beta_{i_k}$, there will be a morphism of $\Lambda-$ graded associative algebras

$$(\phi_{i_1} \otimes \phi_{i_k}) \circ \bar{\Delta}^k : U_q^{-}(g) \to \mathbb{C}[X_{1i_1}] \bar{\otimes} \cdots \bar{\otimes} \mathbb{C}[X_{ki_k}] \tag{4.4}$$

(the cause of double indexation here is the appearance of i_js more than once in the sequence).
And finally, $\mathbb{C}[X_{1i_1}] \bar{\otimes} \cdots \bar{\otimes} \mathbb{C}[X_{ki_k}]$ is an algebra of skew polynomials $\mathbb{C}[X_{1i_1}, \cdots, X_{ki_k}]$, with $\Lambda-$ grading $X_{si_s} X_{ti_t} = q^{<\alpha_{i_s}, \alpha_{i_t}>} X_{ti_t} X_{si_s}$, for $s > t$.
But let us to simplify it as $X_i X_j = q^{<degX_i, degX_j>} X_j X_i$; the one that we will use it always.

So very briefly we constructed the already mentioned family of morphisms (Feigin's homomorphisms) from $U_q^{-}(g)$ (the maximal nilpotent subalgebra of a quantum group associated to an arbitrary Kac-Moody algebra) to the algebra of skew polynomials.

4.1. **the contribution between Quantum Serre relations and screening operators.**

Theorem 4.3. Set $Q = q^2$ and points $x_1, \cdots, x_n$ such that $x_i x_j = Q x_i x_j$ for $i < j$. And set $\Sigma^x = x_1 + \cdots + x_n$. If $Q^N = 1$ and $x_i^N = 0$ for some natural number N, then we claim that $(\Sigma^x)^N = 0$

Proof. It's straightforward, just needs to use q-calculation. $\square$

4.2. **sl(3) case.** As we know, $M_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ is the generalized Cartan matrix for $sl(3)$. Set $M_{q_2} = \begin{bmatrix} q^2 & q^{-1} \\ q^{-1} & q^2 \end{bmatrix}$ and call it Cartan type matrix related to M_2.

Theorem 4.4. Suppose we have two different types of points x_i, Namely, set $(x_{2i-1})_i$, that we will call them of type 1 and $(x_{2i})_i$, that we will call them of type 2 for $i \in I = \{1, 2\}$, and the following $q-$ commutative relations:

$$\begin{cases} x_j x_{j'} = q^2 x_{j'} x_j & \text{if } j < j' \; and \; j, j' \in \{1, 3\} \; and \; j = j' \\ x_i x_{i'} = q^2 x_{i'} x_i & \text{if } i < i' \; and \; i, i' \in \{2, 4\} \; and \; i = i' \\ x_i x_j = q^{-1} x_j x_i & \text{if } i < j \end{cases}$$

Set $\Sigma_1^x = \Sigma_{i \in I} x_{2i+1}$ and $\Sigma_2^x = \Sigma_{i \in I} x_{2i}$. We will call them screening operators.
Then we claim that Σ_1^x and Σ_2^x are satisfying on quantum Serre relations:

$$(\Sigma_1^x)^2 \Sigma_2^x - [2]_q \Sigma_1^x \Sigma_2^x \Sigma_1^x + \Sigma_2^x (\Sigma_1^x)^2 = 0 \tag{4.5}$$

$$(\Sigma_2^x)^2 \Sigma_1^x - [2]_q \Sigma_2^x \Sigma_1^x \Sigma_2^x + \Sigma_1^x (\Sigma_2^x)^2 = 0$$

Proof. It's straightforward, just needs to use q-calculation. □

Theorem 4.5. Prove Theorem 1.2.1 in a general case, i.e. Set points $X_i \in \{X_1, \cdots, X_n\}$ and $Y_i \in \{Y_1, \cdots, Y_n\}$ with the following relations;

$$\begin{cases} X_i X_j = q^2 X_j X_i & \text{if } i < j \\ Y_i Y_j = q^2 Y_j Y_i & \text{if } i < j \\ X_i Y_j = q^{-1} Y_j X_i & \text{if } i < j \end{cases}$$

and the screening operators $\Sigma_1^x = \Sigma_{i=1}^k X_i$ and $\Sigma_1^y = \Sigma_{j=1}^k Y_j$.
We claim that Σ_1^x and Σ_1^y are satisfying in quantum Serre relations.

Proof. Proof by induction on k.
As we see in theorem 4.1.1, it's true for $k = 2$.
Suppose that is true for $k = n$, we will prove that it's true for $k = n + 1$.
As we set it out, $\mathfrak{n}$ is a nilpotent Lie algebra, so the Cartan sub-algebra of $\mathfrak{n}$ is equal to $\mathfrak{n}$ with Chevally generators Σ_1^X and Σ_1^y as they are satisfying in quantum Serre relations.
So we can define $U_q(n) :=< \Sigma_1^x, \Sigma_1^y \mid (\Sigma_1^x)^2 \Sigma_1^y - (q + q^{-1}) \Sigma_1^x \Sigma_1^y \Sigma_1^x + \Sigma_1^y (\Sigma_1^x)^2 = 0 >$.
Let $\mathbb{C}_q[X]$ be the quantum polynomial ring in one variable. We define:
$U_q(n) \bar{\otimes} \mathbb{C}_q[X] :=< \Sigma_1^x, \Sigma_1^y, X \mid (\Sigma_1^x)^2 \Sigma_1^y - (q + q^{-1}) \Sigma_1^x \Sigma_1^y \Sigma_1^x + \Sigma_1^y (\Sigma_1^x)^2 = 0, \Sigma_1^x X = q^2 X \Sigma_1^x, \Sigma_1^y X = q^{-1} X \Sigma_1^y >$.
Here $\bar{\otimes}$ means quantum twisted tensor product.
We define the embedding $U_q(n) \to U_q(n) \bar{\otimes} \mathbb{C}_q[X]$: $\Sigma_1^X \mapsto \Sigma_1^x + X$; $\Sigma_1^y \mapsto \Sigma_1^y$.
Claim 1:
$(\Sigma_1^x + X) \; and \; \Sigma_1^y$ are satisfying on quantum Serre relations.

proof of claim 1:

$(\Sigma_1^x+X)^2\Sigma_1^y-(q+q^{-1})(\Sigma_1^x+X)\Sigma_1^y(\Sigma_1^x+X)+\Sigma_1^y(\Sigma_1^x+X)^2 = (\Sigma_1^x)^2\Sigma_1^y + \Sigma_1^x X\Sigma_1^y + X\Sigma_1^x\Sigma_1^y + X^2\Sigma_1^y - (q+q^{-1})(\Sigma_1^x\Sigma_1^y\Sigma_1^x + X\Sigma_1^y\Sigma_1^x + \Sigma_1^x\Sigma_1^y X + X\Sigma_1^y X) + \Sigma_1^y(\Sigma_1^x)^2 + \Sigma_1^y\Sigma_1^x X + \Sigma_1^y X\Sigma_1^x + \Sigma_1^y X^2 = (\Sigma_1^x)^2\Sigma_1^y - (q+q^{-1})\Sigma_1^x\Sigma_1^y\Sigma_1^x + \Sigma_1^y(\Sigma_1^x)^2 + (q^2X\Sigma_1^x\Sigma_1^y + X\Sigma_1^x\Sigma_1^y + X^2\Sigma_1^y - (q+q^{-1})X\Sigma_1^y\Sigma_1^x - (q+q^{-1})qX\Sigma_1^x\Sigma_1^y - (q+q^{-1})q^{-1}X^2\Sigma_1^y + qX\Sigma_1^y\Sigma_1^x + q^{-1}X\Sigma_1^y\Sigma_1^x + q^{-2}X^2\Sigma_1^y = 0+0=0.$
So it's well defined.
Now set $X = X_{n+1}$.
We will have the new operators $\Sigma_1^{x\prime} = X_1 + \cdots + X_n + X_{n+1}$ and $\Sigma_1^{y\prime} = Y_1 + \cdots + Y_n$.
Now define:

$$U_q(n) \to U_q(n)\bar{\otimes}\mathbb{C}_q[X] \hookrightarrow U_q(n)\bar{\otimes}\mathbb{C}_q[X]\bar{\otimes}\mathbb{C}_q[Y]$$

such that

$$\Sigma_1^x \longmapsto \Sigma_1^x + X \longmapsto \Sigma_1^{x\prime}$$
$$\Sigma_1^y \longmapsto \Sigma_1^y \longmapsto \Sigma_1^{y\prime} + Y$$

Notice that $\mathbb{C}_q[X]\bar{\otimes}\mathbb{C}_q[Y] \cong \mathbb{C} < X, Y | XY = q^{-1}YX >$.
And Define:
$U_q(n)\bar{\otimes}\mathbb{C}_q[X,Y] :=< \Sigma_1^x, \Sigma_1^y, X, Y \mid \Sigma_1^x \, and \, \Sigma_1^y \; stisfying \; q-Serre \; relations \; and \, \Sigma_1^x X = q^2 X\Sigma_1^x, \Sigma_1^y X = q^{-1}X\Sigma_1^y, \Sigma_1^x Y = q^{-1}Y\Sigma_1^x, \Sigma_1^y Y = q^2 Y\Sigma_1^y, XY = q^{-1}YX >$.

Claim 2:
$\Sigma_1^{x\prime} \; and \; (\Sigma_1^{y\prime} + Y)$ are satisfying on quantum Serre relations.

proof of claim 2:
$(\Sigma_1^{x\prime})^2(\Sigma_1^{y\prime}+Y) - (q+q^{-1})\Sigma_1^{x\prime}(\Sigma_1^{y\prime}+Y)\Sigma_1^{x\prime} + (\Sigma_1^{y\prime}+Y)(\Sigma_1^{x\prime})^2 = (\Sigma_1^{x\prime})^2\Sigma_1^{y\prime} + (\Sigma_1^{x\prime})^2 Y - (q+q^{-1})(\Sigma_1^{x\prime}\Sigma_1^{y\prime}\Sigma_1^{x\prime} + \Sigma_1^{x\prime}Y\Sigma_1^{x\prime}) + \Sigma_1^{y\prime}(\Sigma_1^{x\prime})^2 + Y(\Sigma_1^{x\prime})^2 = (\Sigma_1^{x\prime})^2\Sigma_1^{y\prime} - (q+q^{-1})\Sigma_1^{x\prime}\Sigma_1^{y\prime}\Sigma_1^{x\prime} + \Sigma_1^{y\prime}(\Sigma_1^{x\prime})^2 + (\Sigma_1^{x\prime})^2 Y + \Sigma_1^{x\prime}Y = 0 + q^{-2}Y(\Sigma_1^{x\prime})^2 - (q+q^{-1})q^{-1}Y(\Sigma_1^{x\prime})^2 + Y(\Sigma_1^{x\prime})^2 = 0+0=0.$
lets do some part of this computation that maybe make confusion:
$(\Sigma_1^{x\prime})^2 Y = (\Sigma_1^x + X_{n+1})^2 Y = (\Sigma_1^x)^2 Y + X_{n+1}^2 Y + \Sigma_1^x X_{n+1} Y + X_{n+1}\Sigma_1^x Y + q^{-2}Y(\Sigma_1^x)^2 + q^{-2}YX_{n+1}^2 + q^{-2}Y\Sigma_1^x X_{n+1} + q^{-2}YX_{n+1}\Sigma_1^x = q^{-2}(Y((\Sigma_1^{x2} + X_{n+1}^2 + \Sigma_1^x X_{n+1} + X_{n+1}\Sigma_1^x)) = q^{-2}Y(\Sigma_1^{x\prime})^2.$
And $\Sigma_1^{x\prime}Y = (\Sigma_1^x + X_{n+1})Y = \Sigma_1^x Y + X_{n+1}Y = q^{-1}Y\Sigma_1^x + q^{-1}YX_{n+1} = q^{-1}(Y(\Sigma_1^x + X_{n+1})) = q^{-1}Y\Sigma_1^{x\prime}$. And by substituting these, we have the result.
So our definition is well defined.
Now set $Y = Y_{n+1}$ and we are done.

□

4.3. **affinized Lie algebra** $\hat{sl(2)}$**.** As we know, $\hat{M}_2 = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$ is the generalized Cartan matrix for $\hat{sl(2)}$. Set $\hat{M}_{q_2} = \begin{bmatrix} q^2 & q^{-2} \\ q^{-2} & q^2 \end{bmatrix}$ and call it Cartan type matrix related to $\hat{M}_2$.
$\hat{sl(2)}$ is satisfying in Theorems 1.2.1 and 1.2.2 as well; but what we need is

just to change the quantum Serre relations in the following case:

(4.6)
$(\Sigma_1^x)^3\Sigma_1^y-(q^2+1+q^{-2})(\Sigma_1^x)^2\Sigma_1^y\Sigma_1^x+(q^2+1+q^{-2})\Sigma_1^x\Sigma_1^y(\Sigma_1^x)^2-\Sigma_1^y(\Sigma_1^x)^3=0$
$(\Sigma_1^y)^3\Sigma_1^x-(q^2+1+q^{-2})(\Sigma_1^y)^2\Sigma_1^x\Sigma_1^y+(q^2+1+q^{-2})\Sigma_1^y\Sigma_1^x(\Sigma_1^y)^2-\Sigma_1^x(\Sigma_1^y)^3=0$
And to change the $q-$ commutation relations also; according to our new Cartan type matrix

$$\begin{cases} X_iX_j = q^2X_jX_i & \text{if } i<j \\ Y_iY_j = q^2Y_jY_i & \text{if } i<j \\ X_iY_j = q^{-2}Y_jX_i & \text{if } i<j \end{cases}$$

But lets try to prove it in the case of Laurent skew $q-$polynomials $\mathbb{C}[X, X^{-1}]$.

Theorem 4.6. Set points $X_i \in \{X_1, \cdots, X_k\}$ and $X_j^{-1} \in \{X_1^{-1}, \cdots, X_k^{-1}\}$ with the following relations;

$$\begin{cases} X_iX_j = q^2X_jX_i & \text{if } i<j \\ X_iX_j^{-1} = q^{-2}X_j^{-1}X_i & \text{if } i<j \end{cases}$$

and the screening operators $\Sigma_1^x = \Sigma_{i=1}^k X_i$ and $\Sigma_1^{x^{-1}} = \Sigma_{j=1}^k X_j^{-1}$.
Again we claim that Σ_1^x and $\Sigma_1^{x^{-1}}$ are satisfying in quantum Serre relations (1.11).

Proof. Proof by induction on k.
For $k=2$, Set $\Sigma_1^x = x_1+x_2$ and $\Sigma_1^{x^{-1}} = x_1^{-1}+x_2^{-1}$ and as we checked out, it's straightforward to show that they are satisfying in quantum Serre relations (1.11).
Suppose that it's true for $k=n$ components $x_1, \cdots, x_n$. Again as before we define:
$U_q(n) := \{\Sigma_1^x, \Sigma_1^{x^{-1}} | (\Sigma_1^x)^3\Sigma_1^{x^{-1}} - (q^2+1+q^{-2})(\Sigma_1^x)^2\Sigma_1^{x^{-1}}\Sigma_1^x + (q^2+1+q^{-2})\Sigma_1^{x^{-1}}$
$\Sigma_1^x(\Sigma_1^{x^{-1}})^2 - \Sigma_1^{x^{-1}}(\Sigma_1^x)^3 = 0\}$
Define $U_q(n) \to \mathbb{C}_q[X, X^{-1}]$; $\Sigma_1^x \mapsto \lambda X$; $\Sigma_1^{x^{-1}} \mapsto X^{-1}$ for $\lambda \in \mathbb{C}^*$
And define $U_q(n) \to U_q(n)\bar{\otimes}U_q(n) \to U_q(n)\bar{\otimes}\mathbb{C}[X, X^{-1}]$; $\Sigma_1^x \mapsto \Sigma_1^x \otimes 1 + X\bar{\otimes}\Sigma_1^x$;$\Sigma_1^{x^{-1}} \mapsto \Sigma_1^{x^{-1}} \otimes 1 + X^{-1}\bar{\otimes}\Sigma_1^{x^{-1}}$.
And $U_q(n) \to \underbrace{U_q(n)\bar{\otimes}U_q(n)\bar{\otimes}\cdots\bar{\otimes}U_q(n)}_{n\ terms} \to \mathbb{C}[X_1, X_1^{-1}]\bar{\otimes}\mathbb{C}[X_2, X_2^{-1}]\bar{\otimes}\cdots\bar{\otimes}$
$\mathbb{C}[X_n, X_n^{-1}] \cong \mathbb{C}[X_1, X_1^{-1}, \cdots, X_n, X_n^{-1}]$

□

5. Deformation quantization

5.1. Poisson algebara.

Definition 5.1. A Poisson algebara over a (commutative) ring (with unit) $\mathbb{K}$ is a triple $(A, \cdot, \{\})$ where $(A, \cdot)$ is an associative $\mathbb{K}-$algebra and $(A, \{\})$ is a Lie $\mathbb{K}-$algebra, such that the identity:

$$\{a \cdot b, c\} = a \cdot \{b, c\} + \{a, c\} \cdot b \tag{5.1}$$

satisfies for all $a, b, c \in A$.

This means that the axioms for a Poisson algebra are the following:
1. Associativity: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
2. Antisymmetric: $\{a, b\} + \{b, a\} = 0$;
3. Jacobi identity: $\{\{a, b\}, c\} + \{\{b, c\}, a\} + \{\{c, a\}, b\} = 0$;
4. $\{a \cdot b, c\} = a \cdot \{b, c\} + \{a, c\} \cdot b$.

Example 5.2. Every Lie algebra is a Poisson algebra with respect to the null associative product $a \cdot b = 0$.

Example 5.3. Every associative algebra is a Poisson algebra with respect to the null Poisson bracket $\{a, b\} = 0$; such an algebra is called null Poisson algebra.

Example 5.4. An associative algebra A is a Poisson algebra if we put $\{a, b\} = a \cdot b - b \cdot a$. And indead

$$\begin{aligned} \{ab, c\} = & (ab)c - c(ab) \\ & a(bc) - a(cb) + (ac)b - (ca)b \\ & a\{b, c\} + \{a, c\}b; \end{aligned}$$

So we get a Poisson algebra.

Example 5.5. If $\mathfrak{g}$ is a Lie algebra then its universal enveloping algebra $U(\mathfrak{g})$ is a Poisson algebra by means of the same computation performed above.

5.2. **Quantization.** Roughly speaking, a quantization of a commutative associative algebra A_0 over $\mathbb{K}$ is a (not necessarily commutative) deformation of A_0 depending on a parameter $\mathfrak{h}$ (Planck's constant), i.e. an associative algebra A over $\mathbb{K}[[\mathfrak{h}]]-$module.
Given A, we can define a new operation on A_0 (The Poisson bracket) by the formula:

$$\{a \text{ mod } \mathfrak{h}, b \text{ mod } \mathfrak{h}\} = \frac{[a, b]}{\text{mod } \mathfrak{h}} \tag{5.2}$$

Such that as we see is Indeed a Poisson bracket. Thus A_0 becomes a Poisson algebra.

Now we shall slightly change our point of view on quantization.

Definition 5.6. A quantization of a Poisson algebra A_0 is an associative algebra deformation A of A_0 over $\mathbb{K}[[\mathfrak{h}]]$ such that the Poisson bracket on A_0 defined by (6.2.1) is equal to the bracket given a priori.
This approach to quantization is as old as quantum mechanics. It was explained to mathematicians by F. A. Berezin, I. Vey, A. Lichnerowics, M. Flato, D. Sternheimer and others.

6. WEAK FADDEEV-TAKHTAJAN-VOLKOV ALGEBRAS

As it has been mentioned already in [1], the main tools which we use are difference equations, screening operators, Feigin's homomorphisms and adjoint actions and partial differential equations and Cartan matrices and ...

We know that from an abstract view $g = sl_{m+1}$ is an algebra related to the Cartan matrix (a_{ij}), where

$$a_{ij} = \begin{cases} 2 & \text{if } i = j \\ -1 & \text{if } |i-j| = 1 \\ 0 & \text{if } |i-j| > 1 \end{cases}$$

Now suppose that we have an infinite number of points in a definite discrete space such that we can assign them a proper coloring as follows So by letting

$$A_n = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \vdots \\ 0 & -1 & 2 & -1 & 0 \\ \vdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}$$

be the Cartan matrix of sl_{m+1} for $n \in \{1, 2, \cdots, m-1\}$, and so for sl_2 it will consist of just one row and one column, i.e. we have $A_1 = (2)$ and denote by $C[X]$ the skew polynomial ring on generators X_i labeled by $i \in \{-\infty, \cdots -1, 0, 1, \cdots, +\infty\}$ and defining $q-$commutation relations

$$X_i X_j = q^2 X_j X_i \ \text{ for if } i \leq j \tag{6.1}$$

$$\underset{X_{-\infty}}{\circ} - - - - - \cdots - \underset{X_1}{\circ} - \underset{X_2}{\circ} - \underset{X_3}{\circ} - \cdots - - - - - - \underset{X_{+\infty}}{\circ}$$

Definition 6.1. Let's define our Poisson bracket as follows in the case of sl_2:

$$\begin{cases} \{X_i, X_j\} := 2X_i X_j & \text{if } i < j \\ \{X_i, X_i\} := 0 \end{cases}$$

The main problem is to find solutions of the system of difference equations from infinite number of non-commutative variables in quantum case and commutative variables in classical case. It is significant that commutation relations (2.1) depend on the sign of the difference $(i-j)$ only and is based on our Cartan matrix. We should try to find all solutions of the system:

$$(**)\begin{cases}\mathfrak{D}_x^{(n)} \triangleleft \tau_1 = 0\\ H_x^{(n)} \triangleleft \tau_1 = 0\end{cases}$$

And let us define our system of variables as follows

$$\begin{array}{cccccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & X_1^{(11)} & X_1^{(21)} & X_1^{(31)} & X_1^{(41)} & \cdots \\ \cdots & X_2^{(12)} & X_2^{(22)} & X_2^{(32)} & X_2^{(42)} & \cdots \\ \cdots & X_3^{(13)} & X_3^{(23)} & X_3^{(33)} & X_3^{(43)} & \cdots \\ \cdots & X_4^{(14)} & X_4^{(24)} & X_4^{(34)} & X_4^{(44)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

And let us equip this system of variables with lexicographic ordering, i.e. $j_{k_m}i < j_{k_n}i$ if $j_{k_m} < j_{k_n}$. And we need this kind of ordering because we have different kind of set of variables with a proper coloring such that each set has its own color different of its neighbors.

We have $\tau_1 := \tau_1[\cdots, X_1^{(11)}, X_1^{(21)}, X_1^{(31)}, \cdots, X_2^{(12)}, X_2^{(22)}, X_2^{(32)}, \cdots]$ is a multi-variable function depend on $\{x_i^{(ji)}\}$'s for $i, j \in \{-\infty, \cdots, 1, \cdots, n, \cdots, +\infty\}$ and $\mathfrak{D}_x^{(n)}$ comes from

$$\{S_{X_i^{ji}}, \tau_1\}_p = S_{X_i^{ji}}\tau_1 - p^{\deg\tau_1 <\alpha_i,\alpha_j>}\tau_1 S_{X_i^{ji}} \tag{6.2}$$

where $<\alpha_i, \alpha_j>= a_{ij}$ is related to our Cartan matrix and $S_{X_i^{ji}}$ is an screening operator on one of our variable sets, i.e. $S_{X_i^{ji}} = \sum_{j\in\mathbb{Z}} X_i^{ji}$. Then we will obtain the whole set of solutions by using the following shift operator:

$$\tau_2 = \tau_1[X_1^{(11)} \to X_1^{(21)}, X_1^{(21)} \to X_1^{(31)}, \cdots],$$

$$\tau_3 = \tau_2[X_1^{(21)} \to X_1^{(31)}, X_1^{(31)} \to X_1^{(41)}, \cdots] \tag{6.3}$$

$$\vdots$$

Definition 6.2. Let us define our lattice W-algebra based on its generators according to [4] [1];
Generators of lattice W-algebra associated with simple Lie algebra g constitute the functional basis of the space of invariants

$$\tau_i := \mathrm{Inv}_{U_q(n_+)}(\mathbb{C}_q[X_i^{ji} | i \in \mathbb{Z}]) \tag{6.4}$$

with additional requirements

$$H_{X_i^{ji}}(\tau_i) = 0 \quad \text{and} \quad D_{X_i^{ji}}(\tau_i) = 0 \tag{6.5}$$

where $H_{X_i^{ji}}$ and $D_{X_i^{ji}}$ will be specified later.

Equation (2.4) means that the generators have to satisfy in quantum Serre relations and the first equation (2.5) means that they should have zero degree.
Here in this paper we just will work on $g = \mathrm{sl}_n$ and will use $\tau_i^{(n)}$ instead of τ_i.
Where (n) sits for n in sl_n.

6.1. **Lattice W_2 algebra.** Let us first consider the sl_2 case for to open out the concepts of (2.2) and (2.4). And also for to simplifying out notations, let us consider our set of variables as $X_i := X_i^{ji}$.
And as it has shown in [1], it is enough just to work with $S_{X_i^{ji}} =: S_{X_i} = \sum_{i=1}^{3} X_i$, because the other parts for $i > 3$ and $i < 1$ will tend to zero.
By setting $q = e^{-\mathfrak{h}}$, for the Planck constant $\mathfrak{h}$, we will try to find generators of our lattice W_2-algebra, in the case of sl_2.
First step:
First let us try to find $D_X^{(2)}$.
For to do this and for simplicity, we will set $\tau_1 := \tau_1[\cdots, X_1, X_2, X_3, , \cdots]$.
And as it has been defined already, we have

$$\begin{aligned} D_X^{(2)} &:= \{S_{X_i}, \tau_1\} \\ &= \{X_1 + X_2 + X_3, \tau_1\} \\ &= \{X_1, \tau_1\} + \{X_2, \tau_1\} + \{X_3, \tau_1\} \end{aligned}$$

$$= (D_{X_1} + D_{X_2} + D_{X_3})\tau_1 \tag{6.6}$$

Now for to understand what is (2.6), we note that partial $D_{X_i} = \{X_i, \tau_1\}$ and also note that our function $\tau_1[\cdots, X_1, X_2, X_3, \cdots]$ is a polynomial function consist of powers of X_i. What I mean is that, it is enough to find D_{X_i} on just powers of X_j for different values of $j \in \mathbb{Z}$.
So

$$(2.6) = \sum_j (\{X_1, X_j^n\} + \{X_2, X_j^n\} + \{X_3, X_j^n\}) \tag{6.7}$$

Where according to rules which has been showed out in [1], we have

$$\begin{aligned} \{X_1, X_j^n\} &= X_1 X_j^n - q^{2n} X_j^n X_1 \\ &= \begin{cases} 0, & \text{if } j > 1 \\ (1 - q^{4n}) X_1 X_j^n, & \text{if } j < 1 \\ (1 - q^{2n}) X_1 X_j^n, & \text{if } j = 1 \end{cases} \end{aligned}$$

Where by setting $q = e^{-\mathfrak{h}}$ and letting $\mathfrak{h} = 1$ at the end, we will have:

First case: $j > 1$;

$$\{X_1, X_j^n\} = 0;$$

Second case: $j<1$;

$$\begin{aligned}\{X_1, X_j^n\} &= (1-e^{-4n\mathfrak{h}})X_1, X_j^n\\ &\sim (1-(1-4n\mathfrak{h}))X_1, X_j^n\\ &= 4n\mathfrak{h}X_1, X_j^n \sim 4nX_1, X_j^n\\ &= 4X_1X_j\tfrac{\partial X_j^n}{\partial X_j}.\end{aligned}$$

Third case: $j=1$;

$$\begin{aligned}\{X_1, X_1^n\} &= (1-q^{2n})X_1X_1^n\\ &= (1-e^{-2n\mathfrak{h}})X_1X_1^n\\ &\sim (1-(1-2n\mathfrak{h}))X_1X_1^n\\ &= 2n\mathfrak{h}X_1X_1^n \sim 2nX_1X_1^n\\ &= 2X_1^2\tfrac{\partial X_1^n}{\partial X_1}.\end{aligned}$$

And so we have

$$\begin{aligned}(2.7) = &\{X_1, X_1^n\} + \textstyle\sum_{j<1}\{X_1, X_j^n\} + \sum_{j>1}\{X_1, X_j^n\}\\ &+\{X_2, X_1^n\} + \textstyle\sum_{j<2}\{X_2, X_j^n\} + \sum_{j>2}\{X_2, X_j^n\}\\ &+\{X_3, X_1^n\} + \textstyle\sum_{j<3}\{X_3, X_j^n\} + \sum_{j>3}\{X_3, X_j^n\}\\ &= 2X_1^2\tfrac{\partial}{\partial X_1} + 0 + 0\\ &+2X_2^2\tfrac{\partial}{\partial X_2} + 4X_2X_1\tfrac{\partial}{\partial X_1} + 0\\ &+2X_3^2\tfrac{\partial}{\partial X_3} + 4X_3X_2\tfrac{\partial}{\partial X_2} + 4X_3X_1\tfrac{\partial}{\partial X_1}\\ &= 2X_1(X_1{+}2X_2{+}2X_3)\tfrac{\partial}{\partial X_1}{+}2X_2(X_2{+}2X_3)\tfrac{\partial}{\partial X_2}\\ &+2X_3^2\tfrac{\partial}{\partial X_3}.\end{aligned}$$

So we found $D_X^{(2)}$ which is as follows and we can omit 2, because finally we will make the action equal to zero and we can cancel 2 out from both sides. So we have

$$D_X^{(2)} = X_1(X_1+2X_2+2X_3)\frac{\partial}{\partial X_1} + X_2(X_2+2X_3)\frac{\partial}{\partial X_2} + X_3^2\frac{\partial}{\partial X_3} \tag{6.8}$$

Second step:

Now we will try to find $H_X^{(2)}$.

For to find $H_X^{(2)}$, we note that it resembles degree of our polynomial function. So if for example $H_X^{(2)}$ acts on $X_1^nX_2^mX_3^l$, then we should get $(n{+}m{+}l)$. So let us define

$$H_X^{(2)} := \sum_i X_i\frac{\partial}{\partial X_i} \tag{6.9}$$

and then we have;

$$\begin{aligned}H_X^{(2)}(X_1^nX_2^mX_3^l) &= (\textstyle\sum_i X_i\frac{\partial}{\partial_{X_i}})(X_1^nX_2^mX_3^l)\\ &= \textstyle\sum_i X_i\frac{\partial X_1^nX_2^mX_3^l}{\partial_{X_i}}\\ &= X_1\tfrac{\partial X_1^nX_2^mX_3^l}{\partial_{X_1}} + X_2\tfrac{\partial X_1^nX_2^mX_3^l}{\partial_{X_2}} + X_3\tfrac{\partial X_1^nX_2^mX_3^l}{\partial_{X_3}}\\ &= nX_1^nX_2^mX_3^l + mX_1^nX_2^mX_3^l + lX_1^nX_2^mX_3^l\\ &= (n+m+l)X_1^nX_2^mX_3^l.\end{aligned}$$

Which gives us

$$H_X^{(2)}(X_1^n X_2^m X_3^l) = (n+m+l)X_1^n X_2^m X_3^l$$

and in the other side we have

$$\begin{aligned}(n+m+l)X_1^n X_2^m X_3^l &= nX_1X_1^{n-1}X_2^mX_3^l + mX_1^nX_2X_2^{m-1}X_3^l + lX_1^nX_2^mx_3X_3^{l-1}\\ &= X_1\frac{X_2^mX_3^l\partial X_1^n}{\partial_{X_1}} + X_2\frac{X_1^nX_3^l\partial X_2^m}{\partial_{X_2}} + X_3\frac{X_1^nX_2^m\partial X_3^l}{\partial_{X_3}}\\ &= X_1\frac{\partial}{\partial_{X_1}} + X_2\frac{\partial}{\partial_{X_2}} + X_3\frac{\partial}{\partial_{X_3}}\end{aligned}$$

Which gives us

$$(n+m+l)X_1^n X_2^m X_3^l = \sum_i X_i \frac{\partial}{\partial X_i}$$

And it shows that (2.9) is well defined.
Now the only thing which remains is just to find the solutions of the following system of 2-linear homogeneous equations in one unknown τ_1:

(6.10)

$$\begin{cases}(X_1(X_1+2X_2+2X_3)\frac{\partial}{\partial X_1} + X_2(X_2+2X_3)\frac{\partial}{\partial X_2} + X_3^2\frac{\partial}{\partial X_3})\tau_1[\cdot\cdot\cdot, X_1, X_2, X_3,\\ \quad \cdots] = 0,\\ (X_1\frac{\partial}{\partial_{X_1}} + X_2\frac{\partial}{\partial_{X_2}} + X_3\frac{\partial}{\partial_{X_3}})\tau_1[\cdots, X_1, X_2, X_3, \cdots] = 0;\end{cases}$$

Now the goal is to find such $\tau_1[\cdots, X_1, X_2, X_3, \cdots]$ which satisfies in our system of equations (2.10).
The second equation ensures that the solution has degree 0 and also the partial differentials will fix us a multi-variable function dependent on just X_1, X_2, X_3.
The system of PDEs (2.10) can be solved using the procedure described in Chapter V, Sec IV of [5]. And for more details please check out appendix A. And after all it became clear that the system (2.10) has only one functional dependent nontrivial solution:
(6.11)

$$\tau_1^{(2)}[X_1, X_2, X_3] = \frac{(X1+X2)(X2+X3)}{X2(X1+X2+X3)} = \frac{(\sum_{1\le i_1\le 2} X_{i_1}^{(1)})(\sum_{1\le i_1\le 2} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{1\le i_1\le 3} X_{i_1}^{(1)})}$$

And again as before, (2) goes back to 2 in Sl_2 and 1 is a default index which will be used later it for to employ shifting operator.
According to the number of variables, we will have two shifts and then everything will be in a loop.
So here in sl_2 case we have three solutions for our system of linear equations (2.10) which belong to the fraction ring of polynomial functions.

$$(6.12)\qquad \begin{cases}\tau_1^{(2)}[X_1, X_2, X_3] = \frac{(\sum_{1\le i_1\le 2} X_{i_1}^{(1)})(\sum_{1\le i_1\le 2} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{1\le i_1\le 3} X_{i_1}^{(1)})};\\ \tau_2^{(2)}[X_2, X_3, X_4] = \frac{(\sum_{2\le i_1\le 3} X_{i_1}^{(1)})(\sum_{2\le i_1\le 3} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{2\le i_1\le 4} X_{i_1}^{(1)})};\\ \tau_3^{(2)}[X_3, X_4, X_5] = \frac{(\sum_{3\le i_1\le 4} X_{i_1}^{(1)})(\sum_{3\le i_1\le 4} X_{i_1+1}^{(1)})}{X_2^{(1)}(\sum_{3\le i_1\le 5} X_{i_1}^{(1)})};\end{cases}$$

And as it already has mentioned we go to define our non-commutative Poisson algebra according to definition of Poisson brackets given by Poisson himself [6] with this difference that here we work on $q-$commutative ring $\frac{\mathbb{C}[X_i^{ji}]}{X_i^{ji}X_k^{jk} - q^{<\alpha_i,\alpha_k>}X_k^{jk}X_i^{ji}}$, based on the generators which are the solutions

of PDEs system (2.10).
For to do this we will use the following bracket based on

$$\tau_i^{(n)}[\cdots, X_1, X_2, X_3, \cdots] \quad \text{and} \quad \tau_j^{(n)}[\cdots, X_1, X_2, X_3, \cdots]$$

So we have to define our Poisson brackets as what comes in follow:

$$F_j^{(n)} := \{\tau_i^{(n)}, \tau_j^{(n)}\} = \sum_i \frac{\partial \tau_i^{(n)}}{\partial X_i} \sum_j \frac{\partial \tau_j^{(n)}}{\partial X_j} \{X_i, X_j\} \tag{6.13}$$

Where $\{X_i, X_j\}$ is our previously defined Poisson bracket on our set of variables.
For instance in the case of sl_2 we have

$$\begin{aligned}
\{\tau_1^{(2)}, \tau_2^{(2)}\} &= \Big(\frac{\partial \tau_1^{(2)}}{\partial X_1}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_1, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3}\{X_1, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_1, X_4\}\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_2}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_2, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3}\{X_2, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_2, X_4\}\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_3}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_3, X_2\} + \frac{\partial \tau_2^{(2)}}{\partial X_3}\{X_3, X_3\} + \frac{\partial \tau_2^{(2)}}{\partial X_2}\{X_3, X_4\}\Big) \\
&= \Big(\frac{\partial \tau_1^{(2)}}{\partial X_1}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}(2X_1X_2) + \frac{\partial \tau_2^{(2)}}{\partial X_3}(2X_1X_3) + \frac{\partial \tau_2^{(2)}}{\partial X_2}(2X_1X_4)\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_2}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}(0) + \frac{\partial \tau_2^{(2)}}{\partial X_3}(2X_2X_3) + \frac{\partial \tau_2^{(2)}}{\partial X_2}(2X_2X_4)\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_3}\Big)\Big(\frac{\partial \tau_2^{(2)}}{\partial X_2}(-2X_3X_2) + \frac{\partial \tau_2^{(2)}}{\partial X_3}(0) + \frac{\partial \tau_2^{(2)}}{\partial X_2}(2X_3X_4)\Big) \\
&= 2\frac{X_1X_2^2X_3^2X_4(X_1+X_2+X_3+X_4)}{(X_1+X_2)^2(X_2+X_3)^3(X_3+X_4)^2}
\end{aligned}$$

So we have

$$F_2^{(2)} = \{\tau_1^{(2)}, \tau_2^{(2)}\} = \frac{2X_1X_2^2X_3^2X_4(X_1+X_2+X_3+X_4)}{(X_1+X_2)^2(X_2+X_3)^3(X_3+X_4)^2} \tag{6.14}$$

And it is enough to find our brackets on just first generator, because then we are able to find other brackets based on the other generators, so for $\tau_3^{(2)}$ we have in a same process as follows

$$\begin{aligned}
F_3^{(2)} &= \{\tau_1^{(2)}, \tau_3^{(2)}\} \\
&= \Big(\frac{\partial \tau_1^{(2)}}{\partial X_1}\Big)\Big(\frac{\partial \tau_3^{(2)}}{\partial X_3}\{X_1, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4}\{X_1, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5}\{X_1, X_5\}\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_2}\Big)\Big(\frac{\partial \tau_3^{(2)}}{\partial X_3}\{X_2, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4}\{X_2, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5}\{X_2, X_5\}\Big) \\
&+\Big(\frac{\partial \tau_1^{(2)}}{\partial X_3}\Big)\Big(\frac{\partial \tau_3^{(2)}}{\partial X_3}\{X_3, X_3\} + \frac{\partial \tau_3^{(2)}}{\partial X_4}\{X_3, X_4\} + \frac{\partial \tau_3^{(2)}}{\partial X_5}\{X_3, X_5\}\Big) \\
&= \Big(\frac{\partial \tau_1^{(2)}}{\partial X_1}\Big)\Big(\frac{\partial \tau_3^{(2)}}{\partial X_3}(2X_1X_3) + \frac{\partial \tau_3^{(2)}}{\partial X_4}(2X_1X_4) + \frac{\partial \tau_3^{(2)}}{\partial X_5}(2X_1X_5)\Big)
\end{aligned}$$

$$+\Big(\frac{\partial\tau_1^{(2)}}{\partial X_2}\Big)\Big(\frac{\partial\tau_3^{(2)}}{\partial X_3}(2X_2X_3)+\frac{\partial\tau_3^{(2)}}{\partial X_4}(2X_2X_4)+\frac{\partial\tau_3^{(2)}}{\partial X_5}(2X_2X_5)\Big)$$

$$+\Big(\frac{\partial\tau_1^{(2)}}{\partial X_3}\Big)\Big(\frac{\partial\tau_3^{(2)}}{\partial X_3}(0)+\frac{\partial\tau_3^{(2)}}{\partial X_4}(2X_3X_4)+\frac{\partial\tau_3^{(2)}}{\partial X_5}(2X_3X_5)\Big)$$

$$= \frac{-2X_1X_2X_3^2X_4X_5}{(X_1+X_2)(X_2+X_3)^2(X_3+X_4)^2(X_4+X_5)} \tag{6.15}$$

We have to note that we almost are done with our Poisson algebra in sl_2 case, but for our further plan i.e. to find our Volterra system, the differential-difference chain of non-linear equations

$$\begin{cases} H=\sum\limits_i[\ln(\tau_i)]; \\ \dot{\tau_j}=\{\tau_j,H\}=\tau_j\times\sum\limits_i\Gamma_i; \end{cases} \tag{6.16}$$

Where Γ_i stands for $\frac{\tau_1,\tau_i}{\tau_1\tau_i}$ [4]. Which means that we have to write down the brackets $\{\tau_1,\tau_i\}$ in terms of their decompositions to τ_j's for $1\le j\le i$.
So we need to write it as decomposition of our generators and it will be done by using the Mathematica coding which we have produced in Appendix C.
And the result will be as follows
(6.17)

$$\begin{cases} F_2^{(2)}=\{\tau_1^{(2)},\tau_2^{(2)}\}=2(1-\tau_1^{(2)})(1-\tau_2^{(2)})(-1+\tau_1^{(2)}+\tau_2^{(2)}); \\ F_3^{(2)}=\{\tau_1^{(2)},\tau_3^{(2)}\}=-2(1-\tau_1^{(2)})(1-\tau_2^{(2)})(1-\tau_3^{(2)}); \\ F_i^{(2)}=\{\tau_1^{(2)},\tau_i^{(2)}\}=0 \qquad \text{for } |i-1|\ge 3; \end{cases}$$

This result are weaker than Faddeev-Takhtajan-Volkov algebra which has mentioned in [4] and if we continue this for sl_3, then we will have again a weak version of what that has mentioned in [4].

6.2. **Lattice W_3 algebra.** In this case we will use the following defined Poisson bracket based on Cartan matrix $A_2=\begin{bmatrix} 2 & -1 \\ -1 & 2\end{bmatrix}$.
But for to do this according to our previous ordering and list of variables, let us for simplicity set our variables as follows
Set $X_i^{(1i)}:=X_i$ and $X_i^{(2i)}:=Y_i$.

Definition 6.3. Let's define our Poisson bracket as follows in the case of sl_3:

$$\begin{cases} \{X_i,X_j\}:=2X_iX_j & \text{if } i<j; \\ \{Y_i,Y_j\}:=2Y_iY_j & \text{if } i<j; \\ \{X_i,X_i\}:=0; \\ \{Y_i,Y_i\}:=0; \\ \{X_i,Y_j\}:=X_iY_j & \text{if } i>j; \\ \{X_i,Y_j\}:=-X_iY_j & \text{if } i\le j; \end{cases} \tag{6.18}$$

As it comes out, here our set of variables will be as follows:

$$\underset{X_{-\infty}}{\circ} - \cdots - \underset{X_1}{\circ} - \underset{Y_1}{\circ} - \underset{X_2}{\circ} - \underset{Y_2}{\circ} - \underset{X_3}{\circ} - \underset{Y_3}{\circ} - \cdots - \underset{X_{n-2}}{\circ} - \underset{Y_{n-2}}{\circ} - \underset{X_{n-1}}{\circ} - \underset{Y_{n-1}}{\circ} - \underset{X_n}{\circ} - \underset{Y_n}{\circ} - \cdots - \underset{X_{+\infty}}{\circ}$$

And instead of (2.1) we will have the following $q-$commutation relations

$$\begin{cases} X_iX_j = q^2X_jX_i & \text{if } i \leq j; \\ Y_iY_j = q^2Y_jY_i & \text{if } i \leq j; \\ X_iY_j = q^{-1}Y_jX_i & \text{if } i \leq j; \end{cases} \tag{6.19}$$

And we will get the following equations in a same manner as in sl_2:

First case: $i < j$;

$$\begin{aligned} \{X_i, Y_j^n\} &= X_iY_j^n - q^{-n}Y_j^nX_i \\ &= X_iY_j^n - q^0X_iX_j^n \\ &= 0 \end{aligned}$$

Second case: $i \geq j$;

$$\begin{aligned} \{X_i, Y_j^n\} &= X_iY_j^n - q^{-n}Y_j^nX_i \\ &= (1 - q^{-2n})X_iY_j^n \\ &= (1 - e^{2n\mathfrak{h}})X_iY_j^n \\ &\sim (1 - (1 + 2n\mathfrak{h}))X_iY_j^n \\ &= -2n\mathfrak{h}X_iY_j^n \\ &\sim -2nX_iY_j^n \\ &= -2X_iY_j\frac{\partial Y_j^n}{\partial Y_j} \end{aligned}$$

$$\begin{cases} \{X_i, X_j^n\} = 0 & \text{if } i \leq j; \\ \{X_i, X_j^n\} = 4X_iX_j\frac{\partial X_j^n}{\partial X_j} & \text{if } i > j; \\ \{X_i, Y_j^n\} = 0 & \text{if } i < j; \\ \{X_i, Y_j^n\} = -2X_iY_j\frac{\partial Y_j^n}{\partial Y_j} & \text{if } i \geq j; \\ \{Y_j, X_i^n\} = -2Y_jX_i\frac{\partial X_i^n}{\partial X_i} & \text{if } i \leq j; \end{cases} \tag{6.20}$$

According to (2.20) we will try to find $H_X^{(3)}$ as what comes in follow

$$\begin{aligned} &\{X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}, X_0\} \\ &= X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}X_0 - X_0X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3} \\ &= (1 - q^{2\alpha_1+2\alpha_2+2\alpha_3-\beta_1-\beta_2-\beta_3})X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}X_0 \\ &\sim (1 - (1 - n\mathfrak{h}(2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3)))X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}X_0 \\ &= (2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3)n\mathfrak{h}X_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}X_0 \\ &\sim (2\alpha_1 + 2\alpha_2 + 2\alpha_3 - \beta_1 - \beta_2 - \beta_3)nX_1^{\alpha_1}X_2^{\alpha_2}X_3^{\alpha_3}Y_1^{\beta_1}Y_2^{\beta_2}Y_3^{\beta_3}X_0 \\ &= (2X_1\frac{\partial}{\partial X_1} + 2X_2\frac{\partial}{\partial X_2} + 2X_3\frac{\partial}{\partial X_3} - Y_1\frac{\partial}{\partial Y_1} - Y_2\frac{\partial}{\partial Y_2} - Y_3\frac{\partial}{\partial Y_3})\tau_1^{(3)}. \end{aligned}$$

Now let us as usual suppose $i > j$ and then we will define the following same quantities

Here we have for X_is:

$$\begin{aligned}
{}_{X_j}D_{X_i} &:= \{X_i, X_j^n\} \\
&= X_iX_j^n - q^{2n}X_j^nX_i \\
&= (1-q^{4n})X_iX_j^n \\
&= (1-e^{-4n\mathfrak{h}})X_iX_j^n \\
&\sim (1-(1-4n\mathfrak{h}))X_iX_j^n \\
&= 4n\mathfrak{h}X_iX_j^n \\
&\sim 4nX_iX_j^n \\
&= 4nX_iX_j\tfrac{\partial X_j^n}{\partial X_j}.
\end{aligned}$$

And the same will be for Y_is.
And for the different quantities X_i and Y_js we have:

First case: for $i > j$;

$$\begin{aligned}
{}_{Y_j}D_{X_i} &:= \{X_i, Y_j^n\} \\
&= ad_{X_i}Y_j^n \\
&= X_iY_j^n - q^{-n}Y_j^nX_i \\
&= (1-q^{-2n})X_iY_j^n \\
&= (1-e^{-2n\mathfrak{h}})X_iY_j^n \\
&\sim (1-(1-2n\mathfrak{h}))X_iY_j^n \\
&= 2n\mathfrak{h}X_iY_j^n \\
&\sim 2nX_iY_j^n \\
&= 2X_iY_j\tfrac{\partial Y_j^n}{\partial Y_j}.
\end{aligned}$$

Second case: for $i \le j$;
According to what has just mentioned we have

$$\begin{aligned}
{}_{Y_j}D_1^Y &:={}_{Y_j} D_{Y_1} \\
&= 4Y_1Y_j\tfrac{\partial Y_j^n}{\partial Y_j}.
\end{aligned}$$

And

$$\begin{aligned}
{}_{Y_1}D_1^Y &:={}_{Y_1} D_{Y_1} \\
&= 2Y_1^2\tfrac{\partial Y_1^n}{\partial Y_1}.
\end{aligned}$$

And in a same way we can find the desired results for ${}_{Y_j}D_2^Y$ and ${}_{Y_j}D_3^Y$,
So let us define

$$\begin{cases}
{}_YD_1^Y :={}_{Y_1} D_1^Y +{}_{Y_j}^{j<1} D_1^Y +{}_{Y_j}^{j>1} D_1^Y; \\
{}_YD_2^Y :={}_{Y_2} D_2^Y +{}_{Y_j}^{j<2} D_2^Y +{}_{Y_j}^{j>2} D_2^Y; \\
{}_YD_3^Y :={}_{Y_3} D_3^Y +{}_{Y_j}^{j<3} D_3^Y +{}_{Y_j}^{j>3} D_3^Y;
\end{cases} \tag{6.21}$$

And then we will have

$${}_YD_1^Y = Y_1^2\frac{\partial}{\partial Y_1} + \sum_{j<1} 2Y_1Y_j\frac{\partial}{\partial Y_j} + 0$$

And

$${}_YD_2^Y = Y_2^2\frac{\partial}{\partial Y_2} + \sum_{j<2} 2Y_2Y_j\frac{\partial}{\partial Y_j} + 0$$

And

$$ {}_Y D_3^Y = Y_3^2 \frac{\partial}{\partial Y_3} + \sum_{j<2} 2Y_3 Y_j \frac{\partial}{\partial Y_j} + 0 $$

And finally we get

$$ \begin{aligned} {}_Y D_Y^{(3)} &:= {}_Y D_1 + {}_Y D_2 + {}_Y D_3 \\ &= Y_1(Y_1+2Y_2+2Y_3)\tfrac{\partial}{\partial Y_1} + Y_2(Y_2+2Y_3)\tfrac{\partial}{\partial Y_2} + Y_3^2 \tfrac{\partial}{\partial Y_3}. \end{aligned} $$

For $j \geq 1$ we have

$$ \begin{aligned} {}_{X_1} D_{Y_j} &:= \{Y_j, X_1^n\} \\ &= Y_j X_1^n - q^{-n} X_1^n Y_j \\ &= (1 - q^{-2n}) Y_j X_1^n \\ &= (1 - e^{2n\mathfrak{h}}) Y_j X_1^n \\ &\sim (1 - (1 + 2n\mathfrak{h})) Y_j X_1^n \\ &= -2n\mathfrak{h} Y_j X_1^n \\ &\sim -2n Y_j X_1^n \\ &= -2 Y_j X_1 \tfrac{\partial}{\partial X_1} \\ &= -2X_1(Y_1 + Y_2 + Y_3)\tfrac{\partial}{\partial X_1}. \end{aligned} $$

For $j \geq 2$ we have

$$ \begin{aligned} {}_{X_2} D_{Y_j} &:= \{Y_j, X_2^n\} \\ &= Y_j X_2^n - q^{-n} X_2^n Y_j \\ &= (1 - q^{-2n}) Y_j X_2^n \\ &= (1 - e^{2n\mathfrak{h}}) Y_j X_2^n \\ &\sim (1 - (1 + 2n\mathfrak{h})) Y_j X_2^n \\ &= -2n\mathfrak{h} Y_j X_2^n \\ &\sim -2n Y_j X_2^n \\ &= -2 Y_j X_2 \tfrac{\partial}{\partial X_2} \\ &= -2X_2(Y_2 + Y_3)\tfrac{\partial}{\partial X_2}. \end{aligned} $$

For $j \geq 3$ we have

$$ \begin{aligned} {}_{X_3} D_{Y_j} &:= \{Y_j, X_3^n\} \\ &= Y_j X_3^n - q^{-n} X_3^n Y_j \\ &= (1 - q^{-2n}) Y_j X_3^n \\ &= (1 - e^{2n\mathfrak{h}}) Y_j X_3^n \\ &\sim (1 - (1 + 2n\mathfrak{h})) Y_j X_3^n \\ &= -2n\mathfrak{h} Y_j X_3^n \\ &\sim -2n Y_j X_3^n \\ &= -2 Y_j X_3 \tfrac{\partial}{\partial X_3} \\ &= -2X_3 Y_3 \tfrac{\partial}{\partial X_3}. \end{aligned} $$

And after all these, let us define

$$ \begin{aligned} {}_X D_Y^{(3)} &:= {}_{X_1} D_{Y_j} + {}_{X_2} D_{Y_j} + {}_{X_3} D_{Y_j} \\ &= -2X_1(Y_1 + Y_2 + Y_3)\tfrac{\partial}{\partial X_1} - 2X_2(Y_2 + Y_3)\tfrac{\partial}{\partial X_2} \\ &\quad -2X_3 Y_3 \tfrac{\partial}{\partial X_3}. \end{aligned} $$

And finally let us define

$$ \begin{aligned} D_Y^{(3)} &:= {}_Y D_Y^{(3)} + {}_X D_Y^{(3)} \\ &= Y_1(Y_1 + 2Y_2 + 2Y_3)\tfrac{\partial}{\partial Y_1} + Y_2(Y_2 + 2Y_3)\tfrac{\partial}{\partial Y_2} \\ &\quad + Y_3^2 \tfrac{\partial}{\partial Y_3} - 2X_1(Y_1 + Y_2 + Y_3)\tfrac{\partial}{\partial X_1} - 2X_2(Y_2 \end{aligned} $$

$$+Y_3)\frac{\partial}{\partial X_2} - 2X_3Y_3\frac{\partial}{\partial X_3}.$$

Next step:

Now let us try to find $D_X^{(3)}$:

For $i > 1$, let us define ${}_{Y_1}D_{X_i}$ as in what comes in follow:

$$\begin{aligned}
{}_{Y_1}D_{X_i} &:= \{X_i, Y_1^n\} \\
&= X_iY_1^n - q^{-n}Y_1^nX_i \\
&= (1-q^{-2n})X_iY_1^n \\
&= (1-e^{2n\mathfrak{h}})X_iY_1^n \\
&\sim (1-(1+2n\mathfrak{h}))X_iY_1^n \\
&= -2n\mathfrak{h}X_iY_1^n \\
&\sim -2nX_iY_1^n = -2X_iY_1\frac{\partial}{\partial Y_1} \\
&= -2Y_1(X_2+X_3)\frac{\partial}{\partial Y_1}.
\end{aligned}$$

For $i > 2$ we have

$$\begin{aligned}
{}_{Y_2}D_{X_i} &:= \{X_i, Y_2^n\} \\
&= X_iY_2^n - q^{-n}Y_2^nX_i \\
&= (1-q^{-2n})X_iY_2^n \\
&= (1-e^{2n\mathfrak{h}})X_iY_2^n \\
&\sim (1-(1+2n\mathfrak{h}))X_iY_2^n \\
&= -2n\mathfrak{h}X_iY_2^n \\
&\sim -2nX_iY_2^n \\
&= -2X_iY_2\frac{\partial}{\partial Y_2} \\
&= -2Y_2X_3\frac{\partial}{\partial Y_2}.
\end{aligned}$$

For $i > 3$ we have 0.

Let us again have the following definitions

$${}_{Y_1}D_2^X :={}_{Y_1} D_{X_2} = -2Y_1X_2\frac{\partial Y_1^n}{\partial Y_1};$$

$${}_{Y_1}D_3^X :={}_{Y_1} D_{X_3} = -2Y_1X_3\frac{\partial Y_1^n}{\partial Y_1};$$

$${}_{Y_2}D_3^X :={}_{Y_2} D_{X_3} = -2Y_2X_3\frac{\partial Y_1^n}{\partial Y_2};$$

Now let us define

$${}_XD_Y^{(3)} :={}_{Y_1} D_2^X +_{Y_1} D_3^X +_{Y_2} D_3^X = -Y_1(X_2+X_3)\frac{\partial}{\partial Y_1} - Y_2X_3\frac{\partial}{\partial Y_2};$$

And now as before we have

$$\begin{aligned}
{}_{X_j}D_1^X &:={}_{X_j} D_{X_1} \\
&= 4X_1X_j\frac{\partial X_j^n}{\partial X_j}. \\
{}_{X_1}D_1^X &:={}_{X_1} D_{X_1} \\
&= 2X_1^2\frac{\partial X_1^n}{\partial X_1}.
\end{aligned}$$

And in a same way we are able to define for ${}_{X_j}D_2^X$ and ${}_{X_j}D_3^X$. So let us define

(6.22)
$$\begin{cases} {}_XD_1^X := {}_{X_1} D_1^X +_{X_j}^{j<1} D_1^X +_{X_j}^{j>1} D_1^X; \\ {}_XD_2^X := {}_{X_2} D_2^X +_{X_j}^{j<2} D_2^X +_{X_j}^{j>2} D_2^X; \\ {}_XD_3^X := {}_{X_3} D_3^X +_{X_j}^{j<3} D_3^X +_{X_j}^{j>3} D_3^X; \end{cases}$$

Then we will have

$${}_XD_1^X = X_1^2 \frac{\partial}{\partial X_1} + \sum_{j<1} 2X_1X_j \frac{\partial}{\partial X_j} + 0$$

And

$${}_XD_2^X = X_2^2 \frac{\partial}{\partial X_2} + \sum_{j<2} 2X_2X_j \frac{\partial}{\partial X_j} + 0$$

And

$${}_XD_3^X = X_3^2 \frac{\partial}{\partial X_3} + \sum_{j<2} 2X_3Y_j \frac{\partial}{\partial X_j} + 0$$

So we will have
$$\begin{aligned} {}_XD_X^{(3)} &:=_X D_1 +_X D_2 +_X D_3 \\ &= X_1(X_1+2X_2+2X_3)\tfrac{\partial}{\partial X_1} + X_2(X_2+2X_3)\tfrac{\partial}{\partial X_2} \\ &\quad + X_3^2\tfrac{\partial}{\partial X_3}. \end{aligned}$$

And therefore as in (2.10) we will have the following system of PDEs
(6.23)
$$\begin{cases} (X_1(X_1+2X_2+2X_3)\frac{\partial\tau_1^{(3)}}{\partial X_1} + X_2(X_2+2X_3)\frac{\partial\tau_1^{(3)}}{\partial X_2} + X_3^2\frac{\partial\tau_1^{(3)}}{\partial X_2} \\ -Y_1(X_1+X_2+X_3)\frac{\partial\tau_1^{(3)}}{\partial Y_1} - Y_2(X_2+X_3)\frac{\partial f}{\partial Y_2} - Y_3X_3\frac{\partial\tau_1^{(3)}}{\partial Y_3}) = 0; \\ (2X_1\frac{\partial\tau_1^{(3)}}{\partial X_1} + 2X_2\frac{\partial\tau_1^{(3)}}{\partial X_2} + 2X_3\frac{\partial\tau_1^{(3)}}{\partial X_3} - Y_1\frac{\partial\tau_1^{(3)}}{\partial Y_1} - Y_2\frac{\partial\tau_1^{(3)}}{\partial Y_2} - Y_3\frac{\partial\tau_1^{(3)}}{\partial Y_3}) = 0; \\ D_Y^{(3)} = (Y_1(Y_1+2Y_2+2Y_3)\frac{\partial\tau_1^{(3)}}{\partial Y_1} + Y_2(Y_2+2Y_3)\frac{\partial\tau_1^{(3)}}{\partial Y_2} + Y_3^2\frac{\partial\tau_1^{(3)}}{\partial Y_2} \\ -Y_1(X_1+X_2+X_3)\frac{\partial\tau_1^{(3)}}{\partial Y_1} - Y_2(X_2+X_3)\frac{\partial\tau_1^{(3)}}{\partial Y_2} - Y_3X_3\frac{\partial\tau_1^{(3)}}{\partial Y_3}) = 0; \\ (2X_1\frac{\partial\tau_1^{(3)}}{\partial X_1} + 2X_2\frac{\partial\tau_1^{(3)}}{\partial X_2} + 2X_3\frac{\partial\tau_1^{(3)}}{\partial X_3} - Y_1\frac{\partial\tau_1^{(3)}}{\partial Y_1} - Y_2\frac{\partial\tau_1^{(3)}}{\partial Y_2} - Y_3\frac{\partial\tau_1^{(3)}}{\partial Y_3}) = 0; \end{cases}$$

And according to appendix A we have the following functional dependent nontrivial solution for the whole system of PDEs (2.23)

(6.24)
$$\tau_1^{(3)} = \frac{(\Sigma_{1\le i\le j\le 2} X_iY_j)(\Sigma_{1\le i\le j\le 2} X_{i+1}Y_{j+1})}{X_2Y_2(\Sigma_{1\le i\le j\le 3} X_iY_j)};$$

And again as before, (3) goes back to 3 in the Sl_3 and 1 is a default index which later we will use it for to employ our shifting operators.
According to the number of variables, we will have 6 shifts and then after that it will be in a loop.
So here in sl_3 case we have six solutions which belong to the fraction ring

of polynomial functions.
(6.25)

$$\begin{cases} \tau_1^{(3)}[X_1,Y_1,X_2,Y_2,X_3,Y_3] = \frac{X_2Y_2(X_3Y_3+X_2(Y_2+Y_3)+X_1(Y_1+Y_2+Y_3))}{(X_2Y_2+X_1(Y_1+Y_2))(X_3Y_3+X_2(Y_2+Y_3))}; \\ \tau_2^{(3)}[Y_1,X_2,Y_2,X_3,Y_3,X_4] = \frac{X_3Y_2(X_2Y_1+(X_3+X_4)(Y_1+Y_2)+X_4Y_3)}{(X_2Y_1+X_3(Y_1+Y_2))(X_3Y_2+X_4(Y_2+Y_3))}; \\ \tau_3^{(3)}[X_2,Y_2,X_3,Y_3,X_4,Y_4] = \frac{X_3y_3(X_4Y_4+X_3(Y_3+Y_4)+X_2(Y_2+Y_3+Y_4))}{(X_3Y_3+X_2(Y_2+Y_3))(X_4Y_4+X_3(Y_3+Y_4))}; \\ \tau_4^{(3)}[Y_2,X_3,Y_3,X_4,Y_4,X_5] = \frac{X_4Y_3(X_3Y_2+(X_4+X_5)(Y_2+Y_3)+X_5Y_4)}{(X_3Y_2+X_4(Y_2+Y_3))(X_4Y_3+X_5(Y_3+Y_4))}; \\ \tau_5^{(3)}[X_3,Y_3,X_4,Y_4,X_5,Y_5] = \frac{X_4Y_4(X_5Y_5+X_4(Y_4+Y_5)+X_3(Y_3+Y_4+Y_5))}{(X_4Y_4+X_3(Y_3+Y_4))(X_5Y_5+X_4(Y_4+Y_5))}; \\ \tau_6^{(3)}[Y_3,X_4,Y_4,X_5,Y_5,X_6] = \frac{X_5Y_4(X_4Y_3+(X_5+X_6)(Y_3+Y_4)+X_6Y_5)}{(X_4Y_3+X_5(Y_3+Y_4))(X_5Y_4+X_6(Y_4+Y_5))}; \end{cases}$$

Where $\tau_1^{(3)} := \tau_1^{(3)}[\cdots, X_1, Y_1, X_2, Y_2, X_3, Y_3 \cdots]$.
Again by setting $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and according to (2.4) we have to write down the following brackets as a composition of $\tau_i^{(3)}$s, because of algebra structure and it will be done by using Mathematica coding in appendix A.
(6.26)

$$\begin{cases} F_2^{(3)} = \{\tau_1^{(3)}, \tau_2^{(3)}\} = -2(1-\tau_1^{(3)})(1-\tau_2^{(3)})(\tau_1^{(3)}\tau_2^{(3)}); \\ F_3^{(3)} = \{\tau_1^{(3)}, \tau_3^{(3)}\} = 2(1-\tau_1^{(3)})(1-\tau_3^{(3)})(\tau_1^{(3)}\tau_2^{(3)} + \tau_2^{(3)}\tau_3^{(3)} - \tau_2^{(3)}); \\ F_4^{(3)} = \{\tau_1^{(3)}, \tau_4^{(3)}\} = -2(1-\tau_1^{(3)})(1-\tau_4^{(3)}) \\ \qquad (\tau_1^{(3)}\tau_2^{(3)} + \tau_2^{(3)}\tau_3^{(3)} + \tau_3^{(3)}\tau_4^{(3)} - \tau_1^{(3)} - \tau_2^{(3)} - \tau_3^{(3)} - \tau_4^{(3)} + 1); \\ F_5^{(3)} = \{\tau_1^{(3)}, \tau_5^{(3)}\} = 2(1-\tau_1^{(3)})(1-\tau_5^{(3)})(\tau_2^{(3)}\tau_3^{(3)} + \tau_3^{(3)}\tau_4^{(3)} - \tau_2^{(3)} \\ \qquad -\tau_3^{(3)} - \tau_4^{(3)} + 1); \\ F_6^{(3)} = \{\tau_1^{(3)}, \tau_6^{(3)}\} = -2(1-\tau_1^{(3)})(1-\tau_6^{(3)})(\tau_3^{(3)}\tau_4^{(3)} - \tau_4^{(3)} - \tau_3^{(3)} + 1); \\ F_i^{(3)} = \{\tau_1^{(3)}, \tau_i^{(3)}\} = 0 \qquad\qquad \text{for } |i-1| \geq 6; \end{cases}$$

6.3. **Lattice W_4 algebra; main generator.** In this case we will use the following defined Poisson bracket based on Cartan matrix

$$A_3 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

But for to do this according to our previous ordering and list of variables, and the same as what we din in sl_3 case, let us for simplicity set our set of variables as follows:
Set $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and so on.

Definition 6.4. Let's define our Poisson bracket as follows in the case of sl_3:

$$(6.27)\qquad \begin{cases} \{X_i, X_j\} := 2X_iX_j & \text{if } i < j; \\ \{Y_i, Y_j\} := 2Y_iY_j & \text{if } i < j; \\ \{Z_i, Z_j\} := 2Z_iZ_j & \text{if } i < j; \\ \{X_i, X_i\} := 0; \\ \{Y_i, Y_i\} := 0; \\ \{Z_i, Z_i\} := 0; \\ \{X_i, Y_j\} := X_iY_j & \text{if } i > j; \\ \{X_i, Y_j\} := -X_iY_j & \text{if } i \leq j; \\ \{X_i, Z_j\} := 0; \\ \{Y_i, Z_j\} := Y_iZ_j & \text{if } i > j; \\ \{Y_i, Z_j\} := -Y_iZ_j & \text{if } i \leq j; \end{cases}$$

As it comes out that, here our set of variables will be as follows:

$$\underset{-\infty}{\circ} - \cdots - \underset{X_1}{\circ} - \underset{Y_1}{\circ} - \underset{Z_1}{\circ} - \underset{X_2}{\circ} - \underset{Y_2}{\circ} - \underset{Z_2}{\circ} - \cdots - \underset{X_{n-1}}{\circ} - \underset{Y_{n-1}}{\circ} - \underset{Z_{n-1}}{\circ} - \underset{X_n}{\circ} - \underset{Y_n}{\circ} - \underset{Z_n}{\circ} - \cdots - \underset{+\infty}{\circ}$$

And instead of (2.1) we will have the following $q-$commutation relations for $j \in \{1, 2, 3\}$ and as always $i \in \{1, 2, 3\}$:

$$(6.28)\qquad \begin{cases} X_iX_j = q^2X_jX_i & \text{if } i \leq j \\ Y_iY_j = q^2Y_jY_i & \text{if } i \leq j \\ Z_iZ_j = q^2Z_jZ_i & \text{if } i \leq j \\ X_iY_j = q^{-1}Y_jX_i & \text{if } i \leq j \\ Y_iZ_j = q^{-1}Z_jY_i & \text{if } i \leq j \\ X_iZ_j = Z_jX_i \end{cases}$$

And by using the same approach as what we did for sl_2 and sl_3, it became clear that the equations $D^{(4)}_X$, $D^{(4)}_Y$ and $D^{(4)}_Z$ and also $H^{(4)}_X$, $H^{(4)}_Y$ and $H^{(4)}_Z$ will have the following forms:

(6.29)

$$\mathfrak{D}^{(4)}_X = X_1(X_1+2X_2+2X_3)\frac{\partial\tau^{(4)}_1}{\partial X_1}+X_2(X_2+2X_3)\frac{\partial\tau^{(4)}_1}{\partial X_2}+X_3^2\frac{\partial\tau^{(4)}_1}{\partial X_3}-Y_1(X_2+X_3)$$
$$\frac{\partial\tau^{(4)}_1}{\partial Y_1} - Y_2X_3\frac{\partial\tau^{(4)}_1}{\partial Y_2};$$

(6.30)

$$\mathfrak{D}^{(4)}_Y = Y_1(Y_1+2Y_2+2Y_3)\frac{\partial\tau^{(4)}_1}{\partial Y_1}+Y_2(Y_2+2Y_3)\frac{\partial\tau^{(4)}_1}{\partial Y_2}+Y_3^2\frac{\partial\tau^{(4)}_1}{\partial Y_3}-X_1(Y_1+Y_2+Y_3)$$
$$\frac{\partial\tau^{(4)}_1}{\partial X_1}-X_2(Y_2+Y_3)\frac{\partial\tau^{(4)}_1}{\partial X_2}-X_3Y_3\frac{\partial\tau^{(4)}_1}{\partial X_3}-Z_1(Y_2+Y_3)\frac{\partial\tau^{(4)}_1}{\partial z_1}-Z_2y_3\frac{\partial\tau^{(4)}_1}{\partial Z_2};$$

(6.31)

$$\mathfrak{D}^{(4)}_Z = Z_1(Z_1+2Z_2+2Z_3)\frac{\partial\tau^{(4)}_1}{\partial Z_1}+Z_2(Z_2+2Z_3)\frac{\partial\tau^{(4)}_1}{\partial Z_2}+Z_3^2\frac{\partial\tau^{(4)}_1}{\partial Z_3}-Y_1(Z_1+Z_2+Z_3)$$

$$\frac{\partial\tau_1^{(4)}}{\partial Y_1} - Y_2(Z_2+Z_3)\frac{\partial\tau_1^{(4)}}{\partial Y_2} - Y_3Z_3\frac{\partial\tau_1^{(4)}}{\partial Y_3};$$

(6.32)
$$H_X^{(4)} = 2X_1\frac{\partial\tau_1^{(4)}}{\partial X_1} + 2X_2\frac{\partial\tau_1^{(4)}}{\partial X_2} + 2X_3\frac{\partial\tau_1^{(4)}}{\partial X_3} - Y_1\frac{\partial\tau_1^{(4)}}{\partial Y_1} - Y_2\frac{\partial\tau_1^{(4)}}{\partial Y_2} - Y_3\frac{\partial\tau_1^{(4)}}{\partial Y_3};$$

(6.33)
$$H_Y^{(4)} = 2Y_1\frac{\partial\tau_1^{(4)}}{\partial Y_1} + 2Y_2\frac{\partial\tau_1^{(4)}}{\partial Y_2} + 2Y_3\frac{\partial\tau_1^{(4)}}{\partial Y_3} - X_1\frac{\partial\tau_1^{(4)}}{\partial X_1} - X_2\frac{\partial\tau_1^{(4)}}{\partial X_2} - X_3\frac{\partial\tau_1^{(4)}}{\partial X_3} - z_1\frac{\partial\tau_1^{(4)}}{\partial Z_1}$$
$$-Z_2\frac{\partial\tau_1^{(4)}}{\partial Z_2} - Z_3\frac{\partial\tau_1^{(4)}}{\partial Z_3};$$

(6.34) $$H_Z^{(4)} = 2Z_1\frac{\partial\tau_1^{(4)}}{\partial Z_1} + 2Z_2\frac{\partial\tau_1^{(4)}}{\partial Z_2} + 2Z_3\frac{\partial\tau_1^{(4)}}{\partial Z_3} - Y_1\frac{\partial\tau_1^{(4)}}{\partial Y_1} - Y_2\frac{\partial\tau_1^{(4)}}{\partial Y_2} - Y_3\frac{\partial\tau_1^{(4)}}{\partial Y_3};$$

And the functional dependent nontrivial solutions for the whole system of first order partial differential equation is as follows:

(6.35) $$\tau_1^{(4)} = \frac{(\Sigma_{1\le i\le j\le m\le 2}\ x_iy_jz_m)(\Sigma_{1\le i\le j\le m\le 2}\ x_{i+1}y_{j+1}z_{m+1})}{x_2y_2z_2(\Sigma_{1\le i\le j\le m\le 3}\ x_iy_jz_m)};$$

And again as before, (4) goes back to 4 in the Sl_4 and 1 is a default index which later we will use it for to employ our shifting operators.
According to the number of variables, we will have 9 shifts and then after that it will be in a loop.
So here in sl_4 case we have nine solutions:

$$\tau_1^{(4)} := \tau_1^{(4)}[X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_3, Y_3, Z_3];$$

$$\tau_2^{(4)} := \tau_1^{(4)}[X_1 \to Y_1, Y_1 \to Z_1, Z_1 \to X_2, X_2 \to Y_2, Y_2 \to Z_2, Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3];$$

$$\tau_3^{(4)} := \tau_2^{(4)}[Y_1 \to Z_1, Z_1 \to X_2, X_2 \to Y_2, Y_2 \to Z_2, Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4];$$

$$\tau_4^{(4)} := \tau_3^{(4)}[Z_1 \to X_2, X_2 \to Y_2, Y_2 \to Z_2, Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4];$$

$$\tau_5^{(4)} := \tau_4^{(4)}[X_2 \to Y_2, Y_2 \to Z_2, Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4, Y_4 \to Z_4];$$

$$\tau_6^{(4)} := \tau_5^{(4)}[Y_2 \to Z_2, Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4, Y_4 \to Z_4, Z_4 \to X_5];$$

$$\tau_7^{(4)} := \tau_6^{(4)}[Z_2 \to X_3, X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4, Y_4 \to Z_4, Z_4 \to X_5, X_5 \to Y_5];$$

$$\tau_8^{(4)} := \tau_7^{(4)}[X_3 \to Y_3, Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4, Y_4 \to Z_4, Z_4 \to X_5, X_5 \to Y_5, Y_5 \to Z_5];$$

$$\tau_9^{(4)} := \tau_8^{(4)}[Y_3 \to Z_3, Z_3 \to X_4, X_4 \to Y_4, Y_4 \to Z_4, Z_4 \to X_5, X_5 \to Y_5, Y_5 \to Z_5, Z_5 \to X_6];$$

which belong to the fraction ring of polynomial functions.

6.4. **Lattice W_5 algebra; main generator.** In this case we will use the following defined Poisson bracket based on Cartan matrix

$$A_4 = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

But for to do this according to our previous ordering and list of variables, and the same as what we din in sl_4 case, let us for simplicity set our set of variables as follows:
Set $X_i^{(1i)} := X_i$ and $X_i^{(2i)} := Y_i$ and $X_i^{(3i)} := Z_i$ and $X_i^{(4i)} := K_i$ and so on.

Definition 6.5. Let's define our Poisson bracket as follows in the case of sl_4:

$$\begin{cases} \{X_i, X_j\} := 2X_iX_j & \text{if } i < j; \\ \{Y_i, Y_j\} := 2Y_iY_j & \text{if } i < j; \\ \{Z_i, Z_j\} := 2Z_iZ_j & \text{if } i < j; \\ \{K_i, K_j\} := 2K_iK_j & \text{if } i < j; \\ \{X_i, X_i\} := 0; \\ \{Y_i, Y_i\} := 0; \\ \{Z_i, Z_i\} := 0; \\ \{K_i, K_i\} := 0; \\ \{X_i, Y_j\} := X_iY_j & \text{if } i > j; \\ \{X_i, Y_j\} := -X_iY_j & \text{if } i \leq j; \\ \{X_i, Z_j\} := 0; \\ \{X_i, K_j\} := 0; \\ \{Y_i, Z_j\} := Y_iZ_j & \text{if } i > j; \\ \{Y_i, Z_j\} := -Y_iZ_j & \text{if } i \leq j; \\ \{Y_i, K_j\} := Y_iK_j & \text{if } i > j; \\ \{Y_i, K_j\} := -Y_iK_j & \text{if } i \leq j; \end{cases} \tag{6.36}$$

As it comes out that, here our set of variables will be as follows:

$$\underset{-\infty}{\circ} - \cdots - \underset{X_1}{\circ} - \underset{Y_1}{\circ} - \underset{Z_1}{\circ} - \underset{K_1}{\circ} - \underset{X_2}{\circ} - \underset{Y_2}{\circ} - \underset{Z_2}{\circ} - \underset{K_2}{\circ} - \cdots - \underset{X_n}{\circ} - \underset{Y_n}{\circ} - \underset{Z_n}{\circ} - \underset{K_n}{\circ} - \cdots - \underset{+\infty}{\circ}$$

And instead of (2.1) we will have the following $q-$commutation relations for $j \in \{1,2,3\}$ and as always $i \in \{1,2,3\}$:

$$\begin{cases} x_i x_j = q^2 x_j x_i & \text{if } i \leq j \\ y_i y_j = q^2 y_j y_i & \text{if } i \leq j \\ z_i z_j = q^2 z_j z_i & \text{if } i \leq j \\ k_i k_j = q^2 k_j k_i & \text{if } i \leq j \\ x_i y_j = q^{-1} y_j x_i & \text{if } i \leq j \\ y_i z_j = q^{-1} z_j y_i & \text{if } i \leq j \\ z_i k_j = q^{-1} k_j z_i & \text{if } i \leq j \\ x_i z_j = z_j x_i \\ y_i k_j = k_j y_i \\ x_i k_j = k_j x_i \end{cases}$$

And by using the same approach as what we did for sl_2 and sl_3 and sl_4 , it became clear that the equations $D_X^{(5)}$, $D_Y^{(5)}$, $D_Z^{(5)}$ and $D_K^{(5)}$ and also $H_X^{(5)}$, $H_Y^{(5)}$, $H_Z^{(5)}$ and $H_K^{(5)}$ will have the following forms:

(6.37)

$$\mathfrak{D}_X^{(5)} = X_1(X_1+2X_2+2X_3)\frac{\partial \tau_1^{(5)}}{\partial X_1}+X_2(X_2+2X_3)\frac{\partial \tau_1^{(5)}}{\partial X_2}+X_3^2\frac{\partial \tau_1^{(5)}}{\partial X_3}-Y_1(X_2+X_3)$$
$$\frac{\partial \tau_1^{(5)}}{\partial Y_1} - Y_2X_3\frac{\partial \tau_1^{(5)}}{\partial Y_2};$$

(6.38)

$$\mathfrak{D}_Y^{(5)} = Y_1(Y_1+2Y_2+2Y_3)\frac{\partial \tau_1^{(5)}}{\partial Y_1}+Y_2(Y_2+2Y_3)\frac{\partial \tau_1^{(5)}}{\partial Y_2}+Y_3^2\frac{\partial \tau_1^{(5)}}{\partial Y_3}-X_1(Y_1+Y_2+Y_3)$$
$$\frac{\partial \tau_1^{(5)}}{\partial X_1} - X_2(Y_2+Y_3)\frac{\partial \tau_1^{(5)}}{\partial X_2} - X_3Y_3\frac{\partial \tau_1^{(5)}}{\partial X_3} - Z_1(Y_2+Y_3)\frac{\partial \tau_1^{(5)}}{\partial z_1} - Z_2y_3\frac{\partial \tau_1^{(5)}}{\partial Z_2};$$

(6.39)

$$\mathfrak{D}_Z^{(5)} = Z_1(Z_1+2Z_2+2Z_3)\frac{\partial \tau_1^{(5)}}{\partial Z_1}+Z_2(Z_2+2Z_3)\frac{\partial \tau_1^{(5)}}{\partial Z_2}+Z_3^2\frac{\partial \tau_1^{(5)}}{\partial Z_3}-Y_1(Z_1+Z_2+Z_3)$$
$$\frac{\partial \tau_1^{(5)}}{\partial Y_1} - Y_2(Z_2+Z_3)\frac{\partial \tau_1^{(5)}}{\partial Y_2} - Y_3Z_3\frac{\partial \tau_1^{(5)}}{\partial Y_3} - K_1(Z_2+Z_3)\frac{\partial \tau_1^{(5)}}{\partial k_1} - K_2Z_3\frac{\partial \tau_1^{(5)}}{\partial K_2};$$

(6.40)

$$\mathfrak{D}_K^{(5)} = K_1(K_1+2K_2+2K_3)\frac{\partial \tau_1^{(5)}}{\partial K_1}+K_2(K_2+2K_3)\frac{\partial \tau_1^{(5)}}{\partial K_2}+K_3^2\frac{\partial \tau_1^{(5)}}{\partial Z_3}-Z_1(K_1+K_2$$
$$+K_3)\frac{\partial \tau_1^{(5)}}{\partial Z_1} - Z_2(K_2+K_3)\frac{\partial \tau_1^{(5)}}{\partial z_2} - Z_3X_3\frac{\partial \tau_1^{(5)}}{\partial Z_3};$$

(6.41)

$$H_X^{(5)} = 2X_1\frac{\partial \tau_1^{(5)}}{\partial X_1} + 2X_2\frac{\partial \tau_1^{(5)}}{\partial X_2} + 2X_3\frac{\partial \tau_1^{(5)}}{\partial X_3} - Y_1\frac{\partial \tau_1^{(5)}}{\partial Y_1} - Y_2\frac{\partial \tau_1^{(5)}}{\partial Y_2} - Y_3\frac{\partial \tau_1^{(5)}}{\partial Y_3};$$

(6.42)
$$H_Y^{(5)} = 2Y_1\frac{\partial\tau_1^{(5)}}{\partial Y_1}+2Y_2\frac{\partial\tau_1^{(5)}}{\partial Y_2}+2Y_3\frac{\partial\tau_1^{(5)}}{\partial Y_3}-X_1\frac{\partial\tau_1^{(5)}}{\partial X_1}-X_2\frac{\partial\tau_1^{(5)}}{\partial X_2}-X_3\frac{\partial\tau_1^{(5)}}{\partial X_3}-Z_1\frac{\partial\tau_1^{(5)}}{\partial Z_1}$$
$$-Z_2\frac{\partial\tau_1^{(5)}}{\partial Z_2} - Z_3\frac{\partial\tau_1^{(5)}}{\partial Z_3};$$

(6.43)
$$H_Z^{(5)} = 2Z_1\frac{\partial\tau_1^{(5)}}{\partial Z_1}+2Z_2\frac{\partial\tau_1^{(5)}}{\partial Z_2}+2Z_3\frac{\partial\tau_1^{(5)}}{\partial z_3}-Y_1\frac{\partial\tau_1^{(5)}}{\partial Y_1}-y_2\frac{\partial\tau_1^{(5)}}{\partial Y_2}-Y_3\frac{\partial\tau_1^{(5)}}{\partial Y_3}-K_1\frac{\partial\tau_1^{(5)}}{\partial K_1}$$
$$-K_2\frac{\partial\tau_1^{(5)}}{\partial K_2} - K_3\frac{\partial\tau_1^{(5)}}{\partial K_3};$$

(6.44)
$$H_K^{(5)} = 2K_1\frac{\partial\tau_1^{(5)}}{\partial K_1} + 2K_2\frac{\partial\tau_1^{(5)}}{\partial K_2} + 2K_3\frac{\partial\tau_1^{(5)}}{\partial K_3} - Z_1\frac{\partial\tau_1^{(5)}}{\partial Z_1} - Z_2\frac{\partial\tau_1^{(5)}}{\partial Z_2} - Z_3\frac{\partial\tau_1^{(5)}}{\partial Z_3};$$

And the functional dependent nontrivial solutions for the whole system of first order partial differential equation is as follows:

$$\text{(6.45)}\quad \tau_1^{(5)} = \frac{(\Sigma_{1\le i\le j\le m\le l\le 2}\ x_i y_j z_m k_l)(\Sigma_{1\le i\le j\le m\le l\le 2}\ x_{i+1}y_{j+1}z_{m+1}k_{l+1})}{x_2y_2z_2k_2(\Sigma_{1\le i\le j\le m\le l\le 3}\ x_i y_j z_m k_l)};$$

And again as before, (5) goes back to 5 in the Sl_5 and 1 is a default index which later we will use it for to employ our shifting operators.
According to the number of variables, we will have 12 shifts and then after that it will be in a loop.
So here in sl_5 case we have twelve solutions just as what we did in sl_4, and here skip to write them down.

6.5. **Lattice W_n algebra; main generator.** Here for sl_n, we skip to write down all steps which we have done in previous sections and just will write down our main generator of the lattice W_n algebra.
The functional dependent nontrivial solution for the whole system of first order partial differential equations will be as what comes in follow:

$$\text{(6.46)}\qquad \tau_1^{(n)} =$$
$$\frac{(\Sigma_{1\le i_1\le i_2\cdots\le i_{n-1}\le 2}\ x_{i_1}^{(1)}x_{i_2}^{(2)}\cdots x_{i_{n-1}}^{(n-1)})(\Sigma_{1\le i_1\le i_2\cdots\le i_{n-1}\le 2}\ x_{i_1+1}^{(1)}x_{i_2+1}^{(2)}\cdots x_{i_{n-1}+1}^{(n-1)})}{x_2^{(1)}\cdots x_2^{(n-1)}(\Sigma_{1\le i_1\le i_2\cdots\le i_{n-1}\le 3}\ x_{i_1}^{(1)}x_{i_2}^{(2)}\cdots x_{i_{n-1}}^{(n-1)})};$$

We should notice that $x_{i_j}^{(j)}$s are different of each other for any $j \in \{1,\cdots n-1\}$.

7. Mathematica techniks in solving the system of partial differential equations

This section has been completed by getting help from professor Brendan B. Godfrey from Institute for Research in Electronics and Applied Physics (The University of Maryland), in a direct communications and discutions through email and also through a series of questions and discutions in mathematica stackexchange.
And I have to say that without his great Mathematica skills, it nearly was impossible to get such an interesting results!
In this appendix you will be able to see some parts of Mathematica coddings which we have used for to obtain our algebra structures.
And we believe that what is written in this appendix can open a new approach in solving the following system of $q-$linear homogeneous equations in one unknown f.

$$\begin{cases} equ_1(f) = a_{11}\frac{\partial f}{\partial x_1} + a_{21}\frac{\partial f}{\partial x_2} + \cdots + a_{n1}\frac{\partial f}{\partial x_n} = 0 \\ equ_2(f) = a_{12}\frac{\partial f}{\partial x_1} + a_{22}\frac{\partial f}{\partial x_2} + \cdots + a_{n2}\frac{\partial f}{\partial x_n} = 0 \\ \quad\vdots \qquad\qquad \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \\ equ_q(f) = a_{1q}\frac{\partial f}{\partial x_1} + a_{2q}\frac{\partial f}{\partial x_2} + \cdots + a_{nq}\frac{\partial f}{\partial x_n} = 0 \end{cases} \tag{7.1}$$

Where the coefficients a_{ik} are functions of n independent variables $x_1, \cdots, x_n$ and do not contain the unknown function f. [5]
And we have to mention that, to reach to this point was impossible without using Mathematica!

7.1. Lattice W_3 algebra. .

LISTING 1. Example code

```
p = D[f[x1, x2, x3, y1, y2, y3], x1];
q = D[f[x1, x2, x3, y1, y2, y3], x2];
r = D[f[x1, x2, x3, y1, y2, y3], x3];
o = D[f[x1, x2, x3, y1, y2, y3], y1];
x = D[f[x1, x2, x3, y1, y2, y3], y2];
a = D[f[x1, x2, x3, y1, y2, y3], y3];
equ1 = 2 x1 p + 2 x2 q + 2 x3 r − y1 o − y2 x − y3 a;
equ2 = −x1 p − x2 q − x3 r + 2 y1 o + 2 y2 x + 2 y3 a;
equ3 = (x1 (x1 + 2 x2 + 2 x3)) p + (x2 (x2 + 2 x3)) q + (x3^2) r
− (y1 (x2 + x3)) o − y2 x3 x;
equ4 = (y1 (y1 + 2 y2 + 2 y3)) o + (y2 (y2 + 2 y3)) x + (y3^2) a
− (x1 (y1 + y2 + y3)) p − x2 (y2 + y3) q − x3 y3 r;
DSolve[{equ1 == 0, equ2 == 0, equ3 == 0, equ4 == 0}, f, {x1, x2, x3, y1, y2, y3}]
```

As you see *DSolve* returns un-evaluated i.e. it means that it is not able to solve our system of first order partial differential equations.

LISTING 2. Example code

```
DSolve[2 equ1 + equ2 == 0, f[x1, x2, x3, y1, y2, y3], {x1, x2, x3, y1, y2, y3}][[1, 1]]
(* f[x1, x2, x3, y1, y2, y3] −> C[1][x2/x1, x3/x1, y1, y2, y3] *)
DSolve[equ1 + 2 equ2 == 0, f[x1, x2, x3, y1, y2, y3], {x1, x2, x3, y1, y2, y3}][[1, 1]]
(* f[x1, x2, x3, y1, y2, y3] −> C[1][x1, x2, x3, y2/y1, y3/y1] *)
```

Consequently, the dimensionality of this problem can be reduced from six to four.

LISTING 3. Example code

```
f[x1_, x2_, x3_, y1_, y2_, y3_] := g[x2/x1, x3/x1, y2/y1, y3/y1]
equ5 = FullSimplify[(equ3/x1) /. {x2 -> v2 x1, x3 -> v3 x1, y2 -> w2 y1, y3 -> w3 y1
    }]
(*          (v2 + v3)*w3*Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] +
v2*w2*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] -
v3*(1 + 2*v2 + v3)*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] -
v2*(1 + v2)*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)
equ6 = FullSimplify[(equ4/y1) /. {x2 -> v2 x1, x3 -> v3 x1, y2 -> w2 y1, y3 -> w3 y1
    }]
(* -(w3*(1 + 2*w2 + w3)*Derivative[0, 0, 0, 1][g][v2, v3, w2, w3]) -
w2*(1 + w2)*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] +
v3*(1 + w2)*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] +
v2*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)
```

Although *DSolve* cannot solve these equations as a pair either. But it can solve each separately.

LISTING 4. Example code

```
DSolve[equ5 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]]/. C[1] -> c5
(* g[v2, v3, w2, w3] -> c5[(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2] *)
(DSolve[equ6 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]] /.
C[1] -> c6) // FullSimplify
(* g[v2, v3, w2, w3] -> c6[-((v3 (1 + w2))/v2), ((1 + w2) (1 + w2 + w3))/(v2 w3),
-Log[(1 + w2)/(v2 w2)]] *)
```

The first results indicates that g is a function of

LISTING 5. Example code

```
var5 = List @@ %%[[2]]
(* {(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2} *)
```

and also

LISTING 6. Example code

```
var6 = List @@ %%[[2]]
(* {-((v3 (1 + w2))/v2), ((1 + w2) (1 + w2 + w3))/(v2 w3), -Log[(1 + w2)/(v2 w2)]} *)
```

The second list of functions can be simplified by

LISTING 7. Example code

```
var6 [[3]]  = Exp[var6 [[3]]];
var6 [[1]]  = -var6[[1]] var6 [[3]];
var6 [[2]]  = var6 [[2]]  var6 [[3]];
var6
(* {v3 w2, (w2 (1 + w2 + w3))/w3, (v2 w2)/(1 + w2)} *)
```

Now the next step is to combine the previous two expressions for g for to obtain a single expression, presumably as a function of two variables.

The system of PDEs above can be solved using the procedure described in Chapter V, Sec IV of Goursat's Differential Equations [3].
The first step is to find the complete, non-commutative group of differential operators that includes $equ5$ and $equ6$.

LISTING 8. Example code

```
comm[equa_, equb_] :=
Collect[(equa /. {Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] −> D[equb, v2],
Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] −> D[equb, v3],
Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] −> D[equb, w2],
Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] −> D[equb, w3]}) −
(equb /. {Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] −> D[equa, v2],
Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] −> D[equa, v3],
Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] −> D[equa, w2],
Derivative[0, 0, 0, 1][g][v2, v3, w2, w3] −> D[equa, w3]}),
{Derivative[1, 0, 0, 0][g][v2, v3, w2, w3], Derivative[0, 1, 0, 0][g][v2, v3, w2, w3],
Derivative[0, 0, 1, 0][g][v2, v3, w2, w3], Derivative[0, 0, 0, 1][g][v2, v3, w2, w3]},
Simplify]
equ7 = comm[equ5, equ6]
(* −(w3*(v3*(1 + w2 + w3) + v2*(1 + 2*w2 + w3))*Derivative[0, 0, 0, 1][g][v2, v3, w2, w3])
− v2*w2*(1 + w2)*Derivative[0, 0, 1, 0][g][v2, v3, w2, w3] + v3*(v3*(1 + w2)
+ v2*(2 + w2))*Derivative[0, 1, 0, 0][g][v2, v3, w2, w3] +
v2^2*Derivative[1, 0, 0, 0][g][v2, v3, w2, w3] *)
```

which by inspection is independent of $equ5$ and $equ6$. On the other hand, $comm[equ5, equ7]$ and $comm[equ6, equ7]$ do not yield independent equations, again by inspection. Thus $\{equ5, equ6, equ7\}$ is a complete group of three operators in four independent variables. From this information alone, we know that g is an arbitrary function of precisely one first integral. This first integral can be obtained by systematically eliminating variables and equations, one pair at a time, until a single equation of two variable remains. We start by solving any one of the equations.

LISTING 9. Example code

```
DSolve[equ5 == 0, g[v2, v3, w2, w3], {v2, v3, w2, w3}][[1, 1]]
(* g[v2, v3, w2, w3] −> C[1][(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2] *)
```

and use the solution as the basis for a change of variables:

LISTING 10. Example code

```
g[v2_, v3_, w2_, w3_] := h[w2, (v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2]
solw2 = equ5/(v2 w2) // Simplify
(* Derivative[1, 0, 0, 0][h][w2, (v2*(1 + v2 + v3))/v3, (1 + v2)*w2, (v3*w3)/v2] *)
```

indicating that h is independent of $w2$. This leaves us with two equations in three variables

LISTING 11. Example code

```
newvar = Solve[Thread[{b1, b2, b3, b4} == List @@ solw2], {v2, v3, w2, w3}] // Flatten;
(((b3 equ7/(b1 − b3)) // FullSimplify) /. solw2 −> 0 /. newvar) // FullSimplify;\\
Collect[((equ6 // FullSimplify) /. solw2 −> 0 /. newvar) //
FullSimplify, b1, FullSimplify] + % b1/b3
equ10 = %% /. Derivative[0, n1_, n2_, n3_][h][b1, b2, b3, b4] −>
Derivative[n1, n2, n3][h][b2, b3, b4]
(* b4*(b3 + b4 + b2*b4)*Derivative[0, 0, 1][h][b2, b3, b4] + (1 + b3)*
(b3*Derivative[0, 1, 0][h][b2, b3, b4] + (1 + b2)*Derivative[1, 0, 0][h][b2, b3, b4]) *)
equ11 = %% /. Derivative[0, n1_, n2_, n3_][h][b1, b2, b3, b4] −>
Derivative[n1, n2, n3][h][b2, b3, b4]
```

```
(* (−1 + b4)*b4*Derivative[0, 0, 1][h][b2, b3, b4] + (1 + b2 + b3)
*Derivative[1, 0, 0][h][b2, b3, b4] *)
```

Proceeding as before, we next solve one of $equ10$ and $equ11$. (We choose the simpler one.)

LISTING 12. Example code

```
DSolve[equ11 == 0, h[b2, b3, b4], {b2, b3, b4 }][[1, 1, 2]]
(* h[b2, b3, b4] −> C[1][b3][Log[(1 − b4)/((1 + b2 + b3) b4)] *)
```

and use it as the basis for a further change of variables.

LISTING 13. Example code

```
h[b2_, b3_, b4_] := k[b2, b3, (1 − b4)/((1 + b2 + b3) b4)]
solb2 = (equ11/(1 + b2 + b3)) // Simplify
(* Derivative[1, 0, 0][k][b2, b3, (1 − b4)/(b4 + b2*b4 + b3*b4)] *)
```

indicating that k is independent of $b2$. This leaves us with one equation in two variables.

LISTING 14. Example code

```
newvar1 = Solve[Thread[{c2, c3, c4} == List @@ solb2], {b2, b3, b4}] // Flatten;
((equ10 // FullSimplify) /. solb2 −> 0 /. newvar1) // FullSimplify
equ12 = % /. Derivative[0, n1_, n2_][k][c2, c3, c4] −> Derivative[n1, n2][k][c3, c4]
(* −((1 + c4 + 2*c3*c4)*Derivative[0, 1][k][c3, c4]) + c3*(1 + c3)
*Derivative[1, 0][k][c3, c4] *)
```

Finally, $DSolve$ yields

LISTING 15. Example code

```
DSolve[equ12 == 0, k[c3, c4], {c3, c4 }][[1, 1, 2]]
(* k[c3, c4] −> C[1][c3 (1 + c4 + c3 c4)] *)
```

Transforming back to the original independent variables gives

LISTING 16. Example code

```
(((% /. Thread[{c2, c3, c4} −> List @@ solb2]) // Simplify) /.
Thread[{b1, b2, b3, b4} −> List @@ solw2]) // Simplify
(* C[1][(v2 w2 (1 + (1 + v2) w2 + (1 + v2 + v3) w3))/((v2 + v3 + v3 w2) w3)] *)
```

LISTING 17. Example code

```
g[v2_, v3_, w2_, w3_] := %
{equ5, equ6, equ7} // Simplify
(* {0, 0, 0} *)
```

Finally, designating the solution for g as $ansg$,

LISTING 18. Example code

```
(ansg /. {v2 −> x2/x1, v3 −> x3/x1, w2 −> y2/y1, w3 −> y3/y1}) // Simplify
(* C[1][(x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)))/(x1 (x2 y1 + x3 (y1 + y2)) y3
)] *)
f[x1_, x2_, x3_, y1_, y2_, y3_] := %
```

```
{equ1, equ2, equ3, equ4}
(* {0, 0, 0, 0} *)
```

7.2. **Lattice W_4 algebra.** .

LISTING 19. Example code

```
p = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x1];

q = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x2];

r = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], x3];

o = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y1];

x = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y2];

a = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], y3];

b = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z1];

c = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z2];

d = D[f[x1, x2, x3, y1, y2, y3, z1, z2, z3], z3];

equ1 = 2 x1 p + 2 x2 q + 2 x3 r − y1 o − y2 x − y3 a;

equ2 = −x1 p − x2 q − x3 r − z1 b − z2 c − z3 d + 2 y1 o + 2 y2 x + 2 y3 a;

equ3 = 2 z1 b + 2 z2 c + 2 z3 d − y1 o − y2 x − y3 a;

equ4 = (x1 (x1 + 2 x2 + 2 x3)) p + (x2 (x2 +
2 x3)) q + (x3^2) r − (y1 (x2 + x3)) o − y2 x3 x;

equ5 = (y1 (y1 + 2 y2 + 2 y3)) o + (y2 (y2 + 2 y3)) x + (y3^2) a − (x1 (y1 + y2 + y3)) p
− x2 (y2 + y3) q − x3 y3 r − z1 (y2 + y3) b − z2 y3 c;

equ6 = (z1 (z1 + 2 z2 + 2 z3)) b + (z2 (z2 + 2 z3)) c + (z3^2) d − (y1 (z1 + z2 + z3)) o
− y2 ( z2 + z3) x − y3 z3 a;
```

LISTING 20. Example code

```
DSolve[{HX == 0, HY == 0, , HY == 0, EX == 0, EY == 0, EZ == 0},
f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
z2, z3}]
```

Again $DSolve$ returns un-evaluated, meaning that it can not solve the system of equations.

LISTING 21. Example code

```
DSolve[ HX + 3 HZ + 2 HY == 0,
f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
z2, z3 }][[1, 1]]
(*f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
```

```
C[1][x1, x2, x3, y1, y2, y3, z2/z1, z3/z1]*)
```

LISTING 22. Example code

```
DSolve[ HX + HZ + 2 HY == 0,
f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
z2, z3 }][[1, 1]]
(* f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
C[1][x1, x2, x3, y2/y1, y3/y1, z1, z2, z3] *)
```

LISTING 23. Example code

```
DSolve[ 3 HX + HZ + 2 HY == 0,
f[x1, x2, x3, y1, y2, y3, z1, z2, z3], {x1, x2, x3, y1, y2, y3, z1,
z2, z3 }][[1, 1]]
(*f[x1, x2, x3, y1, y2, y3, z1, z2, z3] ->
C[1][x2/x1, x3/x1, y1, y2, y3, z1, z2, z3]*)
```

As before, this computation can be simplified by the substitution,

LISTING 24. Example code

```
f[x1, x2, x3, y1, y2, y3, z1, z2, z3] := g[x2/x1, x3/x1, y2/y1, y3/y1, z2/z1, z3/z1]
```

in which case the six equations become

LISTING 25. Example code

```
Simplify[{equ1, equ2, equ3}]
(* {0, 0, 0} *)

equ4 = Simplify[Simplify[equ4]/x1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
(* (v2 + v3)*w3*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
v2*w2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
v3*(1 + 2*v2 + v3)*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
v2*(1 + v2)*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)

equ5 = Simplify[Simplify[equ5]/y1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
(* k3*(w2 + w3)*Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] +
k2*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -
w3*(1 + 2*w2 + w3)*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -
w2*(1 + w2)*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
v3*(1 + w2)*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
v2*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)

equ6 = Simplify[Simplify[ equ6]/z1 /. {x2 -> x1 v2, x3 -> x1 v3, y2 -> y1 w2,
y3 -> y1 w3, z2 -> z1 k2, z3 -> z1 k3}]
(* -(k3*(1 + 2*k2 + k3)*Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]) -
k2*(1 + k2)*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] +
(1 + k2)*w3*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
w2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
```

As before, this system of first-order $PDEs$ can be solved by using the procedure described in Chapter V, Sec IV of Goursat's Differential Equations. The first step is to find the complete, non-commutative group of differential operators that includes $equ4$, $equ5$, and $equ6$. To do so, we use the function $comm$, generalized from W_3

LISTING 26. Example code

```
drv = {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3],
Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3],
Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3],
Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]};
comm[equa_, equb_] := Collect[
(equa /. {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, v2],
Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, v3],
Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, w2],
Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, w3],
Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -> D[equb, k2],
Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] -> D[equb, k3]}) -
(equb /. {Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, v2],
Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, v3],
Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, w2],
Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, w3],
Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -> D[equa, k2],
Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3] -> D[equa, k3]}),
drv, Simplify]

equ7 = comm[equ4, equ5]
(* (k3*v2*w2 + k3*(v2 + v3)*w3)*Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]
    +
k2*v2*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] -
w3*(v3*(1 + w2 + w3) + v2*(1 + 2*w2 + w3))*
Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] -
v2*w2*(1 + w2)*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
v3*(v3*(1 + w2) + v2*(2 + w2))*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] +
v2^2*Derivative[1, 0, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)

equ8 = comm[equ5, equ6]
(* -(k3*((1 + 2*k2 + k3)*w2 + (1 + k2 + k3)*w3)*
Derivative[0, 0, 0, 0, 0, 1][g][v2, v3, w2, w3, k2, k3]) -
k2*(1 + k2)*w2*Derivative[0, 0, 0, 0, 1, 0][g][v2, v3, w2, w3, k2, k3] +
w3*((2 + k2)*w2 + (1 + k2)*w3)*Derivative[0, 0, 0, 1, 0, 0][g][v2, v3, w2, w3, k2, k3] +
w2^2*Derivative[0, 0, 1, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] -
v3*w2*Derivative[0, 1, 0, 0, 0, 0][g][v2, v3, w2, w3, k2, k3] *)
```

which are independent of the first three operators, increasing the size of the group to five. $comm[equ4, equ6]$ vanishes identically and so does not add an operator. On the other hand, the seven additional commutators involving $equ7$ and $equ8$ yield expressions that are linear combinations of $\{equ4, equ5, equ6, equ7, equ8\}$. Thus, these five operators comprise the entire group.

From this information alone, we know that g is an arbitrary function of precisely one first integral. This first integral can be obtained by systematically eliminating variables and equations, one pair at a time, until a single equation of two variable remains. Start by solving any one of the equations.

LISTING 27. Example code

```
DSolve[equ4 == 0,
g[v2, v3, w2, w3, k2, k3], {v2, v3, w2, w3, k2, k3 }][[1,
 1]] // FullSimplify
(* g[v2, v3, w2, w3, k2, k3] ->
```

```
C[1][k2, k3][(v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2] *)
```

LISTING 28. Example code

```
g[v2_, v3_, w2_, w3_, k2_, k3_] :=
h[w2, (v2 (1 + v2 + v3))/v3, (1 + v2) w2, (v3 w3)/v2, k2, k3];
tr1 = {equ4, equ5, equ6, equ7, equ8} // Simplify;
```

LISTING 29. Example code

```
solw2 = equ4/(v2 w2) // FullSimplify;
```

LISTING 30. Example code

```
newvar = Solve[
Thread[{b1, b2, b3, b4, b5, b6} == List @@ solw2], {v2, v3, w2,
w3, k2, k3}] // Flatten;
tr1p = Collect[FullSimplify[Rest[tr1] /. solw2 -> 0 /. newvar], b1,
FullSimplify] /. b1*(z_) -> 0 /.
Derivative[0, n1_, n2_, n3_, n4_, n5_][h][b1, b2, b3, b4, b5,
b6] -> Derivative[n1, n2, n3, n4, n5][h][b2, b3, b4, b5, b6];
```

LISTING 31. Example code

```
DSolve[First@tr1p == 0,
h[b2, b3, b4, b5, b6], {b2, b3, b4, b5, b6}] /. Log[z_] -> z;
h[b2_, b3_, b4_, b5_, b6_] :=
j[b4, b3, b5, (1 + b2 + b3) b4 b6, b6 (1 - b4)];
tr2 = tr1p // Simplify;
solb4 = First@tr2/(b4 (b4 - 1)) // Simplify;
newvar = Solve[
Thread[{c1, c2, c3, c4, c5} == List @@ solb4], {b2, b3, b4, b5,
b6}] // Flatten;

tr2p = Collect[(Cancel[(c1 - 1) Rest@tr2] /. solb4 -> 0 /. newvar) //
FullSimplify, c1, FullSimplify];
tr2p [[3]] = tr2p [[3]]/ c1;
tr2p = tr2p /. c1 z__ -> 0 /.
Derivative[0, n1_, n2_, n3_, n4_][j][c1, c2, c3, c4, c5] ->
Derivative[n1, n2, n3, n4][j][c2, c3, c4, c5];
```

LISTING 32. Example code

```
DSolve[Last@tr2p == 0, j[c2, c3, c4, c5], {c2, c3, c4, c5}];
j[c2_, c3_, c4_, c5_] := l[c5, c2, c3, (c2 - c4)/(1 + c3 + c5)];
tr3 = -tr2p // Simplify // RotateRight;
solc5 = First@tr3/(c5 (1 + c3 + c5)) // Simplify;
newvar = Solve[
Thread[{d1, d2, d3, d4} == List @@ solc5], {c2, c3, c4, c5}] //
Flatten;
tr3p = Collect[(Rest@tr3 /. solc5 -> 0 /. newvar) // FullSimplify, d1,
FullSimplify] /. d1 z_ -> 0 /.
Derivative[0, n1_, n2_, n3_][l][d1, d2, d3, d4] ->
Derivative[n1, n2, n3][l][d2, d3, d4];
```

LISTING 33. Example code

```
DSolve[Last@tr3p == 0, l[d2, d3, d4], {d2, d3, d4}] // Simplify;
l[d2_, d3_, d4_] := m[d3, (1 + d2) d3, (d2 (1 + d2 - d4))/d4];
tr4 = tr3p // Simplify // RotateRight;
```

```
sold3 = First@tr4/(d2 d3);
newvar = Solve[Thread[{e1, e2, e3} == List @@ sold3], {d2, d3, d4}] //
Flatten;
tr4p = Collect[(−(e2/e1) Rest@tr4 /. sold3 −> 0 /. newvar) //
FullSimplify, e1, FullSimplify] /. e1 z_ −> 0 /.
Derivative[0, n1_, n2_][m][e1, e2, e3] −>
Derivative[n1, n2][m][e2, e3];
```

LISTING 34. Example code

```
(DSolve[Last@tr4p == 0, m[e2, e3], {e2, e3}] // Simplify) /.
Log[z_] −> z;
 ((((((((%[[1, 1, 2]] /. Thread[{e1, e2, e3} −> List @@ sold3]) //
Simplify) /.
Thread[{d1, d2, d3, d4} −> List @@ solc5]) //
Simplify) /.
Thread[{c1, c2, c3, c4, c5} −> List @@ solb4]) //
Simplify) /.
Thread[{b1, b2, b3, b4, b5, b6} −> List @@ solw2] //
Simplify) /. {v2 −> x2/x1, v3 −> x3/x1, w2 −> y2/y1, w3 −> y3/y1,
k2 −> z2/z1, k3 −> z3/z1}) // Simplify
```

Final solution

LISTING 35. Example code

```
C[1][−(((x2 y2 z2 + x1 (y2 z2 + y1 (z1 + z2))) (x3 y3 z3 +
x2 (y3 z3 + y2 (z2 + z3))))/(
x2 y2 z2 (x3 y3 z3 + x2 (y3 z3 + y2 (z2 + z3)) +
x1 (y3 z3 + y2 (z2 + z3) + y1 (z1 + z2 + z3)))))]
```

7.3. **Expressing a fractional multivariate polynomials to its low-order polynomial decomposition.** Suppose we have given the following question.

Question:
Let $f2$ be fractional multivariate polynomial as follows

LISTING 36. Example code

```
f2 = −((2 x1 x2 x3 x4 y1 y2^2 y3 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3) (x3 y3 + x2 (y2 +
    y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2))^2 (x2 y1 + x3 (y1 + y2)) (x3 y3
    + x2 (y2 + y3)) (x3 y2 + x4 (y2 + y3))^2));
```

and also let $k1$ and $k2$ be given as follows

LISTING 37. Example code

```
k1 = (x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2)) (x3 y3 +
    x2 (y2 + y3)));
k2 = (x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3))/((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (
    y2 + y3)));
```

then express $f2$ as a low - order polynomial in $k1$ and $k2$.

This can be done as follows.
First, generate a generic low order polynomial.

LISTING 38. Example code

```
Map[t1^First@# t2^Last@# &, Tuples[Range[0, 3], 2]].Table[Unique["c"], {16}]
(* c3 + c7 t1 + c11 t1^2 + c15 t1^3 + c4 t2 + c8 t1 t2 + c12 t1^2 t2 +
```

```
c16 t1^3 t2 + c5 t2^2 + c9 t1 t2^2 + c13 t1^2 t2^2 + c17 t1^3 t2^2 +
c6 t2^3 + c10 t1 t2^3 + c14 t1^2 t2^3 + c18 t1^3 t2^3 *)
```

and then use SolveAlways. After about twenty seconds we will gwt result

LISTING 39. Example code

```
Flatten@SolveAlways[f2 == (% /. {t1 -> k1, t2 -> k2}), {x1, x2, x3, x4, y1, y2, y3}]
(* {c3 -> 0, c4 -> 0, c5 -> 0, c6 -> 0, c11 -> 0, c15 -> 0, c7 -> 0, c12 -> 2, c16
    -> 0, c8 -> -2, c10 -> 0, c13 -> -2, c14 -> 0, c17 -> 0, c18 -> 0, c9 ->
   2} *)
```

And we have the final solution

LISTING 40. Example code

```
Factor[%% /. %]
(* -2 (-1 + t1) t1 (-1 + t2) t2 *)
```

which is the desired result.
And for completeness we have

LISTING 41. Example code

```
Simplify[f2 == % /. {t1 -> k1, t2 -> k2}]
(* True *)
```

Also here we have much faster alternative:
Because SolveAlways determines the coefficients c for any $\{x1, x2, x3, x4, y1, y2, y3\}$, Solve must be able to obtain the same values for the coefficients c for specific values of $\{x1, x2, x3, x4, y1, y2, y3\}$, and much faster. As before we do have $f2$ and $k1$ and $k2$.

LISTING 42. Example code

```
f2 = -((2 x1 x2 x3 x4 y1 y2^2 y3 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3) (x3 y3 + x2 (y2 +
    y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2))^2 (x2 y1 + x3 (y1 + y2)) (x3 y3
   + x2 (y2 + y3)) (x3 y2 + x4 (y2 + y3))^2));
```

LISTING 43. Example code

```
k1 = (x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)))/((x2 y2 + x1 (y1 + y2)) (x3 y3 +
    x2 (y2 + y3)));
k2 = (x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3))/((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (
   y2 + y3)));
```

LISTING 44. Example code

```
tp = Tuples[Range[0, 3], 2]; tp // Length
(* 16 *)
```

LISTING 45. Example code

```
gp = Map[t1^#[[1]] t2^#[[2]] &, tp].Table[Unique["c"], {tp // Length}]
(* c3 + c7 t1 + c11 t1^2 + c15 t1^3 + c4 t2 + c8 t1 t2 + c12 t1^2 t2 + c16 t1^3 t2 + c5
    t2^2 + c9 t1 t2^2 + c13 t1^2 t2^2 + c17 t1^3 t2^2 + c6 t2^3 + c10 t1 t2^3 + c14 t1
   ^2 t2^3 + c18 t1^3 t2^3 *)
```

LISTING 46. Example code

```
Flatten@Solve[Table[(f2 == (gp /. {t1 -> k1, t2 -> k2})) /. Thread[{x1, x2, x3, x4, y1, y2, y3} -> RandomInteger[{1, 7}, 7]], {n, tp // Length}], List @@ (First@# & /@ (gp /. gp[[1]] -> gp[[1]] z))]
(* {c10 -> 0, c11 -> 0, c12 -> 2, c13 -> -2, c14 -> 0, c15 -> 0, c16 -> 0, c17 -> 0, c18 -> 0, c3 -> 0, c4 -> 0, c5 -> 0, c6 -> 0, c7 -> 0, c8 -> -2, c9 -> 2} *)
```

LISTING 47. Example code

```
Factor[%% /. %]
(* -2 (-1 + t1) t1 (-1 + t2) t2 *)
```

LISTING 48. Example code

```
Simplify[f2 == % /. {t1 -> k1, t2 -> k2}]
(* True *)
```

Question:
Let $f6$ be fractional multivariate polynomial as follows

LISTING 49. Example code

```
f6 = (2 x1 x2 x5 x6 y2 (x2 y1 + x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5))) /((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4 + y5)));
```

and also let $k1$, $k2$, $k3$, $k4$, $k5$ and $k6$ be given as follows

LISTING 50. Example code

```
k1 = ((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))) /(x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)));
k2 = ((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (y2 + y3))) /(x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3));
k3 = ((x3 y3 + x2 (y2 + y3)) (x4 y4 + x3 (y3 + y4))) /(x3 y3 (x4 y4 + x3 (y3 + y4) + x2 (y2 + y3 + y4)));
k4 = ((x3 y2 + x4 (y2 + y3)) (x4 y3 + x5 (y3 + y4))) /(x4 y3 (x3 y2 + (x4 + x5) (y2 + y3) + x5 y4));
k5 = ((x4 y4 + x3 (y3 + y4)) (x5 y5 + x4 (y4 + y5))) /(x4 y4 (x5 y5 + x4 (y4 + y5) + x3 (y3 + y4 + y5)));
k6 = ((x4 y3 + x5 (y3 + y4)) (x5 y4 + x6 (y4 + y5))) /(x5 y4 (x4 y3 + (x5 + x6) (y3 + y4) + x6 y5));
```

then express $f6$ as a low - order polynomial in $k1$, $k2$, $k3$, $k4$, $k5$ and $k6$.

LISTING 51. Example code

```
tp = Tuples[Range[-1, 1], 6]; tp // Length
(* 729 *)
```

LISTING 52. Example code

```
gp = Map[t1 ^#[[1]] t2 ^#[[2]] t3 ^#[[3]] t4 ^#[[4]] t5 ^#[[5]] t6 ^#[[6]] &, tp ]. Table[Unique["c"], {tp // Length}];
```

LISTING 53. Example code

```
sol = Flatten@ Solve[Table[(f6 == (gp /. {t1 −> k1, t2 −> k2, t3 −> k3, t4 −> k4, t5
    −> k5, t6 −> k6})) /. Thread[{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5} −>
    RandomInteger[{1, 11}, 11]], {n, tp // Length}], List @@ (First@# & /@ (gp /. gp
    [[1]] −> gp[[1]] z)) ]; sol /. Rule[_, 0] −> Nothing
(* {Nothing, ..., Nothing, c114 −> −2, c115 −> 2, Nothing,..., Nothing, c123 −> 2,
    c124 −> −2, Nothing, ..., Nothing, c330 −> −2, c331 −> 2, Nothing, ... , Nothing,
    c339 −> 2, Nothing, c340 −> −2, Nothing, ..., Nothing, c357 −> 2, c358 −> −2,
    Nothing, ..., Nothing, c366 −> −2, c367 −> 2, Nothing, ..., Nothing, c87 −> 2, c88
    −> −2, Nothing, ..., Nothing, c96 −> −2, c97 −> 2, Nothing, Nothing} *)
```

LISTING 54. Example code

```
Factor[gp /. sol]
(* (2 (−1 + t1) (−1 + t3) (−1 + t4) (−1 + t6))/(t1 t3 t4 t6) *)
```

Which is the desired result.
And as before for completeness we have

LISTING 55. Example code

```
Simplify[f6 == % /. {t1 −> k1, t2 −> k2, t3 −> k3, t4 −> k4, t5 −> k5, t6 −> k6}]
(* True *)
```

By using Groebner Basis:
Also there is another way for to reach to the solution by using Groebner Basis. But this approach is very slow!

LISTING 56. Example code

```
poly = (2 x1 x2 x5 x6 y2 (x2 y1 + x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5))) /((x2 y2
    + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4
    + y5)));
```

LISTING 57. Example code

```
eqns = {K1 == ((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (y2 + y3))) /(x2 y2 (x3 y3 + x2 (y2 +
    y3) + x1 (y1 + y2 + y3))),
K2 == ((x2 y1 + x3 (y1 + y2)) (x3 y2 + x4 (y2 + y3))) /(x3 y2 (x2 y1 + (x3 + x4) (y1 +
    y2) + x4 y3)),
K3 == ((x3 y3 + x2 (y2 + y3)) (x4 y4 + x3 (y3 + y4))) /(x3 y3 (x4 y4 + x3 (y3 + y4) + x2
     (y2 + y3 + y4))),
K4 == ((x3 y2 + x4 (y2 + y3)) (x4 y3 + x5 (y3 + y4))) /(x4 y3 (x3 y2 + (x4 + x5) (y2 +
    y3) + x5 y4)),
K5 == ((x4 y4 + x3 (y3 + y4)) (x5 y5 + x4 (y4 + y5))) /(x4 y4 (x5 y5 + x4 (y4 + y5) +
    x3 (y3 + y4 + y5))),
K6 == ((x4 y3 + x5 (y3 + y4)) (x5 y4 + x6 (y4 + y5))) /(x5 y4 (x4 y3 + (x5 + x6) (y3 +
    y4) + x6 y5))};
```

Now let us compute Groebner Basis

LISTING 58. Example code

```
gb = GroebnerBasis[eqns, {x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6}];
```

The remainder r gives a representation of poly in terms of $K1$, $K2$, $K3$, $K4$, $K5$ and $K6$.

LISTING 59. Example code

```
{qs, r} = PolynomialReduce[poly, gb, {x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6}];
```

Where r is our solution in $K1$, $K2$, $K3$, $K4$, $K5$ and $K6$. And the following code validates correctness:

LISTING 60. Example code

```
poly == r /. ToRules[And @@ eqns] // Expand
```

And please note that, this may take a while. (May be more than a while! It depends on how powerful is your computer.)

7.4. **Checking symmetries in our shift operators.** . First, before starting, we need to know which variables are employed in our functions. For to do this we employ the following code:
Set

LISTING 61. Example code

```
f9 = (2 x1 x2 x5 y2 y5 y6 z2 (x2 y1 y2 z1 + x2 y1 y3 z1 + x3 y1 y3 z1 + x2 y1 y3 z2 + x3
   y1 y3 z2 + x3 y2 y3 z2) z3^2 z4 (x4 x5 y4 z4 + x4 x6 y4 z4 + x4 x6 y5 z4 + x4 x6
   y4 z5 + x4 x6 y5 z5 + x5 x6 y5 z5))/((x1 y1 z1 + x1 y1 z2 + x1 y2 z2 + x2 y2 z2) (
   x2 y2 z2 + x2 y2 z3 + x2 y3 z3 + x3 y3 z3)^2 (x4 y4 z3 + x4 y5 z3 + x5 y5 z3 + x5
   y5 z4)^2 (x5 y5 z4 + x5 y6 z4 + x6 y6 z4 + x6 y6 z5));
```

and

LISTING 62. Example code

```
k1 = ((x1 y1 z1 + x2 y2 z2 + x1 (y1 + y2) z2) (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3))
   /(x2 y2 z2 (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3 + x1 (y2 z2 + (y2 + y3) z3 + y1 (
   z1 + z2 + z3))));
k2 = (((x2 + x3) y1 z1 + x3 (y1 + y2) z2) ((x3 + x4) y2 z2 + x4 (y2 + y3) z3))/(x3 y2 z2
   (x2 y1 z1 + (x3 + x4) (y2 z2 + y1 (z1 + z2)) + x4 (y1 + y2 + y3) z3));
k3 = ((x2 (y2 + y3) z1 + x3 y3 (z1 + z2)) (x3 (y3 + y4) z2 + x4 y4 (z2 + z3)))/(x3 y3 z2
   (x2 (y2 + y3 + y4) z1 + x3 (y3 + y4) (z1 + z2) + x4 y4 (z1 + z2 + z3)));
k4 = ((x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3) (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4))
   /(x3 y3 z3 (x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4 + x2 (y3 z3 + (y3 + y4) z4 + y2
   (z2 + z3 + z4))));
k5 = (((x3 + x4) y2 z2 + x4 (y2 + y3) z3) ((x4 + x5) y3 z3 + x5 (y3 + y4) z4))/(x4 y3 z3
   (x3 y2 z2 + (x4 + x5) (y3 z3 + y2 (z2 + z3)) + x5 (y2 + y3 + y4) z4));
k6 = ((x3 (y3 + y4) z2 + x4 y4 (z2 + z3)) (x4 (y4 + y5) z3 + x5 y5 (z3 + z4)))/(x4 y4
   z3 (x3 (y3 + y4 + y5) z2 + x4 (y4 + y5) (z2 + z3) + x5 y5 (z2 + z3 + z4)));
k7 = ((x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4) (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5))/(
   x4 y4 z4 (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5 + x3 (y4 z4 + (y4 + y5) z5 + y3 (z3
   + z4 + z5))));
k8 = (((x4 + x5) y3 z3 + x5 (y3 + y4) z4) ((x5 + x6) y4 z4 + x6 (y4 + y5) z5))/(x5 y4 z4
   (x4 y3 z3 + (x5 + x6) (y4 z4 + y3 (z3 + z4)) + x6 (y3 + y4 + y5) z5));
k9 = ((x4 (y4 + y5) z3 + x5 y5 (z3 + z4)) (x5 (y5 + y6) z4 + x6 y6 (z4 + z5)))/(x5 y5 z4
   (x4 (y4 + y5 + y6) z3 + x5 (y5 + y6) (z3 + z4) + x6 y6 (z3 + z4 + z5)));
```

Then by using the following code we will obtain the set of our variables which have been employed

LISTING 63. Example code

```
Union[Cases[#, _Symbol, Infinity]] & /@ {f9}
(* {{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, z1, z2, z3, z4, z5}} *)
```

LISTING 64. Example code

```
Union[Cases[#, _Symbol, Infinity]] & /@ {k1, k2, k3, k4, k5, k6, k7, k8, k9}
```

```
(* {{x1, x2, x3, y1, y2, y3, z1, z2, z3}, {x2, x3, x4, y1, y2, y3, z1, z2, z3}, {x2, x3, x4, y2, y3, y4, z1, z2, z3}, {x2, x3, x4, y2, y3, y4, z2, z3, z4}, {x3, x4, x5, y2, y3, y4, z2, z3, z4}, {x3, x4, x5, y3, y4, y5, z2, z3, z4}, {x3, x4, x5, y3, y4, y5, z3, z4, z5}, {x4, x5, x6, y3, y4, y5, z3, z4, z5}, {x4, x5, x6, y4, y5, y6, z3, z4, z5}} *)
```

Now in what comes below, we specifically mean that for example in sl_4 in a process for finding $F_9^{(4)}$, the substitution

$$\{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, z1, z2, z3, z4, z5\}$$

instead of

$$\{y6, y5, y4, y3, y2, y1, x6, x5, x4, x3, x2, x1, z5, z4, z3, z2, z1\}$$

transforms $\tau_9^{(4)}$ to $\tau_1^{(4)}$, $\tau_8^{(4)}$ to $\tau_2^{(4)}$, $\tau_7^{(4)}$ to $\tau_3^{(4)}$, and $\tau_6^{(4)}$ to $\tau_4^{(4)}$ while leaving $F_9^{(4)}$ unchanged.
Therefore, those four pairs must enter the expression for $F_9^{(4)}$ symmetrically.
As a result, the generic polynomials we have been using above, can be reduced greatly in numbers of terms, a factor of $(\frac{2}{3})^4$. Corresponding running time then should be reduced by a factor of $(\frac{2}{3})^8$, other things being equal. It is possible that additional symmetries exist! It needs to be checked!

Here for simplification and for to be able in codding them, we write instead $F_9^{(4)} := F9$ and $\tau_7^{(4)} := Ki$;

LISTING 65. Example code

```
arg1 = {a1, a2, a3, a4, a5, a6, b1, b2, b3, b4, b5, b6, d1, d2, d3, d4, d5};
arg2 = {a6, a5, a4, a3, a2, a1, b6, b5, b4, b3, b2, b1, d5, d4, d3, d2, d1};
arg3 = {b6, b5, b4, b3, b2, b1, a6, a5, a4, a3, a2, a1, d5, d4, d3, d2, d1};
```

LISTING 66. Example code

```
F9[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_ ] := (2 x1 x2 x5 y2 y5 y6 z2 (x2 y1 y2 z1 + x2 y1 y3 z1 +
x3 y1 y3 z1 + x2 y1 y3 z2 + x3 y1 y3 z2 + x3 y2 y3 z2) z3^2 z4 (x4 x5 y4 z4 + x4 x6 y4 z4 + x4 x6 y5 z4 + x4 x6 y4 z5 + x4 x6 y5 z5 + x5 x6 y5 z5)) /((x1 y1 z1 + x1 y1 z2 + x1 y2 z2 + x2 y2 z2) (x2 y2 z2 + x2 y2 z3 + x2 y3 z3 + x3 y3 z3)^2 (x4 y4 z3 + x4 y5 z3 + x5 y5 z3 + x5 y5 z4)^2 (x5 y5 z4 + x5 y6 z4 + x6 y6 z4 + x6 y6 z5))
```

LISTING 67. Example code

```
Simplify[F9 @@ arg1 == F9 @@ arg3]
(* True *)
```

LISTING 68. Example code

```
K1[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_] := ((x1 y1 z1 + x2 y2 z2 + x1 (y1 + y2) z2) (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3 ))/(x2 y2 z2 (x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3 + x1 (y2 z2 + (y2 + y3) z3 + y1 (z1 + z2 + z3)))));
K2[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_] := (((x2 + x3) y1 z1 + x3 (y1 + y2) z2) ((x3 + x4) y2 z2 + x4 (y2 + y3) z3))/(x3 y2 z2 (x2 y1 z1 + (x3 + x4) (y2 z2 + y1 (z1 + z2)) + x4 (y1 + y2 + y3) z3));
```

```
K3[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := ((x2 (y2 + y3) z1 + x3 y3 (z1 + z2)) (x3 (y3 + y4) z2 + x4 y4 (z2 + z3)))/(x3 y3
     z2 (x2 (y2 + y3 + y4) z1 + x3 (y3 + y4) (z1 + z2) + x4 y4 (z1 + z2 + z3)));
K4[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_,  z5_]
     := ((x2 y2 z2 + x3 y3 z3 + x2 (y2 + y3) z3) (x3 y3 z3 +  x4 y4 z4 + x3 (y3 + y4)
    z4))/(x3 y3 z3 (x3 y3 z3 + x4 y4 z4 +  x3 (y3 + y4) z4 + x2 (y3 z3 + (y3 + y4) z4 +
     y2 (z2 + z3 + z4))));
K5[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := (((x3 + x4) y2 z2 + x4 (y2 + y3) z3) ((x4 + x5) y3 z3 + x5 (y3 + y4) z4))/(x4 y3
     z3 (x3 y2 z2 + (x4 + x5) (y3 z3 + y2 (z2 + z3)) + x5 (y2 + y3 + y4) z4));
K6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := ((x3 (y3 + y4) z2 + x4 y4 (z2 + z3)) (x4 (y4 + y5) z3 +  x5 y5 (z3 + z4)))/(x4
    y4 z3 (x3 (y3 + y4 + y5) z2 + x4 (y4 + y5) (z2 + z3) + x5 y5 (z2 + z3 + z4)));
K7[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := ((x3 y3 z3 + x4 y4 z4 + x3 (y3 + y4) z4) (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5)
    )/(x4 y4 z4 (x4 y4 z4 + x5 y5 z5 + x4 (y4 + y5) z5 + x3 (y4 z4 + (y4 + y5) z5 + y3 (
    z3 + z4 + z5))));
K8[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := (((x4 + x5) y3 z3 + x5 (y3 + y4) z4) ((x5 + x6) y4 z4 + x6 (y4 + y5) z5))/(x5 y4
     z4 (x4 y3 z3 + (x5 + x6) (y4 z4 + y3 (z3 + z4)) + x6 (y3 + y4 + y5) z5));
K9[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_, y6_, z1_, z2_, z3_, z4_, z5_]
    := ((x4 (y4 + y5) z3 + x5 y5 (z3 + z4)) (x5 (y5 + y6) z4 + x6 y6 (z4 + z5)))/(x5 y5
     z4 (x4 (y4 + y5 + y6) z3 + x5 (y5 + y6) (z3 + z4) + x6 y6 (z3 + z4 + z5)));
```

LISTING 69. Example code

```
Simplify[K1 @@ arg1 == K9 @@ arg3]
(* True *)
```

LISTING 70. Example code

```
Simplify[K3 @@ arg1 == K7 @@ arg3]
(* True *)
```

LISTING 71. Example code

```
Simplify[K4 @@ arg1 == K6 @@ arg3]
(* True *)
```

Checking symmetries in F6:

We can find the set of variables in a same way as what we did for $F_9^{(4)}$ and so here we omit most of the calculations.

Again as in $F_9^{(4)}$, here we specifically mean that the substitution $\{x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5\}$ instead of $\{x6, x5, x4, x3, x2, x1, y5, y4, y3, y2, y1\}$ transforms $\tau_6^{(4)}$ to $\tau_1^{(4)}$, $\tau_5^{(4)}$ to $\tau_2^{(4)}$, $\tau_4^{(4)}$ to $\tau_3^{(4)}$ while leaving $F_6^{(4)}$ unchanged.

Therefore, those three pairs must enter the expression for $F_6^{(4)}$ symmetrically.

LISTING 72. Example code

```
arg1 = {a1, a2, a3, a4, a5, a6, b1, b2, b3, b4, b5};
arg2 = {a6, a5, a4, a3, a2, a1, b5, b4, b3, b2, b1};
```

LISTING 73. Example code

```
F6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := (2 x1 x2 x5 x6 y2 (x2 y1 +
      x3 (y1 + y2)) y3^2 y4 (x5 y5 + x4 (y4 + y5)))/((x2 y2 + x1 (y1 + y2)) (x3 y3 + x2 (
      y2 + y3))^2 (x4 y3 + x5 (y3 + y4))^2 (x5 y4 + x6 (y4 + y5)));
```

LISTING 74. Example code

```
F6 @@ arg1 == F6 @@ arg2
(* True *)
```

LISTING 75. Example code

```
K1[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x2 y2 + x1 (y1 + y2)) (x3
      y3 + x2 (y2 + y3)))/(x2 y2 (x3 y3 + x2 (y2 + y3) + x1 (y1 + y2 + y3)));
K2[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x2 y1 + x3 (y1 + y2)) (x3
      y2 + x4 (y2 + y3)))/(x3 y2 (x2 y1 + (x3 + x4) (y1 + y2) + x4 y3));
K3[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x3 y3 + x2 (y2 + y3)) (x4
      y4 + x3 (y3 + y4)))/(x3 y3 (x4 y4 + x3 (y3 + y4) + x2 (y2 + y3 + y4)));
K4[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x3 y2 + x4 (y2 + y3)) (x4
      y3 + x5 (y3 + y4)))/(x4 y3 (x3 y2 + (x4 + x5) (y2 + y3) + x5 y4));
K5[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x4 y4 + x3 (y3 + y4)) (x5
      y5 + x4 (y4 + y5)))/(x4 y4 (x5 y5 + x4 (y4 + y5) + x3 (y3 + y4 + y5)));
K6[x1_, x2_, x3_, x4_, x5_, x6_, y1_, y2_, y3_, y4_, y5_] := ((x4 y3 + x5 (y3 + y4)) (x5
      y4 + x6 (y4 + y5)))/(x5 y4 (x4 y3 + (x5 + x6) (y3 + y4) + x6 y5));
```

LISTING 76. Example code

```
Simplify[K1 @@ arg1 == K6 @@ arg2]
(* True *)
```

LISTING 77. Example code

```
Simplify[K2 @@ arg1 == K5 @@ arg2]
(* True *)
```

LISTING 78. Example code

```
Simplify[K3 @@ arg1 == K4 @@ arg2]
(* True *)
```

ACKNOWLEDGEMENT

The research in this thesis would have taken far longer to complete without the encouragement from many individuals and organizations. It is a delight to acknowledge those who have supported me over the last four years.

I would like to express my sincere gratitude to my Supervisor, Prof. Alexei Kanel-Belov, for the continuous support of my M.Sc. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M.Sc. study.

I also want to express my deepest and sincere thanks to Prof. Yaroslav Pugai, who was the main reason and the main incentive for doing this research.

I thank all of the Institute for information transmission problems (Kharkevich Institute)'s staff for their hospitality over the last one and half year.

And I would like to express my special thanks and gratitude to Prof. Andrei Raigorodskii for all his support and encouragements during my study in the Department of Discrete Mathematics at Moscow Institute of Physics and Technology, who without his guidance and constant feedback this M.Sc. would not have been achievable.

And also I would like to express my gratitude and thanks to Moscow Institute of Physics and Technology's staff and especially all my Professors in the Discrete Mathematics Department and the Department itself that most of my theoretical foundations are built in its amazing scientific space and atmosphere.

I am particularly thankful for the help and advice of Prof. Brendan Godfrey, without whom the learning Mathematica would have been very much steeper and unimaginable.
And finally, I would like to thank Professor Boris Feigin for to suggesting me this interesting problem and enlightening discussions during the preparation for my first article which was my first step in this subject!

REFERENCES

[1] *Razavinia, Farrokh. "Local coordinate systems on quantum flag manifolds."* arXiv preprint arXiv:1610.09443 (2016).

[2] *Razavinia, Farrokh. "Weak Faddeev-Takhtajan-Volkov algebras; Lattice W_n algebras."* arXiv preprint arXiv:1702.07402 (2017).

[3] *Kuwabara, Toshiro and Razavinia, Farrokh. "Lecture notes on quantum groups, HSE; Autumn semester 2012."*

[4] *Pugay, Ya P. "Lattice W algebras and quantum groups."* Theoretical and Mathematical Physics 100.1 (1994): 900-911.

[5] *Goursat, Edouard. A Course in Mathematical Analysis: pt. 2. Differential equations.* [c1917. Vol. 2. Dover Publications, 1916.

[6] *Caressa, Paolo. "The algebra of Poisson brackets."* Young Algebra Seminar, Roma Tor Vergata. 2000.

[7] *Hong, Jin, and Seok-Jin Kang. Introduction to quantum groups and crystal bases.* Vol. 42. American Mathematical Soc., 2002.

[8] *Feigin, B.L.*: talk at RIMS 1992.

[9] *Hikami, Kazuhiro, and Rei Inoue. "Classical lattice W algebras and integrable systems."* Journal of Physics A: Mathematical and General 30.19 (1997): 6911.

[10] *Hikami, Kazuhiro. "Lattice WN algebra and its quantization."* Nuclear Physics B 505.3 (1997): 749-770.

[11] *Antonov, Alexander, Alexander A. Belov, and KarÃ©n Chaltikian. "Lattice conformal theories and their integrable perturbations."* Journal of Geometry and Physics 22.4 (1997): 298-318.

[12] *Belov, Alexander A., and Karen D. Chaltikian. "Lattice analogues of W-algebras and classical integrable equations."* Physics Letters B 309.3-4 (1993): 268-274.

[13] *Berenstein, Arkady. "Group-like elements in quantum groups, and Feigin's conjecture."* arXiv preprint q-alg/9605016 (1996).

[14] *Kac V.G.: Infinite-dimensional Lie algebras*. Cambridge University Press 1990.

[15] *Iohara, Kenji, and Feodor Malikov. "Rings of skew polynomials and Gel'fand-Kirillov conjecture for quantum groups."* Communications in mathematical physics **164.2** (1994): 217–237.

[16] *A. Klimyk and K. Schmudgen. Quantum Groups and Their Representations*. Texts and Monographs in Physics. Springer, New York et al., 1997.