Mini-course on GAP – Lecture 5

Leandro Vendramin

Universidad de Buenos Aires

Dalhousie University, Halifax
January 2020

Now let us concentrate on more advanced applications.

- Representation theory
- Non-commutative ring theory: radical rings
- Non-commutative algebra: Fomin–Kirillov algebras

Let us construct the representation ρ of Alt_4 given by

$$(12)(34) \mapsto \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 0 & -1 \end{pmatrix}, \quad (123) \mapsto \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix}.$$

We use the function `GroupHomomorphismByImages`.

```
gap> A4 := AlternatingGroup(4);;
gap> a := [[0,1,-1],[1,0,-1],[0,0,-1]];;
gap> b := [[0,0,-1],[0,1,-1],[1,0,-1]];;
gap> rho := GroupHomomorphismByImages(A4,\
> [ (1,2)(3,4), (1,2,3) ], [ a, b ]);;
gap> IsGroupHomomorphism(rho);
true
```

This is indeed a faithful representation of Alt_4.

```
gap> IsTrivial(Kernel(rho));
true
```

Representation theory

Just to see how it works, let us compute $\rho_{(132)}$, the image of (132) under ρ. `Display`.

```
gap> Display(Image(rho, (1,3,2)));
[ [  -1,   0,   1 ],
  [  -1,   1,   0 ],
  [  -1,   0,   0 ] ]
```

Now we construct the character χ of ρ. We also check that ρ is irreducible since

$$\langle \chi, \chi \rangle = \frac{1}{12} \sum_{g \in \mathrm{Alt}_4} \chi(g)\chi(g^{-1}) = 1.$$

```
gap> chi := x->TraceMat(x^rho);;
gap> 1/Order(A4)*\
> Sum(List(A4, x->chi(x)*chi(x^(-1))));
1
```

A problem of Brauer

Brauer[1] asked what algebras are group algebras. This question might be impossible to answer. However, we can play with some particular examples.

[1] Lectures on Modern Mathematics, Vol I, 133–175, 1963.

A problem of Brauer

Is $\mathbb{C}^5 \times M_5(\mathbb{C})$ a (complex) group algebra? No. We will show that the groups algebras of groups of order 30 are only

$$\mathbb{C}^{10} \times M_2(\mathbb{C})^5, \quad \mathbb{C}^6 \times M_2(\mathbb{C})^6, \quad \mathbb{C}^2 \times M_2(\mathbb{C})^7, \quad \mathbb{C}^{30}.$$

To prove our claim, we can compute the degrees of the irreducible characters using `CharacterDegrees`. There are four groups of order 30 and none of them has a group algebra isomorphic to $\mathbb{C}^5 \times M_5(\mathbb{C})$.

```
gap> n := 30;;
gap> for G in AllGroups(Size, n) do
> Print(CharacterDegrees(G), "\n");
> od;
[ [ 1, 10 ], [ 2, 5 ] ]
[ [ 1, 6 ], [ 2, 6 ] ]
[ [ 1, 2 ], [ 2, 7 ] ]
[ [ 1, 30 ] ]
```

Contructing irreducible representations

How can we construct irreducible representations of a given group? This can be done with the package `Repsn`, written by Vahid Dabbaghian.

Contructing irreducible representations

Let us construct the irreducible representations of Sym_3. The irreducible characters of a finite group can be constructed with `Irr`:

```
gap> S3 := SymmetricGroup(3);;
gap> l := Irr(S3);
[ Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 1, -1, 1 ] ),
  Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 2, 0, -1 ] ),
  Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 1, 1, 1 ] ) ]
```

Contructing irreducible representations

To construct irreducible representations we need to load the package `repsn`:

```
gap> LoadPackage("repsn");
```

The package contains `IrreducibleAffordingRepresentation`. This function produces irreducible representations from irreducible characters.

Since we are working with Sym_3, we will only need to consider the character of degree two. We will produce the faithful represention $\mathrm{Sym}_3 \to \mathbf{GL}(2, \mathbb{C})$ given by

$$(123) \mapsto \begin{pmatrix} \omega^2 & 0 \\ 0 & \omega \end{pmatrix}, \quad (12) \mapsto \begin{pmatrix} 0 & \omega \\ \omega^2 & 0 \end{pmatrix}.$$

Contructing irreducible representations

Here is the code:

```
gap> f := IrreducibleAffordingRepresentation(l[2]);
[ (1,2,3), (1,2) ] ->
[ [ [ E(3)^2, 0 ], [ 0, E(3) ] ],
  [ [ 0, E(3) ], [ E(3)^2, 0 ] ] ]
gap> Image(f, (1,2,3));
[ [ E(3)^2, 0 ], [ 0, E(3) ] ]
gap> Display(Image(f, (1,2,3)));
[ [  E(3)^2,       0 ],
  [       0,    E(3) ] ]
gap> Display(Image(f, (1,2)));
[ [       0,    E(3) ],
  [  E(3)^2,       0 ] ]
```

An exercise on irreducible representations

Construct the irreducible representations of the groups $\mathbb{D}_8$, $\mathbf{SL}_2(3)$, Alt_4, Sym_4 and Alt_5.

The McKay conjecture

For a finite group G and a prime p such that p divides $|G|$ one defines $\mathrm{Irr}_{p'}(G) = \{\chi \in \mathrm{Irr}(G) : p \nmid \chi(1)\}$.

Conjecture (McKay, 1970)

If $P \in \mathrm{Syl}_p(G)$, then $|\mathrm{Irr}_{p'}(G)| = |\mathrm{Irr}_{p'}(N_G(P))|$.

It is believed that the McKay conjecture is true. Recently Malle and Späth proved the conjecture is true for $p = 2$.

The McKay conjecture

We write a naive function that checks the conjecture for a given group.

```
gap> McKay := function(G)
> local N, n, m, p;
> for p in PrimeDivisors(Order(G)) do
> N := Normalizer(G, SylowSubgroup(G, p));
> n := Number(Irr(G), x->Degree(x) mod p <> 0);
> m := Number(Irr(N), x->Degree(x) mod p <> 0);
> if not n = m then
> return false;
> fi;
> od;
> return true;
> end;
function( G ) ... end
```

The McKay conjecture

With this function is now easy to check the McKay conjecture in several small examples.

```
gap> McKay(SL(2,3));
true
gap> McKay(MathieuGroup(11));
true
gap> McKay(SuzukiGroup(8));
true
gap> McKay(PSL(2,7));
true
```

The McKay conjecture

How can we check the conjecture say for other sporadic simple groups?

The package `AtlasRep` provides a nice interface between GAP and databases such as the Atlas of Group Representations. The package contains information of simple groups such as generators, matrix and permutation representations, maximal subgroups, conjugacy classes.

The McKay conjecture

With `AtlasGroup` we create sporadic simple groups. Let us check McKay conjecture for the first Janko group, a non-abelian simple group of order 175560.

```
gap> J1 := AtlasGroup("J1");;
gap> Order(J1);
175560
gap> McKay(J1);
true
```

The Isaacs–Navarro conjecture

For $k \in \mathbb{Z}$ and a finite group G let

$$M_k(G) = |\{\chi \in \mathrm{Irr}p'(G) : \chi(1) \equiv \pm k \bmod p\}|.$$

Conjecture (Isaacs–Navarro, 2002)

If $P \in \mathrm{Syl}_p(G)$, then $M_k(G) = M_k(N_G(P))$.

The Isaacs–Navarro conjecture

Here we have a function that checks the Isaacs–Navarro conjecture:

```
gap> IsaacsNavarro := function(G, k, p)
> local mG, mN, N;
> N := Normalizer(G, SylowSubgroup(G, p));
> mG := Number(Filtered(Irr(G), x->Degree(x)\
> mod p <> 0), x->Degree(x) mod p in [-k,k] mod p);
> mN := Number(Filtered(Irr(N), x->Degree(x)\
> mod p <> 0), x->Degree(x) mod p in [-k,k] mod p);
> if mG = mN then
> return mG;
> else
> return false;
> fi;
> end;
function( G, k, p ) ... end
```

The Isaacs–Navarro conjecture

Let us check that the Isaacs–Navarro conjecture is true for the group $\mathbf{SL}_2(3)$. We only need to check the conjecture for $k \in \{1, 2\}$ and $p \in \{2, 3\}$.

```
gap> IsaacsNavarro(SL(2,3), 1, 2);
4
gap> IsaacsNavarro(SL(2,3), 1, 3);
6
gap> IsaacsNavarro(SL(2,3), 2, 2);
0
gap> IsaacsNavarro(SL(2,3), 2, 3);
6
```

The Ore conjecture

In 1951 Ore conjectured that every element of a finite non-abelian simple group is a commutator. In 2010 Liebeck, O'Brien, Shalev and Tiep proved the Ore conjecture.

In 1896 Frobenius proved that an element g of a finite group is a commutator if and only if

$$\sum \frac{\chi(g)}{\chi(1)} \neq 0,$$

where the sum is over the set of all irreducible characters of G.

We write a function that for a given element g of a group G, returns the sum used by Frobenius to test whether the element g is a commutator of G.

```
gap> IsCommutator := function(group, g)
> local f, s;
> s := 0;
> for f in Irr(group) do
> s := s+g^f/Degree(f);
> od;
> return s;
> end;
function( group, g ) ... end
```

The Ore conjecture

We verify the conjecture for several small non-abelian simple groups.

```
gap> G := AlternatingGroup(5);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := AlternatingGroup(6);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := PSL(2,7);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := PSL(2,8);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
```

The Ore conjecture

The calculations needed only depend on the character table of the group, so we can make things better.

```
gap> Ore := function(ct)
> local f, s, x;
> for x in [1..NrConjugacyClasses(ct)] do
> s := 0;
> for f in Irr(ct) do
> s := s+f[x]/Degree(f);
> od;
> if s = 0 then
> return false;
> fi;
> od;
> return true;
> end;
function( ct ) ... end
```

Now it is easy to verify Ore's conjecture for several simple groups!

```
gap> Ore(CharacterTable("J1"));
true
gap> Ore(CharacterTable("Co1"));
true
gap> check_Ore(CharacterTable("M24"));
true
gap> check_Ore(CharacterTable("Suz"));
true
gap> check_Ore(CharacterTable("HS"));
true
gap> Ore(CharacterTable("B"));
true
gap> Ore(CharacterTable("M"));
true
```

Non-commutative ring theory

A ring R is said to be Jacobson radical if

$$R = \{x \in R : \text{there exists } y \in R \text{ such that } x + y + xy = 0\}.$$

To check whether a finite ring is Jacobson radical:

```
gap> IsJacobsonRadical := function(ring)
> local x, rad;
> rad := [];
> for x in ring do
> if not First(ring,\
> y->x+y+x*y=Zero(ring)) = fail then
> Add(rad, x);
> fi;
> od;
> return Size(ring)=Size(rad);
> end;
function( ring ) ... end
```

The ring $\mathbb{Z}/3$ of integers mod 3 is not Jacobson radical. The subring $\{0, 2\}$ of $\mathbb{Z}/4$ is Jacobson radical.

```
gap> IsJacobsonRadical(Integers mod 3);
false
gap> ring := Integers mod 4;;
gap> subring := Subring(ring, [ZmodnZObj(0,4),\
> ZmodnZObj(2,4)]);;
gap> Elements(subring);
[ ZmodnZObj( 0, 4 ), ZmodnZObj( 2, 4 ) ]
gap> IsJacobsonRadical(subring);
true
```

Non-commutative ring theory

A ring R is Jacobson radical if and only if the operation $R \times R \to R$, $(x, y) \mapsto x \circ y = x + y + xy$, turns R into a group. This group is the adjoint group of the Jacobson radical ring R.

Non-commutative ring theory

Here is the code:

```
gap> AdjointGroup := function(ring)
> local x, y, l, perms;
> if not IsJacobsonRadical(ring) then
> return fail;
> fi;
> perms := NullMat(Size(ring), Size(ring));
> l := AsList(ring);
> for x in l do
> for y in l do
> perms[Position(l, x)][Position(l, y)]:=/
> Position(l, x+y+x*y);
> od;
> od;
> return Group(List(perms, PermList));
> end;
function( ring ) ... end
```

To construct other examples we will use the small rings database included in GAP . The database contains all rings of size ≤ 15. The the number of rings of size n can be computed with `NumberSmallRings`:

```
gap> List([1..15], x->[x, NumberSmallRings(x)]);
[ [ 1, 1 ], [ 2, 2 ], [ 3, 2 ], [ 4, 11 ],
  [ 5, 2 ], [ 6, 4 ], [ 7, 2 ], [ 8, 52 ],
  [ 9, 11 ], [ 10, 4 ], [ 11, 2 ], [ 12, 22 ],
  [ 13, 2 ], [ 14, 4 ], [ 15, 4 ] ]
```

Non-commutative ring theory

To obtain rings from the database one uses the function `SmallRing`.
Several functions that can be used: `Subrings`, `Ideals`, `DirectSum`.

Non-commutative ring theory

The ring $R_{4,3}$ (that is the third ring of size four of the database) is a commutative ring without one.

```
gap> ring := SmallRing(4,3);
<ring with 1 generators>
gap> GeneratorsOfRing(ring);
[ a ]
gap> IsRingWithOne(ring);
false
gap> IsCommutative(ring);
true
```

It is not Jacobson radical:

```
gap> IsJacobsonRadical(ring);
false
gap> AdjointGroup(ring);
fail
```

Non-commutative ring theory

To display multiplication and addition tables we use the functions `ShowMultiplicationTable` and `ShowAdditionTable`.

```
gap> ShowMultiplicationTable(ring);
*    | 0*a a   2*a -a
-----+-----------------
0*a  | 0*a 0*a 0*a 0*a
a    | 0*a a   2*a -a
2*a  | 0*a 2*a 0*a 2*a
-a   | 0*a -a  2*a a
gap> ShowAdditionTable(ring);
+    | 0*a a   2*a -a
-----+-----------------
0*a  | 0*a a   2*a -a
a    | a   2*a -a  0*a
2*a  | 2*a -a  0*a a
-a   | -a  0*a a   2*a
```

Non-commutative ring theory

The ring $R_{8,10}$ (that is the 10-th ring of size eight of the dababase) is non-commutative Jacobson radical ring with adjoint group isomorphic to the dihedral group of eight elements:

```
gap> ring := SmallRing(8,10);
<ring with 2 generators>
gap> IsRingWithOne(ring);
false
gap> IsCommutative(ring);
false
gap> IsJacobsonRadical(ring);
true
gap> StructureDescription(AdjointGroup(ring));
"D8"
```

Non-commutative ring theory

There are 22 radical rings of size eight:

```
gap> n := 8;;
gap> Number([1..NumberSmallRings(n)], \
> x->IsJacobsonRadical(SmallRing(n,x)));
22
```

How many radical rings of size 15 are there?

Non-commutative algebra

Let A be an algebra given by generators and relations. We address the following problems:

- How can we check if A is finite-dimensional?
- Can we compute the center or the radical of A?

Non-commutative algebra

To address these problems our tool is the Gröbner basis package `gbnp`. The package was written by A. Cohen and J. Knopper. First we need to load the package:

```
gap> LoadPackage("gbnp");
```

Non-commutative algebra

Let A be the algebra over $\mathbb{F}_2$ generated by a, b, c, d with relations

$$
\begin{aligned}
&a^2 = b^2 = c^2 = d^2 = 0, \\
&ba + db + ad = 0, \\
&ca + bc + ab = 0, \\
&da + cd + ac = 0, \\
&cb + dc + bd = 0, \\
&cad + bac + dab = 0.
\end{aligned}
$$

Then $\dim A = 36$.

Non-commutative algebra

Here is the code:

```
gap> A := FreeAssociativeAlgebraWithOne(GF(2), \
> "a", "b", "c", "d");;
gap> a := A.a;;
gap> b := A.b;;
gap> c := A.c;;
gap> d := A.d;;
gap> rels := [ a^2, b^2, c^2, d^2, \
> b*a+d*b+a*d, c*a+b*c+a*b, \
> d*a+c*d+a*c, c*b+d*c+b*d, \
> c*a*d+b*a*c+d*a*b ];;
gap> G := SGrobner(GP2NPList(rels));;
gap> DimQA(G,4);
36
```

Non-commutative algebra

For parameters α and β let $C_{\alpha,\beta}$ be the Clifford algebra with generators x, y and relations

$$x^2 = \alpha, \quad y^2 = \alpha, \quad xy + yx = \beta.$$

Let us prove that the algebra $C_{\alpha,\beta}$ is 4-dimensional with basis $1, x, y, xy$. First we need to define the base field with two indeterminates `a,b`:

```
gap> field := FunctionField(Rationals, 2);;
gap> ind := IndeterminatesOfFunctionField(field);;
gap> a := ind[1];;
gap> b := ind[2];;
```

Now we define an associate algebra with generators `x,y` over `field`.

```
gap> A := FreeAssociativeAlgebraWithOne(\
> field, "x", "y");;
gap> x := A.1;;
gap> y := A.2;;
gap> one := One(A);;
```

Non-commutative algebra

We put the defining relations in a list `rels`, compute a Gröbner basis for the ideal of relations and compute the dimension and a linear basis of the algebra with generators `x`,`y` and relations `rels`. `GBNP.ConfigPrint` that gives the name in which the variables will be printed on the screen.

```
gap> GBNP.ConfigPrint("x","y");;
gap> rels := [x^2-one*a,\
> y^2-one*a, x*y+y*x-one*b];;
gap> G := SGrobner(GP2NPList(rels));;
gap> dim := DimQA(G,2);
4
gap> basis := BaseQA(G, 2, dim);;
gap> PrintNPList(basis);
 1
 x
 y
 xy
```

Non-commutative algebra

From `basis` we see that the n-th variable is called with `[[[n]],[1]]`, while the unit of the algebra is `[[[]],[1]]`. We can multiply the n-th and m-th variables via `[[[n,m]],[1]]` or with the function `MulNP`. We can form linear combinations of the variables nth and mth with `[[[n],[m]],[a,b]]`, where a and b are coefficients on the base field. Similarly for products (concatenations of variables). This can also be achieved with the function `AddNP` (or a concatenation of this same function). We can ask for the standard algebraic input of a combination with the command `PrintNP`.

```
gap> PrintNP([[[1]],[1]]);
 x
```

Non-commutative algebra

Once we have computed the Gröbner basis associated to a given ideal, we can ask for a reduced expression on the corresponding quotient with `StrongNormalFormNP`.

```
gap> u := [[[]],[1]];;
gap> r := AddNP(MulNP(x,x),u,1,-a);;
[ [ [ 1, 1 ], [  ] ], [ 1, -x_1 ] ]
gap> PrintNP(r);
 x^2 + -x_1
gap> StrongNormalFormNP(r, G);
[ [  ], [  ] ]
gap> PrintNP(last);
 0
```

Non-commutative algebra

To make calculations we need the structure constants of the algebra.

```
gap> tab := EmptySCTable(dim, Zero(field));;
gap> for x1 in basis do
> for x2 in basis do
> l := [];
> xy := MulQA(x1, x2, G);
> if not IsZero(xy) then
> for k in [1..Size(xy[1])] do
> pos := Position(List(basis, z->z[1][1]),\
> xy[1][k]);
> Add(l, [xy[2][k], pos]);
> od;
> SetEntrySCTable(tab, Position(basis, x1), \
> Position(basis, x2), Flat(l));
> fi;
> od;
> od;
```

Non-commutative algebra

With this code we create a new object for our algebra, now given by its structure contants. With this object one can easily compute things such as the center or the radical.

```
gap> alg := AlgebraByStructureConstants(field,\
> tab);;
gap> Dimension(Center(alg));
1
gap> Dimension(RadicalOfAlgebra(alg));
0
```

Non-commutative algebra

When dealing with algebras defined by homogeneous relations with respect to a certain assignment of weights to the generators, it is also possible to compute a truncated Gröbner basis. In some cases where the full Gröbner basis cannot be computed or takes too much time, this truncated variant can be also used to make computations. It is called with `SGrobnerTrunc(K,p,n)` where K is a list of relations, p is a weight vector and n is truncation degree.

Here we have an example:

```
gap> A := FreeAssociativeAlgebraWithOne(Rationals,\
> "a", "b", "c");;
gap> a := A.a;;
gap> b := A.b;;
gap> c := A.c;;
gap> one := One(A);;
gap> rels := [ a^4,  b^5, b*a-c^3 ];;
gap> K := GP2NPList(rels);;
gap> G := SGrobner(K);;
gap> Gs:= SGrobnerTrunc(K,4,[1,2,1]);;
gap> r := a^4;;
gap> PrintNP(StrongNormalFormNP(\
> [ [ [ 1, 1, 1, 1 ] ], [ 1 ] ],Gs));
0
```

Fomin–Kirillov algebras

Fomin and Kirillov[2] introduced the quadratic algebras $\mathcal{E}_n$ to study the combinatorics of the cohomology of flag manifolds.

Definition:
Let $\mathcal{E}_n$ be the algebra (of type A_{n-1}) with generators $x_{(ij)}$, where $i, j \in \{1, \dots, n\}$, and relations

$$\begin{aligned}
&x_{(ij)} + x_{(ji)} = 0, \\
&x_{(ij)}^2 = 0, \\
&x_{(ij)}x_{(jk)} + x_{(jk)}x_{(ki)} + x_{(ki)}x_{(ij)} = 0, \\
&x_{(ij)}x_{(kl)} = x_{(kl)}x_{(ij)}
\end{aligned}$$

for any distinct i, j, k, l.

[2] Advances in geometry, 147–182, Progr. Math., 172, Birkhäuser, 1999

Fomin–Kirillov algebras

Remarks:

- $\mathcal{E}_n$ is quadratic,
- $\mathcal{E}_n$ is graded: $\deg(x_{(ij)}) = 1$,
- $\mathcal{E}_n$ provides a solution for the classical Yang-Baxter equation:

$$[x_{(ij)}, x_{(jk)}] = [x_{(jk)}, x_{(ik)}] + [x_{(ik)}, x_{(ij)}]$$

where $[u, v] = uv - vu$ is the usual commutator.

Fomin–Kirillov algebras

The algebra $\mathcal{E}_3$ with generators a, b, c and relations

$$a^2 = 0, \quad b^2 = 0, \quad c^2 = 0, \quad ab + ca + bc = 0, \quad ba + ac + cb = 0.$$

satisfies $\dim \mathcal{E}_3 = 12$ and

$$1,\ a,\ b,\ c,\ ab,\ ac,\ ba,\ bc,\ aba,\ abc,\ bac,\ abac$$

is a linear basis of the algebra.

Here is the algebra:

```
gap> A := FreeAssociativeAlgebraWithOne(Rationals,\
> "a", "b", "c");;
gap> a := A.a;;
gap> b := A.b;;
gap> c := A.c;;
gap> one := One(A);;
gap> rels := [ a^2, b^2, c^2, a*b+b*c+c*a,\
> a*c+b*a+c*b ];;
gap> K := GP2NPList(rels);;
gap> G := SGrobner(K);;
```

The calculations:

```
gap> DimQA(G, 3);
12
gap> basis := BaseQA(G, 3, [1, 1, 1]);;
gap> PrintNPList(basis);
 1
 a
 b
 c
 ab
 ac
 ba
 bc
 aba
 abc
 bac
 abac
```

Fomin–Kirillov algebras

The Hilbert series of a graded algebra $B = \oplus_{n \in \mathbb{N}} B_n$ is

$$H(t) = \sum_{n \geq 0} \dim(B_n) t^n.$$

Problems (Fomin and Kirillov)

- Is $\mathcal{E}_n$ finite-dimensional?
- If $\mathcal{E}_n$ is finite-dimensional, compute $\dim \mathcal{E}_n$.
- Compute the Hilbert series of $\mathcal{E}_n$.

Prove that $\mathcal{E}_4$ and $\mathcal{E}_5$ are finite-dimensional:

	dimension
$\mathcal{E}_4$	576
$\mathcal{E}_5$	8294400

Fomin–Kirillov algebras

Prove that the Hilbert series of $\mathcal{E}_6$ is

$$\mathcal{H}(t) = 1 + 15t + 125t^2 + 765t^3 + 3831t^4 + 16605t^5 + \cdots$$

How many coefficients can you compute?

Conjectures

- $\dim \mathcal{E}_n = \infty$ for $n \geq 6$.
- $\dim(\mathcal{E}_n)_k \sim \binom{\binom{n}{2}}{k}$ for $k \to \infty$.