MSRI GRADUATE WORKSHOP: COURSE ON NONCOMMUTATIVE PROJECTIVE GEOMETRY

D. ROGALSKI

1. Exercise set 1

1. The point of this exercise is to use the computer algebra system GAP to do some noncommutative Grobner basis calculations. Some of these would be doable by hand, but you will get a little exposure to GAP this way and see how easy it is to compute a Grobner basis by computer.

If you did not load all of the packages to GAP, you need to at least have the noncommutative grobner bases package (GBNP). Open a Gap window. First run the following code once:

```
LoadPackage("GBNP");
SetInfoLevel(InfoGBNP,0);
SetInfoLevel(InfoGBNPTime,0);
```

The basic procedure you use for this exercise is the following. The third line with the relations changes depending on what relations you want to find a Grobner basis for. It calculates the Grobner basis using degree lex order with $x < y < z$, and up to degree 12. If the results up to degree 12 are still not a Grobner basis, the program stops anyway. The number 12 can be adjusted in the last line. Run the following:

```
A:=FreeAssociativeAlgebraWithOne(Rationals, "x", "y", "z");
x:=A.x;; y:=A.y;; z:=A.z;; o:=One(A);;
uerels:=[y*x - 2*x*y, z*y - 3*y*z, z*x - 5*x*z];
uerelsNP:=GP2NPList(uerels);;
PrintNPList(uerelsNP);
GBNP.ConfigPrint(A);
GB:=SGrobnerTrunc(uerelsNP, 12, [1,1,1]);;
PrintNPList(GB);
```

The calculation should return the original set of relations and nothing extra, because they are already a Grobner basis (as we checked, the single overlap resolves.)

Date: June 17, 2012.

Now try the following further examples:

(A) Run the following two calculations, each of which computes (part of a) Grobner basis for the Jordan plane.

```
A:=FreeAssociativeAlgebraWithOne(Rationals, "x", "y");
x:=A.x;; y:=A.y;; o:=One(A);;
uerels:=[y*x - x*y - x*x];
uerelsNP:=GP2NPList(uerels);;
PrintNPList(uerelsNP);
GBNP.ConfigPrint(A);
GB:=SGrobnerTrunc(uerelsNP, 12, [1,1,1]);;
PrintNPList(GB);
```

```
A:=FreeAssociativeAlgebraWithOne(Rationals, "x", "y");
x:=A.x;; y:=A.y;; o:=One(A);;
uerels:=[x*y - y*x - y*y];
uerelsNP:=GP2NPList(uerels);;
PrintNPList(uerelsNP);
GBNP.ConfigPrint(A);
GB:=SGrobnerTrunc(uerelsNP, 12, [1,1,1]);;
PrintNPList(GB);
```

Explain the difference between the calculations. In the second one, the calculation suggests what the full Grobner basis would be if the computer were not cut off at degree 12. (You can prove exactly what you would get for the full Grobner basis, by induction.) Calculate the corresponding k-basis of reduced words and verify that it also has the same expected Hilbert series.

(B). In this example, you run the program for the Sklyanin algebra

$$S = k\langle x, y, z\rangle/(ayx + bxy + cz^2, axz + bzx + cy^2, azy + byz + cx^2),$$

up to degree 8 only. One first has to specify a, b, c; we chose $2, 3, 5$ but any three not too special integers would give a similar result.

```
A:=FreeAssociativeAlgebraWithOne(Rationals, "x", "y", "z");
x:=A.x;; y:=A.y;; z:=A.z;; o:=One(A);;
a:=2;
```

```
b:=3;
c:=5;
uerels:=[a*y*x + b*x*y + c*z*z, a*x*z + b*z*x + c*y*y, a*z*y + b*y*z + c*x*x];
uerelsNP:=GP2NPList(uerels);;
PrintNPList(uerelsNP);
GBNP.ConfigPrint(A);
GB:=SGrobnerTrunc(uerelsNP, 8, [1,1,1]);;
PrintNPList(GB);
```

Scroll through and look at what sort of output you get. As far as I know, no one has managed to get something useful from Grobner bases for this algebra.

Remark 1.1. Although the Grobner basis calculation for the Sklyanin algebra with general values of a, b, c as above will be infinite, I wonder is there a predictable pattern to the leading words of the new relations the algorithm produces? I have no idea if anyone has studied this.

You could also play around with changing a, b, c. For some special values the Grobner basis is finite. For example, you could take $a = b = c = 1$. However, in this case the algebra ends up having exponential growth (prove it from the Grobner basis you get!), whereas for generic a, b, c it is known to have GK-dimension 3.

(C) Consider the following algebra. It will turn out to be ArtinSchelter regular. I present this as an example where the given three relations are not a grobner basis, and yet there is a still a small finite grobner basis that allows you to compute the Hilbert series of the algebra. Let

$$A = k\langle x, y, z\rangle/(zx - xz, zy - yz, z^2 - x^2 - y^2).$$

Run the following:

```
A:=FreeAssociativeAlgebraWithOne(Rationals, "x", "y", "z");
x:=A.x;; y:=A.y;; z:=A.z;; o:=One(A);;
uerels:=[z*x  + x*z, y*z  + z*y, z*z - x*x - y*y];
uerelsNP:=GP2NPList(uerels);;
PrintNPList(uerelsNP);
GBNP.ConfigPrint(A);
GB:=SGrobnerTrunc(uerelsNP, 12, [1,1,1]);;
PrintNPList(GB);
```

Using the Grobner basis the program produces, Calculate a k-basis of reduced words for this algebra and prove that the algebra has Hilbert series $1/(1-t)^3$.

2. Consider rings of the form $k\langle x, y\rangle/(f)$ where $0 \neq f$ is a single degree 2 relation, so $f = ax^2 + bxy + cyx + dy^2$ for some $a, b, c, d \in k$.

It turns out that up to isomorphism, A is one of the four following algebras: (1) the quantum plane ($f = yx - qxy$ for some $0 \neq q \in k$); (2) the Jordan plane ($f = yx - xy - x^2$); (3) $f = yx$; or (4) $f = x^2$. (To show this, one applies an invertible change of variable $x' = ax + by, y' = cx + dy$, which does not change the algebra $k\langle x, y\rangle/(f)$ up to isomorphism, but which changes f to something simpler. You could work on proving this as part of the exercise, but it is a bit tedious.)

(a) In cases (3) and (4), write down a graded free resolution of the trivial module k_A (we did the Jordan plane in class and the quantum plane is similar). Recalling that the length of a minimal resolution of k_A is the same as the global dimension of A, What is the global dimension of each algebra?

(b). Applying the diamond lemma to examples (3) and (4), find a k-basis of reduced words for each algebra. What are the GK-dimensions and Hilbert series of these algebras?

2. Additional exercises

3. Consider the free algebra $A = k\langle x, y\rangle$, but graded so that x has degree 1 and y has degree 2. Calculate $\dim_k A_n$ for small n, and notice the pattern; prove the pattern continues for all n. Then prove that $h_A(t) = 1/(1-t-t^2)$.

4. Let A be finitely generated connected $\mathbb{N}$-graded and suppose that $P = \oplus_{i=1}^m A[-s_i]$ and $Q = \oplus_{j=1}^n A[-t_j]$ are graded free modules of finite rank.

Suppose that $\phi : P \to Q$ is a graded homomorphism of right A-modules. We think of P and Q as column vectors and ϕ as left multiplication by the matrix M of homogeneous elements in A.

Applying the functor $\mathrm{Hom}_A(-, A)$, we get the map $\phi^* : \mathrm{Hom}_A(Q, A) \to \mathrm{Hom}_A(P, A)$ which we can identify with $\phi^* : \oplus_{j=1}^n A[t_j] \to \oplus_{i=1}^m A[s_i]$. This is a map of graded left A-modules.

($\mathrm{Hom}_A(N, A)$ is a left module for any right module N because A is an $(A-A)$-bimodule. The left module action is $[a \cdot \psi](x) = a\psi(x)$, where $\psi \in \mathrm{Hom}_A(N, A)$, $a \in A$, $x \in N$.)

Show that if we now think of these free modules as row vectors, then the map ϕ^* is represented by *right* multiplication by the *same* matrix M.