/home/m262f99/CHYZAK/worksheetsV.4/oredemo1.mws

Math 262a, Fall 1999, Glenn Tesler
Ore Algebras, Non-Commutative Division and Euclidean Algorithm
11/2/99

This uses software packages by Frederick Chyzak. Follow the installation instructions on the class homepage, or the following commands will not work.

```
> restart;
> with(Ore_algebra); with(Groebner); with(Holonomy);
  with(Mgfun);
```

[*Ore_to_DESol*, *Ore_to_RESol*, *Ore_to_diff*, *Ore_to_shift*, *annihilators*, *applyopr*, *diff_algebra*, *poly_algebra*, *qshift_algebra*, *rand_skew_poly*, *shift_algebra*, *skew_algebra*, *skew_elim*, *skew_gcdex*, *skew_pdiv*, *skew_power*, *skew_prem*, *skew_product*]

[*fglm_algo*, *gbasis*, *gsolve*, *hilbertdim*, *hilbertpoly*, *hilbertseries*, *inter_reduce*, *is_finite*, *is_solvable*, *leadcoeff*, *leadmon*, *leadterm*, *normalf*, *pretend_gbasis*, *reduce*, *spoly*, *termorder*, *testorder*, *univpoly*]

[*algeq_to_dfinite*, *dfinite_add*, *dfinite_mul*, *holon_closure*, *holon_defint*, *holon_defqsum*, *holon_defsum*, *holon_diagonal*, *hypergeom_to_dfinite*, *takayama_algo*]

[*diag_of_sys*, *int_of_sys*, *pol_to_sys*, *sum_of_sys*, *sys*sys*, *sys+sys*]

Define a noncommutative algebra with a shift operator
Sn f(n,k,x) = f(n+1,k,x)
and a difference (Delta) operator
Dk f(n,k,x) = f(n,k+1,x)-f(n,k,x)
and a differential operator
Dx f(n,k,x) = d/dx f(n,k,x)
(There are only so many letters available, so he used D to denote these two separate things.)
There are other predefined types of Ore algebras, as well as the ability to define your own.

```
> A:=skew_algebra(shift=[Sn,n],delta=[Dk,k],diff=[Dx,x],po
  lynom=k);
```

$$A := Ore\_algebra$$

We can multiply two operators together with skew_product(f,g,A):

```
> showprod := proc(f,g,A)
        print(f &* g = skew_product(f,g,A))
  end:
> showprod(x,Dx,A);
```

$$x \mathbin{\&*} Dx = x\,Dx$$

```
> showprod(Dx,x,A);
```

$$Dx \mathbin{\&*} x = 1 + x\,Dx$$

```
> showprod(n,Sn,A); showprod(Sn,n,A);
```

$$n \mathbin{\&*} Sn = n\,Sn$$

$$Sn \mathbin{\&*} n = (n+1)\,Sn$$

```
> showprod(Dx,x^2,A);
```

$$Dx \mathbin{\&*} x^2 = Dx\,x^2 + 2\,x$$

Warning: Maple understands commutative polynomials, but doesn't really understand noncommutative ones. Chyzak's software recognizes this and works around it. Any monomial in x, Dx (or n, Sn, etc.) that is properly of the form x^a Dx^b may be represented by Maple in either order; that way, or Dx^b x^a, but Chyzak's software assumes it's intended the x's be left and the Dx's be right.

```
> showprod(Sn*Dx,n*x^2,A);
```

$$Sn\,Dx \mathbin{\&*} n\,x^2 = (2\,x\,n + 2\,x)\,Sn + (n\,x^2 + x^2)\,Sn\,Dx$$

We can also apply an operator to a function:

```
> applyopr(Dx,sin(x),A);
```

$$\cos(x)$$

```
> applyopr(Dx,x,A);
```

$$1$$

```
> applyopr(Dx,x^2,A);
```

$$2\,x$$

Random skew polynomials can be generated (for instance, to create random input for routines):

```
> rand_skew_poly(x,A);
```

$$-85\,x^5 - 55\,x^4 - 37\,x^3 - 35\,x^2 + 97\,x + 50$$

```
> rand_skew_poly([x,Dx],A);
```

$$-8\,x^5 - 93\,x^4 + (45\,x + 43\,x^4)\,Dx + (92 - 62\,x^3)\,Dx^2$$

```
> rand_skew_poly([x,Dx],terms=5,A);
```

$$-61 - 50\,Dx - 12\,x^3 - 18\,Dx^3 + 31\,x^2\,Dx^2$$

Application. Find operators in this algebra that annihilate binomial(n,k)

```
> el := hypergeom_to_dfinite(binomial(n,k),A);
```

$$el := [Dk\,(k+1) - n + 2\,k + 1,\, Dx,\, Sn\,(n+1-k) - n - 1]$$

Verify that they do annihilate it. Apply the operators to the function.

```
> applyopr(el[1],binomial(n,k),A);
```

$$(1 - n + 2\,k)\,\mathrm{binomial}(n, k) + (k+1)\,(\mathrm{binomial}(n, k+1) - \mathrm{binomial}(n, k))$$

```
> sumtools[simpcomb](");
```

$$0$$

```
> map(applyopr,el,binomial(n,k),A);
```

$$[(1 - n + 2\,k)\,\mathrm{binomial}(n, k) + (k+1)\,(\mathrm{binomial}(n, k+1) - \mathrm{binomial}(n, k)),\, 0,$$
$$(-n-1)\,\mathrm{binomial}(n, k) + (n+1-k)\,\mathrm{binomial}(n+1, k)]$$

```
> map(sumtools[simpcomb],");
```

$$[0, 0, 0]$$

Noncommutative division in K(n)[Sn]

```
> A := shift_algebra([Sn,n]);
```

$$A := Ore\_algebra$$

```
> f1 := skew_power((n+1)*Sn,2,A);
```

$$f1 := (n^2 + 3\,n + 2)\,Sn^2$$

```
> f2 := skew_product(Sn+5,f1,A) + Sn+9;
```

$$f2 := (5\,n^2 + 15\,n + 10)\,Sn^2 + (n^2 + 5\,n + 6)\,Sn^3 + Sn + 9$$

```
> d1 := skew_pdiv(f2,(n+1)*Sn,Sn,A);
```

$$d1 := [n+1,\, Sn^2\,n^2 + 3\,Sn^2\,n + 5\,Sn\,n^2 + 10\,Sn\,n + 2\,Sn^2 + 5\,Sn + 1,\, 9\,n + 9]$$

```
> skew_product(d1[1],f2,A) -
  skew_product(d1[2],(n+1)*Sn,A);
```

$$9\,n + 9$$

```
> d2 := skew_pdiv(f2,(n+1)*Sn,n,A);
```

$$d2 := [1,\, Sn^2\,n + 2\,Sn^2 + 5\,Sn\,n + 5\,Sn,\, Sn + 9]$$

```
> skew_product(d2[1],f2,A) -
  skew_product(d2[2],(n+1)*Sn,A);
```

$$Sn + 9$$

skew_pdiv(p,q,x,A) ---> [u, v, r]
where u*p - v*q = r of x-degree lower than q.
v and r are polynomials in x, while u is a coefficient.

```
> skew_pdiv((n+3)^5,n^2+2,n,A);
```

$$[1,\, n^3 + 15\,n^2 + 88\,n + 240,\, 229\,n - 237]$$

```
> skew_pdiv((n+Sn)^3,Sn+n^2,n,A);
```

$$[1, n+3\,Sn, -7\,Sn\,n+3\,Sn^2\,n+Sn^3-3\,Sn-3\,Sn^2]$$

```
> skew_product("[2],Sn+n^2,A) + "[3];
```

$$n^3+(7\,n+3\,n^2+3)\,Sn-7\,Sn\,n+3\,Sn^2\,n+Sn^3-3\,Sn$$

Noncommutative Euclidean Algorithm for K<n,Sn>

skew_gcdex(p,q,x,A)

- The function skew_gcdex performs an extended skew gcd algorithm on the skew polynomials p and q viewed as polynomials in x with coefficients in their other indeterminates. It returns a list [g,a,b,u,v] such that up+vq=0 and ap+bq=g. Hence, g is a gcd of p and q (in an algebra where all coefficient indeterminates are invertible), while up and vq are lcm's of p and q.

```
> P := skew_product(Sn^2+n*Sn+3,Sn+n,A);
  Q := skew_product(Sn^3+n*Sn+3,Sn+n,A);
  G := skew_gcdex(P,Q,Sn,A);
```

$$P := 3\,n+(3+n^2+n)\,Sn+(2\,n+2)\,Sn^2+Sn^3$$

$$Q := Sn^4+3\,n+(3+n^2+n)\,Sn+Sn^2\,n+(n+3)\,Sn^3$$

$$G := [9\,n^2+54\,Sn+9\,Sn\,n+54\,n,\ 4\,n^2+n^3+9-(4\,n+3)\,Sn-(3-2\,n-n^2)\,Sn^2,$$
$$9+3\,n-4\,n^2-n^3-(-3+2\,n+n^2)\,Sn,$$
$$-Sn\,n^2-9\,Sn\,n-3\,n-Sn^3\,n-6\,Sn^3+Sn^2-21-17\,Sn,$$
$$Sn\,n^2+9\,Sn\,n+Sn^2\,n+3\,n+6\,Sn^2+17\,Sn+21]$$

This says the GCD of (P,Q) is

```
> G[1];
```

$$9\,n^2+54\,Sn+9\,Sn\,n+54\,n$$

```
> factor(");
```

$$9\,(6+n)\,(n+Sn)$$

(Instead of working in K(n)[Sn], we use only polynomials, i.e., K<n,Sn>, so this K(n)-multiple of what we expected (Sn+n) was produced to keep denominators clear.)
This GCD can be expressed as a linear combination g=a*P+b*Q:

```
> skew_product(G[2],P,A)+skew_product(G[3],Q,A);
```

$$(-21+10\,n^2+2\,n+2\,n^3)\,Sn^4+54\,n+(7\,n^3-12\,n+9\,n^2+5\,n^4+18+n^5)\,Sn$$
$$+(-15+9\,n^3+5\,n^2+2\,n^4-8\,n)\,Sn^2+(12\,n^2-19\,n+8\,n^3-30+n^4)\,Sn^3$$
$$+(-3+2\,n+n^2)\,Sn^5+(21-10\,n^2-2\,n-2\,n^3)\,Sn^4+9\,n^2$$
$$+(36-7\,n^3-9\,n^2-n^5+21\,n-5\,n^4)\,Sn+(-9\,n^3+15-5\,n^2-2\,n^4+8\,n)\,Sn^2$$
$$+(-12\,n^2+19\,n-8\,n^3+30-n^4)\,Sn^3+(3-2\,n-n^2)\,Sn^5$$

```
> collect(",Sn,factor);
```

$$(54 + 9\,n)\,Sn + 9\,n\,(6 + n)$$

```
> simplify("-G[1]);
```

$$0$$

Also, the LCM can be computed as a multiple of P or of Q:

```
> skew_product(G[4],P,A):
  collect(",Sn,factor);
```

$$(-n - 6)\,Sn^6 + (-2\,n^2 - 47 - 20\,n)\,Sn^5 + (-101 - n^3 - 14\,n^2 - 64\,n)\,Sn^4$$
$$+ (-134 - 24\,n^2 - 95\,n - 2\,n^3)\,Sn^3 + (-n^4 - 141\,n - 121 - 55\,n^2 - 12\,n^3)\,Sn^2$$
$$+ (-6\,n^3 - 108\,n - 54\,n^2 - 114)\,Sn - 9\,n\,(7 + n)$$

```
> skew_product(G[5],Q,A):
  collect(",Sn,factor);
```

$$(6 + n)\,Sn^6 + (2\,n^2 + 47 + 20\,n)\,Sn^5 + (101 + n^3 + 14\,n^2 + 64\,n)\,Sn^4$$
$$+ (134 + 24\,n^2 + 95\,n + 2\,n^3)\,Sn^3 + (n^4 + 141\,n + 121 + 55\,n^2 + 12\,n^3)\,Sn^2$$
$$+ (6\,n^3 + 108\,n + 54\,n^2 + 114)\,Sn + 9\,n\,(7 + n)$$

```
>
```