Modular curve X1(11)

1 Modular curve example

Example I_1 in Greuel, Laplagne, Seelisch, can be found in Magma's `Normalisation` description as an example with commentary:

```
> // now try a harder case - a singular affine form of modular curve X1(11)
> I := ideal<P | (x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2)>;
> time Js := Normalisation(I: FFMin := false);
Time: 0.110
> #Js;
1
> J := Js[1][1];
> Groebner(J);
> J;
Ideal of Polynomial ring of rank 5 over Rational Field
Lexicographical Order
Variables: $.1, $.2, $.3, $.4, $.5
Groebner basis:
[
    $.1*$.3 - $.1 - 6*$.3 + $.4*$.5^2 - 4*$.4*$.5 + 6*$.4 - $.5^5 + $.5^4 +
        11*$.5^3 - 16*$.5^2 + 2*$.5 + 6,
    $.1*$.4 + 2*$.3 - $.4*$.5^2 + 2*$.4*$.5 - 2*$.4 + $.5^4 - 4*$.5^3 + 4*$.5^2
        - 2,
    $.1*$.5 - 2*$.3 + $.4 + $.5^3 - 2*$.5^2 + $.5 + 1,
    $.2 - $.3 + $.5^3 - $.5^2,
    $.3^2 + 3*$.3 - 2*$.4*$.5^2 + 4*$.4*$.5 - 4*$.4 - $.5^6 + 2*$.5^5 + $.5^4 -
        10*$.5^3 + 10*$.5^2 - 4,
    $.3*$.4 - $.3 - $.4*$.5^3 + $.4*$.5^2 - $.4*$.5 + $.4 - $.5^4 + 2*$.5^3 -
        2*$.5^2 + 1,
    $.3*$.5 + $.3 - $.4 - $.5^4 + 2*$.5^2 - $.5 - 1,
    $.4^2 - 2*$.4*$.5^2 + $.4*$.5 + $.4 - $.5^5
]
> time Js := Normalisation(I);
Time: 1.110
> J := Js[1][1];
> Groebner(J);
> J;
Ideal of Polynomial ring of rank 2 over Rational Field
Lexicographical Order
Variables: $.1, $.2
Groebner basis:
[
    $.1^2*$.2 + 2*$.1*$.2 + $.1 - $.2^2 + 2*$.2 + 1
```

```
]
> // Minimised result is a cubic equation in K[x,y] - as good as we could get!
> // This example takes MUCH longer with the general method - even setting
> // UseMax := true.
```

It seems to be a curve of genus 1, so it should have a presentation (as it does a few lines above) with *one*, I repeat *1* , relation.

[Note that I had to run it without the `FFmin :  false`

```
F:=Rationals();
P<x,y>:=PolynomialRing(F,2);
I:=ideal<P|(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2)>;
time Js:= Normalisation(I);
J := Js[1][1];
time Groebner(J);
J;

time: 0.180
Time: 0.000
[
    $.1^2*$.2 + 2*$.1*$.2 + $.1 - $.2^2 + 2*$.2 + 1
]
```

to get the result claimed, which slowed down the running time slightly.

2 Type I variation

The *type I* variant of this is gotten by using the new variables $z := y - x$ and $w := zx$ to get

$$w^7 + 3w^6z + w^6 + 3w^5z^3 + 6w^5z^2 + 9w^4z^4 + 4w^3z^6 - w^3z^5 - 3w^2z^7 - 3wz^9 - z^{11} = 0$$

with implict weights $wt(w) = 11$ and $wt(z) = 7$.

This should therefore be a *genus 1 curve* with one relation involving functions of weights 3 and 2 such as:

$$f_3^2 + 6f_3f_2 - 5f_3 - f_2^3 + 4f_2^2 - 23f_2 + 2 = 0$$

produced by the qth-power algorithm implemented in Magma, with

$$w = f_3f_2^4 + 7f_3f_2^3 + 16f_3f_2^2 + 15f_3f_2 + 5f_3 + f_2^5 + 4f_2^4 - 10f_2^3 - 50f_2^2 - 59f_2 - 22$$

$$z = 2f_3f_2^2 + 3f_3f_2 + 2f_3 + 2f_2^3 + 2f_2^2 - 8f_2 - 8.$$

3 Singular

```
                     SINGULAR                                 /
 A Computer Algebra System for Polynomial Computations       /   version 3-1-3
                                                           0<
 by: W. Decker, G.-M. Greuel, G. Pfister, H. Schoenemann     \   March 2011
FB Mathematik der Universitaet, D-67653 Kaiserslautern        \
> LIB "normal.lib";
> ring r=0,(y,x),dp;
>  ideal i=(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2);
> list nor=normal(i);nor;
//   characteristic : 0
//   number of vars : 9
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4) T(5) T(6) T(7)
//        block   2 : ordering dp
//                  : names    y x
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=3y4x5+13y3x6-16x9+13y5x3+36y4x4-y3x5-48x8+y7+22y6x+46y5x2+14y4x3
          +16y3x4+96y2x5-144yx6+6y6-y5x+16y4x2+176y3x3-144y2x4-48y5+96y4x-48y3x2
      _[2]=12y4x5+49y3x6-61x9+52y5x3+131y4x4-183x8+4y7+87y6x+165y5x2+61y4x3
          +61y3x4+366y2x5-549yx6+16y6+61y4x2+671y3x3-549y2x4-183y5+366y4x-183y3x2
      _[3]=9y4x5+36y3x6-45x9+39y5x3+96y4x4-135x8+3y7+65y6x+121y5x2+45y4x3
          +45y3x4+270y2x5-405yx6+12y6+45y4x2+495y3x3-405y2x4-135y5+270y4x-135y3x2
      _[4]=6y4x5+27y3x6-33x9+27y5x3+72y4x4-99x8+2y7+46y6x+90y5x2+33y4x3+33y3x4
          +198y2x5-297yx6+9y6+33y4x2+363y3x3-297y2x4-99y5+198y4x-99y3x2
      _[5]=7y4x5+19y3x6-26x9+y6x2+26y5x3+51y4x4-78x8+4y7+40y6x+67y5x2+26y4x3
          +26y3x4+156y2x5-234yx6+7y6+26y4x2+286y3x3-234y2x4-78y5+156y4x-78y3x2
      _[6]=3y4x6-3x10+13y5x4+9y4x5+12y3x6-34x9+y7x+25y6x2+31y5x3+36y4x4+3y3x5
          +18y2x6-102x8+22y7+42y6x+53y5x2+28y4x3+58y3x4+204y2x5-306yx6+6y6-9y5x
          +43y4x2+374y3x3-306y2x4-102y5+204y4x-102y3x2
      _[7]=3y5x5+22y4x6-25x10+13y6x3+75y5x4+12y4x5+6y3x6-106x9+y8+25y7x+119y6x2
          +58y5x3+43y4x4+25y3x5+150y2x6-318x8+64y7+53y6x+49y5x2+56y4x3+306y3x4
          +636y2x5-954yx6+6y6-75y5x+181y4x2+1166y3x3-954y2x4-318y5+636y4x-318y3x2
      _[8]=y6
> def R=nor[1][1];
> setring R;
> option(redSB);
> ideal s=std(norid);s;
s[1]=y*x^7-x^8+3*y^2*x^5-3*y*x^6+4*y^3*x^3-3*y^2*x^4-y^5+2*y^4*x-y^3*x^2
s[2]=236*T(7)*y*x^2+962*T(7)*x^3-325*T(7)*y^2+403*T(7)*y*x+31*y*x^5-31*x^6
    -236*y^3*x^2+195*y^2*x^3+22*y*x^4+19*x^5-383*y^4+352*y^3*x+410*y^2*x^2
    -595*y*x^3+247*x^4+444*y^3-1171*y^2*x+338*y*x^2-468*y^2
s[3]=236*T(7)*y^2*x+1850*T(7)*x^3-625*T(7)*y^2+775*T(7)*y*x-708*y*x^6+708*x^7
```

```
    -3068*y^2*x^4+1875*y*x^5+1193*x^6-236*y^4*x-5900*y^3*x^2+1791*y^2*x^3
    +4962*y*x^4-1325*x^5-5275*y^4+96*y^3*x+7342*y^2*x^2-4775*y*x^3+475*x^4
    +4884*y^3-4975*y^2*x+650*y*x^2-900*y^2
s[4]=236*T(7)*y^3+2294*T(7)*x^3-775*T(7)*y^2+961*T(7)*y*x-708*y^2*x^5
    -5192*y*x^6+5900*x^7-3068*y^3*x^3-17700*y^2*x^4+17193*y*x^5+3575*x^6
    -236*y^5-5900*y^4*x-28084*y^3*x^2+17693*y^2*x^3+11562*y*x^4-1643*x^5
    -20229*y^4+4924*y^3*x+13050*y^2*x^2-5921*y*x^3+589*x^4+6396*y^3-6169*y^2*x
    +806*y*x^2-1116*y^2
s[5]=236*T(7)*x^4+222*T(7)*x^3-75*T(7)*y^2+93*T(7)*y*x-236*y^2*x^4-11*y*x^5
    +247*x^6-708*y^3*x^2+45*y^2*x^3+1058*y*x^4-395*x^5-869*y^4-64*y^3*x
    +1674*y^2*x^2-1281*y*x^3+57*x^4+1464*y^3-1305*y^2*x+78*y*x^2-108*y^2
s[6]=T(6)*y-T(7)*x
s[7]=354*T(6)*x^3+30*T(7)*x^3+9*T(7)*y^2-115*T(7)*y*x+118*T(7)*x^2-27*y*x^5
    +27*x^6-147*y^2*x^3+76*y*x^4+71*x^5-9*y^4-200*y^3*x+54*y^2*x^2+267*y*x^3
    -139*x^4-86*y^3+251*y^2*x-184*y*x^2-118*x^3+282*y^2-354*y*x
s[8]=10089*T(5)*x-2655*T(6)*x^2-20178*T(6)*x-10279*T(7)*x^3+1477*T(7)*y^2
    +1397*T(7)*y*x-5133*T(7)*x^2-1239*T(7)*y+9204*T(7)*x-4431*y*x^5+4431*x^6
    -8922*y^2*x^3+4691*y*x^4+4231*x^5-1477*y^4-7485*y^3*x-10254*y^2*x^2
    +18192*y*x^3-3407*x^4-16113*y^3+21145*y^2*x-4538*y*x^2-354*x^3+5589*y^2
    -708*y*x+885*x^2+2655*y
s[9]=40356*T(5)*y-23364*T(6)*x^2+73910*T(7)*x^3-12755*T(7)*y^2+7745*T(7)*y*x
    -29028*T(7)*x^2+37524*T(7)*y-88500*T(7)*x+38265*y*x^5-38265*x^6
    +91905*y^2*x^3-104890*y*x^4+12985*x^5+12755*y^4+89400*y^3*x-51042*y^2*x^2
    -30849*y*x^3+18001*x^4+30300*y^3-98981*y^2*x+27850*y*x^2+4956*x^3
    -27288*y^2+9912*y*x+7788*x^2+23364*y
s[10]=13452*T(4)*x-6372*T(6)*x^2-40356*T(6)*x-26714*T(7)*x^3+4187*T(7)*y^2
     +727*T(7)*y*x-4248*T(7)*x^2-5664*T(7)*y+24780*T(7)*x-12561*y*x^5+12561*x^6
     -27717*y^2*x^3+23806*y*x^4+3911*x^5-4187*y^4-25260*y^3*x-4614*y^2*x^2
     +30129*y*x^3-8629*x^4-25128*y^3+44345*y^2*x-9622*y*x^2-3540*x^3+17244*y^2
     -7080*y*x+2124*x^2+6372*y
s[11]=2242*T(4)*y-3186*T(6)*x^2+5928*T(7)*x^3-1089*T(7)*y^2+1171*T(7)*y*x
     -4366*T(7)*x^2+3894*T(7)*y-7788*T(7)*x+3267*y*x^5-3267*x^6+8229*y^2*x^3
     -10612*y*x^4+2383*x^5+1089*y^4+8270*y^3*x-7950*y^2*x^2+615*y*x^3+1243*x^4
     -214*y^3-6299*y^2*x+2086*y*x^2+472*x^3-1790*y^2+944*y*x+1062*x^2+3186*y
s[12]=40356*T(3)-161424*T(4)+242136*T(5)-23364*T(6)*x^2+40356*T(6)*x-161424*T(6)
     +73910*T(7)*x^3-12755*T(7)*y^2+7745*T(7)*y*x-29028*T(7)*x^2+37524*T(7)*y
     -88500*T(7)*x+40356*T(7)+38265*y*x^5-38265*x^6+91905*y^2*x^3-104890*y*x^4
     +12985*x^5+12755*y^4+89400*y^3*x-51042*y^2*x^2-30849*y*x^3+18001*x^4
     +30300*y^3-98981*y^2*x+27850*y*x^2+4956*x^3-27288*y^2+9912*y*x+7788*x^2
     +23364*y
s[13]=40356*T(2)*x+147972*T(4)-363204*T(5)+74340*T(6)*x^2-242136*T(6)*x
     +282492*T(6)-249850*T(7)*x^3+41275*T(7)*y^2-10825*T(7)*y*x+49560*T(7)*x^2
     -113280*T(7)*y+293820*T(7)*x-80712*T(7)-123825*y*x^5+123825*x^6
     -286725*y^2*x^3+293150*y*x^4-6425*x^5-41275*y^4-271500*y^3*x+56490*y^2*x^2
     +194745*y*x^3-62285*x^4-190200*y^3+390145*y^2*x-64874*y*x^2-43896*x^3
     +146520*y^2-87792*y*x-11328*x^2-47436*y+13452*x
```

```
s[14]=40356*T(2)*y-65844*T(6)*x^2+193610*T(7)*x^3-35255*T(7)*y^2+35645*T(7)*y*x
     -124608*T(7)*x^2+111864*T(7)*y-237180*T(7)*x+105765*y*x^5-105765*x^6
     +264705*y^2*x^3-336190*y*x^4+71485*x^5+35255*y^4+264900*y^3*x
     -249762*y^2*x^2+9651*y*x^3+45721*x^4-8400*y^3-203741*y^2*x+102226*y*x^2
     -14160*x^3-17208*y^2-28320*y*x+35400*x^2+106200*y
s[15]=13452*T(1)*x-26904*T(2)+134520*T(4)-201780*T(5)+23364*T(6)*x^2
     -40356*T(6)*x+147972*T(6)-73910*T(7)*x^3+12755*T(7)*y^2-7745*T(7)*y*x
     +29028*T(7)*x^2-37524*T(7)*y+88500*T(7)*x-40356*T(7)-38265*y*x^5+38265*x^6
     -91905*y^2*x^3+104890*y*x^4-12985*x^5-12755*y^4-89400*y^3*x+51042*y^2*x^2
     +30849*y*x^3-18001*x^4-30300*y^3+98981*y^2*x-27850*y*x^2-4956*x^3+27288*y^2
     -9912*y*x-7788*x^2-23364*y-13452
s[16]=20178*T(1)*y-147972*T(4)+363204*T(5)-44604*T(6)*x^2+20178*T(6)*x
     -282492*T(6)+133760*T(7)*x^3-24005*T(7)*y^2+21695*T(7)*y*x-76818*T(7)*x^2
     +74694*T(7)*y-162840*T(7)*x+80712*T(7)+72015*y*x^5-72015*x^6+178305*y^2*x^3
     -220540*y*x^4+42235*x^5+24005*y^4+177150*y^3*x-150402*y^2*x^2-10599*y*x^3
     +31861*x^4+10950*y^3-151361*y^2*x+65038*y*x^2-4602*x^3-22248*y^2-9204*y*x
     +21594*x^2+58056*y+6726*x
s[17]=40356*T(7)^2+4842720*T(4)-15980976*T(5)-1418832*T(6)*x^2+17433792*T(6)
     -6384190*T(7)*x^3+1113331*T(7)*y^2-3186865*T(7)*y*x+210984*T(7)*x^2
     -5722764*T(7)*y+4167996*T(7)*x-6295536*T(7)-3582129*y*x^5+3582129*x^6
     -9501573*y^2*x^3+11907926*y*x^4-2406353*x^5-1153687*y^4-9569796*y^3*x
     +10241934*y^2*x^2-814839*y*x^3-2285741*x^4+1799556*y^3+5495065*y^2*x
     -4810238*y*x^2+1166784*x^3+542556*y^2+2454636*y*x+271164*x^2+1055628*y
s[18]=20178*T(6)*T(7)+585162*T(4)-3087234*T(5)+98766*T(6)*x^2+100890*T(6)*x
     +4418982*T(6)-1373282*T(7)*x^3+271259*T(7)*y^2-469805*T(7)*y*x
     +805704*T(7)*x^2-1365024*T(7)*y+685344*T(7)*x-1816020*T(7)-813777*y*x^5
     +813777*x^6-2153085*y^2*x^3+2909476*y*x^4-756391*x^5-271259*y^4
     -2212002*y^3*x+2713980*y^2*x^2-729057*y*x^3-315439*x^4+771612*y^3
     +1018577*y^2*x-968572*y*x^2+223020*x^3-360306*y^2+459492*y*x+708*x^2
     -78588*y+20178*x
s[19]=40356*T(5)*T(7)-1251036*T(4)-363204*T(5)+528876*T(6)*x^2+403560*T(6)*x
     +4479516*T(6)-1007798*T(7)*x^3+218489*T(7)*y^2-477011*T(7)*y*x
     +1657428*T(7)*x^2-1265196*T(7)*y-603924*T(7)*x-2461716*T(7)-655467*y*x^5
     +655467*x^6-1832559*y^2*x^3+2915842*y*x^4-1083283*x^5-218489*y^4
     -1961820*y^3*x+3375462*y^2*x^2-1985997*y*x^3+135377*x^4+1591548*y^3
     -60337*y^2*x-433030*y*x^2+118944*x^3-1035900*y^2+170628*y*x+160008*x^2
     -84960*y+80712*x
s[20]=6726*T(4)*T(7)-443916*T(4)+363204*T(5)+128502*T(6)*x^2+100890*T(6)*x
     +605340*T(6)-79496*T(7)*x^3+22301*T(7)*y^2-79847*T(7)*y*x+294174*T(7)*x^2
     -179478*T(7)*y-266562*T(7)*x-423738*T(7)-66903*y*x^5+66903*x^6
     -210417*y^2*x^3+398884*y*x^4-188467*x^5-22301*y^4-239190*y^3*x
     +558606*y^2*x^2-438321*y*x^3+74303*x^4+328338*y^3-163117*y^2*x+13892*y*x^2
     -354*x^3-206964*y^2-34338*y*x+58056*x^2+12744*y+20178*x
s[21]=3363*T(2)*T(7)-384503*T(4)+272403*T(5)+115935*T(6)*x^2+90801*T(6)*x
     +608703*T(6)-92739*T(7)*x^3+24057*T(7)*y^2-76823*T(7)*y*x+281312*T(7)*x^2
     -176823*T(7)*y-228153*T(7)*x-403560*T(7)-72171*y*x^5+72171*x^6
```

```
-220002*y^2*x^3+400031*y*x^4-180029*x^5-24057*y^4-246385*y^3*x
+538521*y^2*x^2-402471*y*x^3+62221*x^4+304544*y^3-129176*y^2*x-1955*y*x^2
+3127*x^3-192810*y^2-21771*y*x+49914*x^2+7375*y+17936*x
s[22]=2124*T(1)*T(7)-4248*T(2)-34692*T(4)-12744*T(5)+31860*T(6)*x^2+8496*T(6)*x
+152928*T(6)-35414*T(7)*x^3+7595*T(7)*y^2-16073*T(7)*y*x+59472*T(7)*x^2
-41772*T(7)*y-11328*T(7)*x-84960*T(7)-22785*y*x^5+22785*x^6-63321*y^2*x^3
+99706*y*x^4-36385*x^5-7595*y^4-67560*y^3*x+111414*y^2*x^2-66063*y*x^3
+7019*x^4+47484*y^3+701*y^2*x-5350*y*x^2-1416*x^3-25596*y^2-7788*y*x
+4956*x^2-7080*y+708*x-2124
s[23]=10089*T(6)^2-80712*T(2)+548169*T(4)-1493172*T(5)+290457*T(6)*x^2
-110979*T(6)*x+1805931*T(6)-605644*T(7)*x^3+113236*T(7)*y^2-136108*T(7)*y*x
+486219*T(7)*x^2-453651*T(7)*y+530646*T(7)*x-726408*T(7)-339708*y*x^5
+339708*x^6-866424*y^2*x^3+1150076*y*x^4-283652*x^5-113236*y^4-877860*y^3*x
+938544*y^2*x^2-205170*y*x^3-81986*x^4+132528*y^3+538105*y^2*x-225254*y*x^2
-9027*x^3-1890*y^2-21417*y*x-56463*x^2-196293*y-23541*x-30267
s[24]=10089*T(5)*T(6)-161424*T(2)+1055982*T(4)-2259936*T(5)+509229*T(6)*x^2
-272403*T(6)*x+2280114*T(6)-826633*T(7)*x^3+150679*T(7)*y^2-153481*T(7)*y*x
+584985*T(7)*x^2-543921*T(7)*y+963411*T(7)*x-857565*T(7)-452037*y*x^5
+452037*x^6-1132194*y^2*x^3+1440557*y*x^4-308363*x^5-150679*y^4-1133595*y^3*x
+1027266*y^2*x^2-74823*y*x^3-120206*x^4-38211*y^3+861703*y^2*x-213155*y*x^2
-78057*x^3+203724*y^2-179655*y*x-105846*x^2-330990*y-57171*x-60534
s[25]=3363*T(4)*T(6)-80712*T(2)+585162*T(4)-1220769*T(5)+241605*T(6)*x^2
-144609*T(6)*x+1150146*T(6)-427937*T(7)*x^3+77405*T(7)*y^2-74459*T(7)*y*x
+282138*T(7)*x^2-270633*T(7)*y+517725*T(7)*x-413649*T(7)-232215*y*x^5
+232215*x^6-578328*y^2*x^3+725773*y*x^4-147445*x^5-77405*y^4-576855*y^3*x
+498945*y^2*x^2-4809*y*x^3-72091*x^4-40170*y^3+464561*y^2*x-123583*y*x^2
-35046*x^3+114012*y^2-76818*y*x-60357*x^2-170982*y-30267*x-30267
s[26]=10089*T(2)*T(6)-433827*T(2)+3087234*T(4)-6456960*T(5)+1316349*T(6)*x^2
-776853*T(6)*x+6154290*T(6)-2278537*T(7)*x^3+412594*T(7)*y^2-400234*T(7)*y*x
+1519899*T(7)*x^2-1451931*T(7)*y+2749164*T(7)*x-2229669*T(7)-1237782*y*x^5
+1237782*x^6-3085185*y^2*x^3+3879473*y*x^4-794288*x^5-412594*y^4
-3079005*y^3*x+2679468*y^2*x^2-51909*y*x^3-373742*x^4-201801*y^3+2459584*y^2*x
-639572*y*x^2-195762*x^3+607113*y^2-431880*y*x-317715*x^2-912789*y-161424*x
-161424
s[27]=40356*T(1)*T(6)+40356*T(1)-322848*T(2)+2300292*T(4)-5004144*T(5)
+1191564*T(6)*x^2-686052*T(6)*x+4963788*T(6)-1803670*T(7)*x^3+327733*T(7)*y^2
-326215*T(7)*y*x+1265196*T(7)*x^2-1200768*T(7)*y+2186304*T(7)*x-1937088*T(7)
-983199*y*x^5+983199*x^6-2456859*y^2*x^3+3108530*y*x^4-651671*x^5-327733*y^4
-2456100*y^3*x+2159682*y^2*x^2-102957*y*x^3-256091*x^4-159972*y^3
+1921315*y^2*x-404534*y*x^2-212400*x^3+534636*y^2-478608*y*x-249216*x^2
-747648*y-121068*x-80712
s[28]=13452*T(5)^2-430464*T(2)+3582716*T(4)-7048848*T(5)+1332456*T(6)*x^2
-860928*T(6)*x+5824716*T(6)-2439334*T(7)*x^3+427951*T(7)*y^2-313813*T(7)*y*x
+1282896*T(7)*x^2-1477596*T(7)*y+3170424*T(7)*x-1910184*T(7)-1283853*y*x^5
+1283853*x^6-3124029*y^2*x^3+3694586*y*x^4-570557*x^5-427951*y^4-3066960*y^3*x
+1983606*y^2*x^2+741693*y*x^3-514241*x^4-902064*y^3+3213829*y^2*x-845174*y*x^2
```

```
-168504*x^3+834540*y^2-404268*y*x-385860*x^2-1005124*y-179360*x-161424
s[29]=4484*T(4)*T(5)-215232*T(2)+2004348*T(4)-3981792*T(5)+637200*T(6)*x^2
   -452884*T(6)*x+3188124*T(6)-1365910*T(7)*x^3+239935*T(7)*y^2-178245*T(7)*y*x
   +698324*T(7)*x^2-792252*T(7)*y+1732476*T(7)*x-1008900*T(7)-719805*y*x^5
   +719805*x^6-1753245*y^2*x^3+2078890*y*x^4-325645*x^5-239935*y^4-1722400*y^3*x
   +1157826*y^2*x^2+388689*y*x^3-303985*x^4-443980*y^3+1761533*y^2*x-531694*y*x^2
   -54044*x^3+409604*y^2-117056*y*x-225852*x^2-543036*y-94164*x-80712
s[30]=13452*T(2)*T(5)-1156872*T(2)+10591208*T(4)-20985120*T(5)+3472032*T(6)*x^2
   -2434812*T(6)*x+16868808*T(6)-7224750*T(7)*x^3+1268145*T(7)*y^2-934875*T(7)*y*x
   +3695760*T(7)*x^2-4219680*T(7)*y+9206124*T(7)*x-5367348*T(7)-3804435*y*x^5
   +3804435*x^6-9261135*y^2*x^3+10964250*y*x^4-1703115*x^5-1268145*y^4
   -9094500*y^3*x+6037110*y^2*x^2+2116143*y*x^3-1595043*x^4-2438880*y^3
   +9377655*y^2*x-2759934*y*x^2-323556*x^3+2222892*y^2-714372*y*x-1188732*x^2
   -2889112*y-502208*x-430464
s[31]=171*T(1)*T(5)+342*T(1)-2394*T(2)+25194*T(4)-50958*T(5)+8829*T(6)*x^2-6498*T(6)*x
   +40185*T(6)-17746*T(7)*x^3+3103*T(7)*y^2-2197*T(7)*y*x+8850*T(7)*x^2-10392*T(7)*y
   +22722*T(7)*x-13338*T(7)-9309*y*x^5+9309*x^6-22593*y^2*x^3+26534*y*x^4-3941*x^5
   -3103*y^4-22140*y^3*x+13992*y^2*x^2+5826*y*x^3-3884*x^4-6702*y^3+23494*y^2*x
   -6608*y*x^2-909*x^3+5742*y^2-1932*y*x-3057*x^2-7233*y-1083*x-513
s[32]=6726*T(4)^2-484272*T(2)+5004144*T(4)-10048644*T(5)+1350864*T(6)*x^2
   -1069434*T(6)*x+7869420*T(6)-3416390*T(7)*x^3+601595*T(7)*y^2-458105*T(7)*y*x
   +1721148*T(7)*x^2-1906644*T(7)*y+4230300*T(7)*x-2421360*T(7)-1804785*y*x^5
   +1804785*x^6-4404345*y^2*x^3+5248810*y*x^4-844465*x^5-601595*y^4-4332600*y^3*x
   +3053082*y^2*x^2+867669*y*x^3-791341*x^4-933060*y^3+4286441*y^2*x-1448992*y*x^2
   -49914*x^3+888882*y^2-52746*y*x-584808*x^2-1310508*y-221958*x-181602
s[33]=177*T(2)*T(4)-22833*T(2)+232519*T(4)-465687*T(5)+64782*T(6)*x^2-50445*T(6)*x
   +365505*T(6)-158868*T(7)*x^3+27954*T(7)*y^2-21126*T(7)*y*x+80004*T(7)*x^2
   -89208*T(7)*y+197532*T(7)*x-112926*T(7)-83862*y*x^5+83862*x^6-204534*y^2*x^3
   +243372*y*x^4-38838*x^5-27954*y^4-201120*y^3*x+140112*y^2*x^2+41652*y*x^3
   -36552*x^4-45276*y^3+200580*y^2*x-66405*y*x^2-3009*x^3+42399*y^2-4248*y*x
   -27081*x^2-61301*y-10384*x-8496
s[34]=40356*T(1)*T(4)+121068*T(1)-807120*T(2)+9779604*T(4)-19976220*T(5)
   +2778192*T(6)*x^2-2340648*T(6)*x+15294924*T(6)-6871198*T(7)*x^3+1205329*T(7)*y^2
   -882811*T(7)*y*x+3358752*T(7)*x^2-3826032*T(7)*y+8526444*T(7)*x-4883076*T(7)
   -3615987*y*x^5+3615987*x^6-8798079*y^2*x^3+10402442*y*x^4-1604363*x^5-1205329*y^4
   -8636820*y^3*x+5830134*y^2*x^2+1981695*y*x^3-1585667*x^4-2128656*y^3
   +8782711*y^2*x-2875202*y*x^2-124608*x^3+1864728*y^2-128148*y*x-1222008*x^2
   -2684028*y-376656*x-161424
s[35]=3363*T(2)^2-773490*T(2)+7793192*T(4)-15580779*T(5)+2242944*T(6)*x^2
   -1718493*T(6)*x+12258135*T(6)-5333509*T(7)*x^3+937732*T(7)*y^2-703088*T(7)*y*x
   +2684677*T(7)*x^2-3013779*T(7)*y+6659802*T(7)*x-3803553*T(7)-2813196*y*x^5
   +2813196*x^6-6857007*y^2*x^3+8145861*y*x^4-1288854*x^5-937732*y^4-6739685*y^3*x
   +4639086*y^2*x^2+1443708*y*x^3-1218573*x^4-1585583*y^3+6777405*y^2*x-2195484*y*x^2
   -124903*x^3+1460952*y^2-204966*y*x-905709*x^2-2070310*y-350873*x-285855
s[36]=6726*T(1)*T(2)+40356*T(1)-242136*T(2)+2892180*T(4)-5898702*T(5)+852432*T(6)*x^2
   -706230*T(6)*x+4526598*T(6)-2036990*T(7)*x^3+356975*T(7)*y^2-258805*T(7)*y*x
```

```
     +994268*T(7)*x^2-1142004*T(7)*y+2539596*T(7)*x-1452816*T(7)-1070925*y*x^5
     +1070925*x^6-2603685*y^2*x^3+3072210*y*x^4-468525*x^5-356975*y^4-2554600*y^3*x
     +1698642*y^2*x^2+608481*y*x^3-466473*x^4-660580*y^3+2623221*y^2*x-837708*y*x^2
     -47318*x^3+569622*y^2-65490*y*x-360372*x^2-798624*y-114342*x-47082
s[37]=10089*T(1)^2+30267*T(1)-40356*T(2)+739860*T(4)-1583973*T(5)+228861*T(6)*x^2
     -201780*T(6)*x+1180413*T(6)-556643*T(7)*x^3+97274*T(7)*y^2-68426*T(7)*y*x
     +263553*T(7)*x^2-308865*T(7)*y+691716*T(7)*x-403560*T(7)-291822*y*x^5+291822*x^6
     -707919*y^2*x^3+830347*y*x^4-122428*x^5-97274*y^4-693495*y^3*x+447708*y^2*x^2
     +179367*y*x^3-128128*x^4-192171*y^3+724118*y^2*x-233476*y*x^2-8496*x^3+152946*y^2
     -3540*y*x-106554*x^2-225498*y-23541*x+10089

ring r=0,(x,y),dp;
>  ideal i=(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2);
> list nor=normal(i);nor;
//   characteristic : 0
//   number of vars : 6
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4)
//        block   2 : ordering dp
//                  : names    x y
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=x2y5
      _[2]=x4y4
      _[3]=x5y3+2x3y4+xy5
      _[4]=x7y2-3x3y4-2xy5
      _[5]=y6
> def R=nor[1][1];
> setring R;
> option(redSB);
> ideal s=std(norid);s;
s[1]=x^8-x^7*y+3*x^6*y-3*x^5*y^2+3*x^4*y^2-4*x^3*y^3+x^2*y^3
    -2*x*y^4+y^5
s[2]=6*T(4)*x^3-5*T(4)*x^2*y-T(4)*y^3+6*T(4)*x*y-3*T(4)*y^2
    -3*x^5-x^4-7*x^3*y+18*x^3+5*x^2*y-3*x*y^2+y^3+18*x*y-12*y^2
s[3]=T(4)*y^4-x^7+3*x^3*y^2+2*x*y^3
s[4]=T(4)*x*y^3-x^7+3*x^6-3*x^5*y+6*x^4*y-4*x^3*y^2+3*x^2*y^2
    -2*x*y^3+y^4
s[5]=T(4)*x^2*y^2+3*T(4)*y^3-x^7-3*x^5*y+6*x^5-x^4*y-4*x^3*y^2
    +12*x^3*y-x^2*y^2-x*y^3+y^4+6*x*y^2
s[6]=6*T(3)*y^2+T(4)*x^2*y-T(4)*y^3+3*T(4)*y^2-3*x^5-x^4
    -7*x^3*y-x^2*y-3*x*y^2+y^3
s[7]=6*T(3)*x*y+T(4)*x^2*y-T(4)*y^3+2*T(4)*x*y+T(4)*y^2-3*x^5
    -x^4-7*x^3*y-2*x^3-x^2*y-3*x*y^2+y^3-2*x*y+2*y^2
s[8]=T(3)*x^2-2*T(3)*y-T(4)*y
```

```
s[9]=2*T(2)*y-12*T(3)*y-T(4)*x^2*y+T(4)*y^3-2*T(4)*x^2-T(4)*y^2
    -6*T(4)*y+3*x^5+x^4+7*x^3*y+2*x^3+x^2*y+3*x*y^2-y^3
    +2*x^2-2*y^2
s[10]=T(2)*x+6*T(3)*x-7*T(3)*y+2*T(4)*x-2*T(4)*y-x+2*y
s[11]=T(1)*y-x^2
s[12]=T(1)*x-3*T(3)*x+3*T(3)*y-T(4)*x+T(4)*y+x-y
s[13]=3*T(4)^2-18*T(1)-30*T(2)-81*T(3)*x+180*T(3)*y-18*T(3)
     +4*T(4)*x^2*y-7*T(4)*y^3+15*T(4)*x^2+T(4)*x*y+5*T(4)*y^2
     -33*T(4)*x+69*T(4)*y+3*T(4)-21*x^5-7*x^4-49*x^3*y-x^3-4*x^2*y
     -21*x*y^2+7*y^3-12*x^2+2*x*y+7*y^2+15*x-27*y+12
s[14]=T(3)*T(4)+T(1)+3*T(2)+8*T(3)*x-13*T(3)*y+6*T(3)-T(4)*x^2
     +3*T(4)*x-5*T(4)*y+T(4)+x^2-2*x+2*y-2
s[15]=T(2)*T(4)+4*T(1)+4*T(2)+14*T(3)*x-13*T(3)*y-6*T(3)-T(4)*x^2
     +5*T(4)*x-5*T(4)*y-3*T(4)+x^2-4*x+2*y
s[16]=T(1)*T(4)-T(1)-T(2)-3*T(3)*x+6*T(3)-T(4)*x+3*T(4)+x
s[17]=T(3)^2-T(2)-3*T(3)*x-4*T(3)-T(4)*x-T(4)+x+1
s[18]=T(2)*T(3)-T(1)-T(2)-3*T(3)*x+2*T(3)-T(4)*x+T(4)+x
s[19]=T(1)*T(3)-2*T(3)-T(4)
s[20]=T(2)^2-2*T(1)-3*T(2)-T(4)*x
s[21]=T(1)*T(2)+T(1)+2*T(2)-T(3)*x
s[22]=T(1)^2-T(2)
```

```
> ring r=11,(y,x),dp;
>  ideal i=(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2);
> list norp=normalP(i,"withRing");norp;
//   characteristic : 11
//   number of vars : 6
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4)
//        block   2 : ordering dp
//                  : names    y x
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=y3x4-4y2x5+2y6+2y4x2+y3x3-5y5+5y4x
      _[2]=y4x3-3y2x5-y6+4y4x2-3y5+3y4x
      _[3]=y7+2y2x5-y6+5y3x3-3y5+3y4x
      _[4]=x8+3y2x5+3yx6+2y6-3y4x2+4y3x3+3y2x4-2y5+y4x+y3x2
      _[5]=y5x-5y4x2-5y3x3+5y5-5y4x
> def Rp=norp[1][1];
> setring Rp;
> option(redSB);
> ideal sp=std(norid);sp;
sp[1]=y*x^7-x^8+3*y^2*x^5-3*y*x^6+4*y^3*x^3-3*y^2*x^4-y^5+2*y^4*x-y^3*x^2
sp[2]=T(4)*x^4+3*T(4)*y^3-5*T(4)*y^2*x-4*T(4)*y*x^2-2*T(4)*x^3-4*T(4)*y^2
     -2*T(4)*y*x+5*x^6+4*y*x^4-3*x^5-2*y^2*x^2+5*y*x^3+x^4-4*y^3-4*y^2*x
     +2*y*x^2-4*x^3-4*y^2-4*y*x
sp[3]=T(4)*y*x^3+3*T(4)*y^3+3*T(4)*y^2*x+T(4)*y*x^2-4*T(4)*x^3+3*T(4)*y^2
     -4*T(4)*y*x+5*x^6+4*y*x^4-3*x^5-4*y^2*x^2-3*y*x^3-2*x^4+2*y^3+4*y^2*x
     +3*x^3+3*y^2+3*y*x
sp[4]=T(4)*y^2*x^2+3*T(4)*y^3+5*x^6+4*y*x^4-3*x^5-2*y^3*x-y^2*x^2+5*y*x^3
     -4*y^3-3*y^2*x
sp[5]=T(4)*y^3*x-3*T(4)*y^3-2*T(4)*y^2*x-T(4)*y*x^2+4*T(4)*x^3-3*T(4)*y^2
     +4*T(4)*y*x+5*x^6+4*y*x^4+5*x^5-2*y^4+y^3*x-4*y^2*x^2-2*y*x^3+2*x^4
     -y^3-3*x^3-3*y^2-3*y*x
sp[6]=T(4)*y^4+4*T(4)*y^3-T(4)*y^2*x+5*T(4)*y*x^2+2*T(4)*x^3+4*T(4)*y^2
     +2*T(4)*y*x-2*y*x^6+2*x^7+5*y^2*x^4+5*y*x^5-5*x^6+3*y^3*x^2+3*y^2*x^3
     -2*y*x^4-3*x^5-3*y^4-4*y^3*x-4*y^2*x^2-y*x^3+x^4+5*y^3+4*x^3+4*y^2+4*y*x
sp[7]=T(3)*x^2+T(3)*y-4*T(4)*y*x^2+4*T(4)*x^3+5*T(4)*y^2+T(4)*y*x+5*T(4)*x^2
     +5*T(4)*y-4*y^2*x+5*y*x^2+4*x^3-4*y^2-y*x-3*x^2-3*y
sp[8]=T(3)*y*x+2*T(4)*y*x^2-2*T(4)*x^3-3*T(4)*y^2+3*T(4)*y*x-y^3+2*y^2*x
     -2*y*x^2-5*x^3+2*y^2+3*y*x
sp[9]=T(3)*y^2+5*T(4)*y^3-3*T(4)*y^2*x-3*T(4)*y*x^2+T(4)*x^3-4*T(4)*y^2
     +T(4)*y*x-4*x^4-5*y^3-4*y^2*x+y*x^2+2*x^3-y^2+2*y*x
sp[10]=T(2)*x+5*T(3)*y+2*T(3)*x-T(4)*y*x^2+T(4)*x^3+3*T(4)*y^2-T(4)*y*x
      -2*T(4)*x^2-T(4)*y-4*T(4)*x+2*y^2*x-3*y*x^2+3*x^3+y^2-3*y*x+3*x^2
      -5*y-2*x
sp[11]=T(2)*y-3*T(3)*y+5*T(4)*y*x^2-5*T(4)*x^3-4*T(4)*y^2+5*T(4)*y*x
```

```
      -T(4)*x^2+3*T(4)*y+y^2*x+4*y*x^2-4*x^3-4*y^2-2*y*x+3*x^2+5*y
sp[12]=T(1)*x+3*T(3)*y+5*T(3)*x-5*T(4)*y*x^2+5*T(4)*x^3+4*T(4)*y^2
      -5*T(4)*y*x+T(4)*x^2-T(4)*y-2*T(4)*x-y^2*x-4*y*x^2+4*x^3-y^2-y*x+3*y+x
sp[13]=T(1)*y-5*T(3)*y+T(4)*y*x^2-T(4)*x^3-3*T(4)*y^2+T(4)*y*x+2*T(4)*x^2
      +5*T(4)*y-2*y^2*x+3*y*x^2-3*x^3-3*y^2+4*y*x+3*x^2+y
sp[14]=T(4)^2+2*T(1)+3*T(2)-2*T(3)*y+2*T(3)*x+2*T(3)-4*T(4)*y*x^2
      +4*T(4)*x^3+T(4)*y^2+2*T(4)*y*x+2*T(4)*x^2-T(4)*y+2*T(4)-3*y^2*x
      -y*x^2+x^3-2*y^2+4*y*x+3*x^2+5*y+x+1
sp[15]=T(3)*T(4)+T(1)-4*T(2)-2*T(3)*y+5*T(3)*x+3*T(3)+3*T(4)*y^2*x
      +2*T(4)*y*x^2-5*T(4)*x^3+5*T(4)*y^2+2*T(4)*y*x+4*T(4)*x^2+3*T(4)*y
      -4*T(4)*x-3*T(4)+5*y^3-y^2*x-5*y*x^2+2*x^3+5*y^2-2*y*x+4*x^2+2*y+3*x
sp[16]=T(2)*T(4)+5*T(1)-2*T(2)+4*T(3)*y-2*T(3)*x-3*T(3)-3*T(4)*y*x^2
      +3*T(4)*x^3-2*T(4)*y^2+5*T(4)*y*x-3*T(4)*x^2+2*T(4)*y-5*T(4)*x+3*T(4)
      -5*y^2*x+2*y*x^2-2*x^3-5*y^2+5*x^2+y-4*x+3
sp[17]=T(1)*T(4)-2*T(1)-4*T(2)-2*T(3)*y+3*T(3)*x+T(3)-4*T(4)*y*x^2+4*T(4)*x^3
      +T(4)*y^2+2*T(4)*y*x-3*T(4)*x^2+2*T(4)*y-4*T(4)*x-T(4)-3*y^2*x-y*x^2
      +x^3-3*y^2-2*y*x+x^2-5*y-4*x-5
sp[18]=T(3)^2-5*T(1)-2*T(2)-4*T(3)*y-4*T(3)*x+2*T(3)+3*T(4)*y^2*x
      -4*T(4)*y*x^2+3*T(4)*x^3-5*T(4)*y^2-2*T(4)*y*x+3*T(4)*x^2+3*T(4)*y
      +5*T(4)-x^5-3*y*x^3-x^4+5*y^3-4*y^2*x+2*y*x^2+x^3+5*y^2+5*y*x+2*x^2
      -y+4*x-5
sp[19]=T(2)*T(3)-3*T(1)-4*T(2)-5*T(3)*y-5*T(3)*x+3*T(3)+4*T(4)*y^2*x
      -2*T(4)*y*x^2-2*T(4)*x^3-5*T(4)*y^2+3*T(4)*y*x+2*T(4)*x^2-2*T(4)*y
      +5*T(4)*x+2*T(4)+3*y^3-3*y^2*x+5*y*x^2+2*x^3+4*y^2-4*y*x+x^2-3*y+3*x-1
sp[20]=T(1)*T(3)-4*T(1)-2*T(2)+3*T(3)*y-4*T(3)*x-T(3)+3*T(4)*y^2*x+T(4)*y*x^2
      -4*T(4)*x^3-3*T(4)*y^2+2*T(4)*y*x+T(4)*x^2-2*T(4)*y+3*T(4)*x+3*T(4)
      +5*y^3+y^2*x+3*y*x^2+5*x^3+4*y^2-2*y*x+3*x^2+y+2*x+2
sp[21]=T(2)^2+5*T(2)+3*T(3)*y+2*T(3)*x-2*T(3)-5*T(4)*y*x^2+5*T(4)*x^3
      +4*T(4)*y^2+2*T(4)*y*x+5*T(4)*x^2-3*T(4)*x-T(4)-y^2*x-4*y*x^2+4*x^3
      -y^2+2*y*x+3*x^2+3*y-4*x-4
sp[22]=T(1)*T(2)-5*T(1)-4*T(2)+3*T(3)*y-4*T(3)-5*T(4)*y*x^2+5*T(4)*x^3
      +4*T(4)*y^2+3*T(4)*y*x+4*T(4)*x^2+T(4)*y-4*T(4)*x-3*T(4)-y^2*x
      -4*y*x^2+4*x^3-y^2+4*y*x+5*x^2-5*y-4*x+1
sp[23]=T(1)^2+3*T(3)*y+4*T(3)*x-5*T(3)-5*T(4)*y*x^2+5*T(4)*x^3+4*T(4)*y^2
      +T(4)*y*x-5*T(4)*x^2-4*T(4)*y+T(4)*x+3*T(4)-y^2*x-4*y*x^2+4*x^3-y^2
      +x^2+5*y-5*x

ring r=11,(x,y),dp;
>  ideal i=(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2);
> list norp=normalP(i,"withRing");norp;
//   characteristic : 11
//   number of vars : 5
//        block   1 : ordering dp
//                  : names    T(2) T(3) T(5) T(6)
//        block   2 : ordering dp
//                  : names    x
```

```
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=xy4-2y5
      _[2]=2xy4-4y5
      _[3]=x4y2-5y6
      _[4]=y7-4y5
      _[5]=x6y-3y5
      _[6]=x7+2x5y+2y6+x3y2-3y5
      _[7]=x2y3-2xy4
> def Rp=norp[1][1];
> setring Rp;
> normap;
normap[1]=x
normap[2]=-5*T(2)*x
> option(redSB);
> ideal sp=std(norid);sp;
sp[1]=T(6)^2*x^2-4*T(6)^2*x+T(6)^2+3*T(2)*x^6-4*T(2)*x^5-5*T(2)*x^4
     -3*T(2)*x^3-3*T(2)*x^2-T(2)*x+T(2)+4*T(3)*x^7-T(3)*x^6+2*T(3)*x^5
     -5*T(3)*x^4-T(3)*x^3-4*T(3)*x^2+2*T(3)*x+4*T(3)+T(5)*x^7
     -5*T(5)*x^6+2*T(5)*x^5-2*T(5)*x^4+5*T(5)*x^3-2*T(5)*x^2+T(5)*x
     -4*T(5)-T(6)*x^7+3*T(6)*x^6+5*T(6)*x^5+2*T(6)*x^4-4*T(6)*x^3
     +5*T(6)*x^2+3*T(6)*x-T(6)-x^7+5*x^6-3*x^5+2*x^4-x^2-2*x-3
sp[2]=T(5)*T(6)-3*T(6)^2+2*T(2)*x^4-4*T(2)*x^3-2*T(2)*x^2+T(2)*x+5*T(2)
     -T(3)*x^5-4*T(3)*x^4+T(3)*x^3-2*T(3)*x^2-2*T(3)*x+T(3)-3*T(5)*x^5
     +5*T(5)*x^4+2*T(5)*x^3+5*T(5)*x^2+T(5)+3*T(6)*x^5+T(6)*x^4
     -3*T(6)*x^3+T(6)*x^2-5*T(6)*x+2*T(6)+3*x^5-3*x^4-4*x^3-4*x^2-x
sp[3]=T(3)*T(6)+4*T(6)^2*x-3*T(6)^2+T(2)*x^5-T(2)*x^4+3*T(2)*x^3
     +3*T(2)*x+T(2)+5*T(3)*x^6+5*T(3)*x^5-T(3)*x^4-4*T(3)*x^3
     -3*T(3)*x^2+5*T(3)*x-T(3)+4*T(5)*x^6-4*T(5)*x^5+4*T(5)*x^4
     -2*T(5)*x^2-T(5)*x-4*T(5)-4*T(6)*x^6-4*T(6)*x^5+3*T(6)*x^4
     +4*T(6)*x^3+4*T(6)*x^2+3*T(6)*x+4*T(6)-4*x^6+4*x^5-3*x^4-3*x^3
     +3*x^2+x+3
sp[4]=T(2)*T(6)-2*T(6)^2*x+5*T(2)*x^5-5*T(2)*x^4-3*T(2)*x^3-2*T(2)*x^2
     +2*T(2)*x-5*T(2)+3*T(3)*x^6+3*T(3)*x^5+T(3)*x^4-5*T(3)*x^3
     -4*T(3)*x^2+5*T(3)*x+3*T(3)-2*T(5)*x^6+2*T(5)*x^5+5*T(5)*x^4
     -2*T(5)*x^3-5*T(5)*x^2-4*T(5)+2*T(6)*x^6+2*T(6)*x^5-3*T(6)*x^4
     -3*T(6)*x^3-2*T(6)*x^2+2*x^6-2*x^5-4*x^4+3*x^3-2*x^2+3*x-4
sp[5]=T(5)^2-4*T(6)^2*x-4*T(6)^2-T(2)*x^5+4*T(2)*x^3-5*T(2)*x^2
     +5*T(2)*x-4*T(2)-5*T(3)*x^6+T(3)*x^5+5*T(3)*x^4-2*T(3)*x^3
     +4*T(3)*x^2+4*T(3)*x+4*T(3)-4*T(5)*x^6+5*T(5)*x^4-2*T(5)*x^3
     -T(5)*x^2-T(5)*x+2*T(5)+4*T(6)*x^6-3*T(6)*x^5-4*T(6)*x^4
     +3*T(6)*x^3+T(6)*x^2+4*T(6)+4*x^6-x^4-5*x^2-3*x+1
sp[6]=T(3)*T(5)+5*T(6)^2*x-5*T(6)^2+4*T(2)*x^5+3*T(2)*x^4-3*T(2)*x^3
     +3*T(2)*x^2-5*T(2)-2*T(3)*x^6-3*T(3)*x^4-3*T(3)*x^3+T(3)*x^2
     -T(3)*x+5*T(5)*x^6+T(5)*x^5-4*T(5)*x^4-5*T(5)*x^3+T(5)*x^2
```

```
     -4*T(5)*x+4*T(5)-5*T(6)*x^6-2*T(6)*x^4-T(6)*x^3-2*T(6)*x^2
     +3*T(6)*x+5*T(6)-5*x^6-x^5+5*x^4-3*x^3+x^2-3*x-3
sp[7]=T(2)*T(5)-4*T(6)^2*x+2*T(6)^2-T(2)*x^5-4*T(2)*x^4-3*T(2)*x^3
     +5*T(2)*x^2-4*T(2)*x-T(2)-5*T(3)*x^6+3*T(3)*x^5-4*T(3)*x^4
     +5*T(3)*x^3-4*T(3)*x^2-T(3)*x-3*T(3)-4*T(5)*x^6-5*T(5)*x^5
     -T(5)*x^4+2*T(5)*x^3-4*T(5)*x^2-4*T(5)*x+3*T(5)+4*T(6)*x^6
     +2*T(6)*x^5+T(6)*x^4+4*T(6)*x^3+2*T(6)*x^2+2*T(6)*x-2*T(6)
     +4*x^6+5*x^5+5*x^4+x^3+3*x^2+4*x+5
sp[8]=T(3)^2-3*T(6)^2*x+3*T(6)^2+2*T(2)*x^5+T(2)*x^4+5*T(2)*x^3
     +2*T(2)*x^2+2*T(2)*x+3*T(2)-T(3)*x^6+3*T(3)*x^5+5*T(3)*x^4
     -3*T(3)*x^3+3*T(3)*x^2-3*T(3)*x-3*T(3)-3*T(5)*x^6+4*T(5)*x^5
     +5*T(5)*x^4-4*T(5)*x^3+4*T(5)*x^2+5*T(5)*x-4*T(5)+3*T(6)*x^6
     +2*T(6)*x^5-4*T(6)*x^4+4*T(6)*x^3+T(6)*x^2-2*T(6)*x-3*T(6)
     +3*x^6-4*x^5-5*x^4+5*x^3-4*x^2
sp[9]=T(2)*T(3)+2*T(6)^2*x+2*T(6)^2-5*T(2)*x^5+T(2)*x^3-4*T(2)*x^2
     -4*T(2)*x+2*T(2)-3*T(3)*x^6+5*T(3)*x^5+2*T(3)*x^4+4*T(3)*x^3
     +3*T(3)*x^2-2*T(3)*x-3*T(3)+2*T(5)*x^6+5*T(5)*x^3+3*T(5)*x^2
     +5*T(5)*x+3*T(5)-2*T(6)*x^6-4*T(6)*x^5+5*T(6)*x^4-5*T(6)*x^3
     +T(6)*x^2-4*T(6)*x-2*T(6)-2*x^6-5*x^4-3*x^3-2*x^2-4*x+5
sp[10]=T(2)^2-2*T(6)^2*x-4*T(6)^2+5*T(2)*x^5+5*T(2)*x^4-3*T(2)*x^3
      -2*T(2)*x^2+2*T(2)*x+3*T(2)+3*T(3)*x^6-2*T(3)*x^5+4*T(3)*x^3
      +5*T(3)*x^2-5*T(3)*x-3*T(3)-2*T(5)*x^6-2*T(5)*x^5-T(5)*x^4
      +5*T(5)*x^3-3*T(5)*x^2-5*T(5)+2*T(6)*x^6-5*T(6)*x^5+3*T(6)*x^3
      +T(6)*x^2-4*T(6)*x+4*T(6)+2*x^6+2*x^5+3*x^4-4*x^3-5*x^2-3*x+2
sp[11]=T(6)^3-5*T(6)^2*x+5*T(6)^2-4*T(2)*x^7-3*T(2)*x^6-4*T(2)*x^5
      +3*T(2)*x^4-5*T(2)*x^3+4*T(2)*x^2+4*T(2)*x+5*T(2)+5*T(3)*x^8
      -T(3)*x^7+T(3)*x^6+5*T(3)*x^5-5*T(3)*x^3-4*T(3)*x^2+4*T(3)*x
      -4*T(3)+4*T(5)*x^8-5*T(5)*x^6-3*T(5)*x^5+5*T(5)*x^4+3*T(5)*x^3
      +4*T(5)*x^2-2*T(5)*x-4*T(6)*x^8+3*T(6)*x^7-3*T(6)*x^6
      +4*T(6)*x^5+5*T(6)*x^4+5*T(6)*x^3-5*T(6)*x^2+4*T(6)*x+5*T(6)
      -3*x^7+4*x^6+5*x^5-4*x^4-2*x^3+3*x-4
```

4 Magma

`IntegralClosure` gets a similar result if y is treated as the independent variable:

```
> FF<y>:=FunctionField(Q);
> P<x>:=PolynomialRing(FF);
> f:=(x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2);
> Ff<X>:=RationalExtensionRepresentation(FunctionField(f));
> C<Y>:=CoefficientRing(Ff);
> INT:=Integers(C);
> IC:=IntegralClosure(INT,Ff);
> B:=Basis(IC);for i in [1..#B] do i, B[i]; end for;
1 1
2 X
3 1/Y*X^2
4 1/Y*X^3
5 1/Y^2*X^4
6 1/Y^3*X^5 + 2/Y^2*X^3 + 1/Y*X
7 1/Y^3*X^6 + 3/Y^2*X^4 + 3/Y*X^2
8 1/Y^4*X^7 + 3/Y^3*X^5 + 3/Y^2*X^3 - 1/Y*X^2 + 1/Y*X
```

5 Macaulay2

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone
i1 : R=QQ[x,y];
i2 : I=ideal((x-y)*x*(y+x^2)^3-y^3*(x^3+x*y-y^2));
i3 : A=R/I;
i5 : toString icf

o5 = {(x^5+2*x^3*y+x*y^2)/(y^3), (x^2)/(y), x, y}

i6 : G=gens gb presentation integralClosure A

o6 = | x8-x7y+3x6y-3x5y2+3x4y2-4x3y3+x2y3-2xy4+y5 w_(5,0)y-x2
       w_(5,0)x6-x7+3x6-3x5y+3x4y-4x3y2+x2y2-2xy3+y4
       w_(5,0)^2x4-w_(5,0)x5+3w_(5,0)x4-3x5+3x4-4x3y+x2y-2xy2+y3
       w_(5,0)^3x2-w_(5,0)^2x3+3w_(5,0)^2x2-3w_(5,0)x3+3w_(5,0)x2-4x3+x2-2xy+y2
       w_(5,0)^4-w_(5,0)^3x+3w_(5,0)^3-3w_(5,0)^2x+3w_(5,0)^2-4w_(5,0)x+w_(5,0)-2x+y
       w_(6,0)y-w_(5,0)^2x-2w_(5,0)x-x w_(6,0)x-w_(5,0)^3-2w_(5,0)^2-w_(5,0)
       w_(6,0)w_(5,0)^2+w_(6,0)w_(5,0)-w_(5,0)^3x-3w_(5,0)^2x-w_(5,0)^2-4w_(5,0)x-w_(5,0
       w_(6,0)^2-w_(6,0)w_(5,0)-2w_(6,0)-w_(5,0)^3x-3w_(5,0)^2x-w_(5,0)^2-4w_(5,0)x-x+y+
```

6 Type I using my qth-power algorithm in Magma

My qth power algorithm gives a non-minimized ideal of relations:

```
    f_2^2 + 30*f_4 + 30*f_3 + 5,
    f_3^2 + 30*f_6 + 3*f_5 + 30*f_4 + 19*f_3 + 19*f_2 + 11,
    f_3*f_2 + 30*f_5 + f_4 + 3*f_3 + 17,
    f_4^2 + 30*f_8 + 2*f_5 + 28*f_2,
    f_4*f_3 + 30*f_7 + 3*f_6 + 6*f_5 + 29*f_4 + 19*f_3 + 13*f_2 + 30,
    f_4*f_2 + 30*f_6 + 29*f_5 + 4*f_3 + 6*f_2 + 21,
    f_5^2 + 30*f_3*f_7 + 29*f_8 + 4*f_6 + 21*f_5 + 6*f_4 + 11*f_3 + 14*f_2 + 29,
    f_5*f_4 + 30*f_2*f_7 + 2*f_8 + 29*f_6 + 23*f_5 + 30*f_4 + 8*f_3 + 18*f_2 +11,
    f_5*f_3 + 30*f_8 + 3*f_7 + 28*f_6 + 16*f_5 + 27*f_4 + 18*f_3 + 2*f_2 + 4,
    f_5*f_2 + 30*f_7 + f_6 + 4*f_5 + 25*f_3 + 21*f_2 + 20,
    f_6^2 + 30*f_5*f_7 + 3*f_4*f_7 + 26*f_3*f_7 + 16*f_2*f_7 + 8*f_8 + f_7 +
        29*f_6 + 12*f_5 + 15*f_4 + 7*f_2 + 12,
    f_6*f_5 + 30*f_4*f_7 + 3*f_3*f_7 + 4*f_2*f_7 + 25*f_8 + 30*f_7 + 24*f_6 +
        20*f_5 + 13*f_4 + 17*f_3 + 15*f_2 + 24,
    f_6*f_4 + 30*f_3*f_7 + 29*f_8 + f_7 + 9*f_6 + 22*f_5 + 2*f_4 + 26*f_3 +
        2*f_2 + 28,
    f_6*f_3 + 30*f_2*f_7 + 5*f_8 + 28*f_7 + 22*f_6 + 8*f_5 + f_4 + 19*f_3 +
        23*f_2 + 28,
    f_6*f_2 + 30*f_8 + f_7 + f_6 + 27*f_5 + f_4 + 6*f_3 + 9*f_2 + 26,
    f_8^2 + 30*f_2*f_7^2 + 2*f_8*f_7 + 28*f_6*f_7 + 27*f_5*f_7 + 30*f_4*f_7 +
        6*f_3*f_7 + 2*f_2*f_7 + 5*f_8 + 23*f_7 + 2*f_6 + 6*f_5 + 5*f_4 + 22*f_3
        + 10*f_2 + 25,
    f_8*f_6 + 30*f_7^2 + 3*f_6*f_7 + 4*f_5*f_7 + 28*f_4*f_7 + 3*f_3*f_7 +
        14*f_2*f_7 + 2*f_8 + 11*f_7 + 23*f_6 + 17*f_5 + 12*f_4 + 11*f_3 + 13*f_2 + 29,
    f_8*f_5 + 30*f_6*f_7 + 27*f_2*f_7 + 3*f_8 + 3*f_7 + 3*f_6 + 12*f_5 + 6*f_4 + 17*f_3 +
    f_8*f_4 + 30*f_5*f_7 + f_2*f_7 + 2*f_8 + 30*f_6 + 25*f_5 + 30*f_4 + 4*f_3 +
        12*f_2 + 21,
    f_8*f_3 + 30*f_4*f_7 + 3*f_3*f_7 + 6*f_2*f_7 + 21*f_8 + 22*f_7 + 28*f_6 +
        26*f_5 + 26*f_4 + 15*f_3 + 24*f_2 + 18,
    f_8*f_2 + 30*f_3*f_7 + 29*f_2*f_7 + 2*f_8 + 3*f_7 + f_6 + 2*f_5 + f_4 +
        26*f_3 + 23*f_2 + 15
]
time1= 10.100
```

Minimizing first produces

```
 f_15 - f_3*f_2^6 - 12*f_3*f_2^5 - 2*f_2^6 - 51*f_3*f_2^4 + 13*f_2^5 -
       104*f_3*f_2^3 + 129*f_2^4 - 108*f_3*f_2^2 + 351*f_2^3 - 52*f_3*f_2 +
       425*f_2^2 - 8*f_3 + 224*f_2 + 36,
 f_14 - f_2^7 + 2*f_3*f_2^5 - 6*f_2^6 + 15*f_3*f_2^4 - 20*f_2^5 +
       44*f_3*f_2^3 - 68*f_2^4 + 63*f_3*f_2^2 - 179*f_2^3 + 44*f_3*f_2 -
       266*f_2^2 + 12*f_3 - 196*f_2 - 56,
 f_13 - f_3*f_2^5 + f_2^6 - 13*f_3*f_2^4 - f_2^5 - 40*f_3*f_2^3 + 29*f_2^4 -
```

```
        52*f_3*f_2^2 + 131*f_2^3 - 40*f_3*f_2 + 176*f_2^2 - 16*f_3 + 140*f_2 +
        64,
 f_12 - f_2^6 + 3*f_3*f_2^4 - 3*f_2^5 + 16*f_3*f_2^3 - 13*f_2^4 +
        34*f_3*f_2^2 - 59*f_2^3 + 35*f_3*f_2 - 132*f_2^2 + 14*f_3 - 148*f_2 -
        64,
 f_11 - f_3*f_2^4-f_2^5-7*f_3*f_2^3-3*f_2^4-18*f_3*f_2^2+11*f_2^3-20*f_3*f_2+57*f_2^2-8*
    f_10 - f_2^5 + 4*f_3*f_2^3 + f_2^4 + 11*f_3*f_2^2 - 13*f_2^3 + 5*f_3*f_2 -
        59*f_2^2 - 2*f_3 - 40*f_2 + 4,
    f_9 - f_3*f_2^3 - 2*f_2^4 - 3*f_3*f_2^2 - 2*f_2^3 - 2*f_3*f_2 + 8*f_2^2 +
        8*f_2,
    f_8 - f_2^4 + 2*f_3*f_2^2 - f_2^3 + 4*f_3*f_2 - 8*f_2^2 + f_3 - 20*f_2 - 5,
    f_7 - f_3*f_2^2 - 2*f_2^3 - 3*f_3*f_2 - 2*f_2^2 - 2*f_3 + 8*f_2 + 8,
    f_3^2 - f_2^3 + 6*f_3*f_2 + 4*f_2^2 - 5*f_3 - 23*f_2 + 2,
    f_6 - f_2^3 + 3*f_3*f_2 + 2*f_2^2 - 11*f_2 - 8,
    f_5 - f_3*f_2 - f_2^2 - 2*f_3 + 9,
    f_4 - f_2^2 + f_3 - 5
```

from which

```
    f_3^2 - f_2^3 + 6*f_3*f_2 + 4*f_2^2 - 5*f_3 - 23*f_2 + 2
```

can be read off. After all a genus computation gives $g = 1$.

7 type I using my qth-power algorithm in Macaulay

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : loadPackage "QthPower";

i2 : wtr=matrix{{11,7}};

i3 : R=QQ[w,z,Weights=>entries weightGrevlex(wtr)];

i4 : I={w^7+3*w^6*z+w^6+3*w^5*z^3+6*w^5*z^2+9*w^4*z^4+4*w^3*z^6-w^3*z^5-3*w^2*z^7-3*w*z^

i5 : time ic=rationalIntegralClosure(wtr,R,I);toString ic
 19
z
154
     12
(3, z  )
     20
(5, z  )
      19
(13, z  )
      19
(17, z  )
      19
(19, z  )
      19
(23, z  )
      19
(29, z  )
     -- used 21.6351 seconds

o6 = ({
     z^12,
     w^4*z^6+2*w^2*z^9-3*w^5*z^4-4*w^3*z^7-w*z^10+6*w^6*z^2+6*w^4*z^5+w^2*z^8
         -10*z^11+22*w^5*z^3+39*w^3*z^6-15*w*z^9-5*w^6*z+31*w^4*z^4-6*w^2*z^7
         +6*z^10-10*w^5*z^2-24*w^3*z^5+11*w*z^8+w^6-26*w^4*z^3+5*w^2*z^6-z^9
         -2*w^5*z+4*w^3*z^4-2*w*z^7+5*w^4*z^2-w^2*z^5+w^5,
     w^6*z^3-2*w^2*z^9+12*w^5*z^4+21*w^3*z^7-18*w^6*z^2-12*w^4*z^5-4*w^2*z^8
        +30*z^11-65*w^5*z^3-117*w^3*z^6+45*w*z^9+15*w^6*z-93*w^4*z^4
        +18*w^2*z^7-18*z^10+30*w^5*z^2+72*w^3*z^5-33*w*z^8-3*w^6+78*w^4*z^3
        -15*w^2*z^6+3*z^9+6*w^5*z-12*w^3*z^4+6*w*z^7-15*w^4*z^2+3*w^2*z^5-3*w^5,
     w*z^11-w^2*z^9+w^3*z^7-w^4*z^5+w^5*z^3-w^6*z+z^10-3*w^5*z^2-4*w^3*z^5
        +2*w*z^8-5*w^4*z^3+w^2*z^6-w^5*z,
```

```
      w^3*z^8+4*w^2*z^9-3*w^5*z^4-8*w^3*z^7-2*w*z^10+8*w^6*z^2+12*w^4*z^5
         +2*w^2*z^8-15*z^11+29*w^5*z^3+58*w^3*z^6-21*w*z^9-7*w^6*z+41*w^4*z^4
         -8*w^2*z^7+9*z^10-15*w^5*z^2-36*w^3*z^5+16*w*z^8+2*w^6-37*w^4*z^3
         +7*w^2*z^6-2*z^9-w^5*z+8*w^3*z^4-4*w*z^7+10*w^4*z^2-2*w^2*z^5+2*w^5,
      w^5*z^5-5*w^2*z^9+18*w^5*z^4+38*w^3*z^7+8*w*z^10-42*w^6*z^2-49*w^4*z^5
         -9*w^2*z^8+75*z^11-157*w^5*z^3-291*w^3*z^6+108*w*z^9+40*w^6*z
         -216*w^4*z^4+42*w^2*z^7-49*z^10+87*w^5*z^2+196*w^3*z^5-89*w*z^8
         -9*w^6+209*w^4*z^3-40*w^2*z^6+9*z^9+13*w^5*z-36*w^3*z^4+18*w*z^7
         -45*w^4*z^2+9*w^2*z^5-9*w^5,
      w^2*z^10+2*w^2*z^9-w^5*z^4-4*w^3*z^7-w*z^10+3*w^6*z^2+6*w^4*z^5+w^2*z^8
         -6*z^11+10*w^5*z^3+23*w^3*z^6-8*w*z^9-2*w^6*z+15*w^4*z^4-3*w^2*z^7
         +3*z^10-4*w^5*z^2-12*w^3*z^5+5*w*z^8+w^6-11*w^4*z^3+2*w^2*z^6-z^9
         +w^5*z+4*w^3*z^4-2*w*z^7+5*w^4*z^2-w^2*z^5+w^5
},{
      p_0^2+2*p_0*p_6+5*p_0-3*p_1*p_6+2*p_1-4*p_2*p_6+6*p_2-p_3*p_6+5*p_3
         +6*p_4*p_6-9*p_4-p_5*p_6^2+2*p_5*p_6-21*p_5-8*p_6+25,
      p_0*p_1-29*p_0+3*p_1*p_6-8*p_1+4*p_2*p_6-14*p_2-3*p_3*p_6-19*p_3
         +3*p_4*p_6+42*p_4+14*p_5*p_6+75*p_5-p_6^2-20*p_6-126,
      p_0*p_2+3*p_0-p_1*p_6+3*p_1+12*p_2+6*p_3-14*p_4-4*p_5*p_6-33*p_5
         +3*p_6+40,
      p_0*p_3+2*p_0-p_1-p_2*p_6-6*p_2-p_3+4*p_4+p_5*p_6+12*p_5-10,
      p_0*p_4-10*p_0-3*p_1-5*p_2-p_3*p_6-5*p_3+3*p_4*p_6+15*p_4
         +6*p_5*p_6+24*p_5-9*p_6-44,
      p_0*p_5+2*p_0+p_1+2*p_2+p_3-p_4*p_6-5*p_4-2*p_5*p_6-8*p_5+3*p_6+15,
      p_1^2+39*p_0-33*p_1-p_2*p_6+12*p_2+3*p_3*p_6+15*p_3-5*p_4*p_6
         -15*p_5*p_6-117*p_5+p_6+198, p_1*p_2-6*p_0-7*p_1-42*p_2-p_3*p_6
         -18*p_3+3*p_4*p_6+48*p_4+4*p_5*p_6+108*p_5-p_6-162,
      p_1*p_3-2*p_0+9*p_1+22*p_2+2*p_3-p_4*p_6-36*p_4-60*p_5+p_6+90,
      p_1*p_4+5*p_0-9*p_1+8*p_2+p_3-12*p_4-p_5*p_6-39*p_5-3*p_6+90,
      p_1*p_5-p_0+p_1-4*p_2+p_3+6*p_4+9*p_5+p_6-36,
      p_2^2-2*p_0+4*p_1+21*p_2+6*p_3-p_4*p_6-20*p_4-48*p_5+60,
      p_2*p_3+2*p_0-2*p_1-8*p_2-p_3+8*p_4-p_5*p_6+18*p_5-20,
      p_2*p_4-p_0-3*p_1-15*p_2-4*p_3+18*p_4+33*p_5+3*p_6-58,
      p_2*p_5+p_1+4*p_2-6*p_4-10*p_5-p_6+20, p_3^2-p_0+2*p_2-3*p_5,
      p_3*p_4+3*p_1+6*p_2-2*p_3-12*p_4-18*p_5-p_6+30,
      p_3*p_5-p_1-2*p_2+4*p_4+6*p_5-10,
      p_4^2-p_1+3*p_2-p_3-12*p_4-12*p_5+42,
      p_4*p_5-p_2+p_3+3*p_4-14,
      p_5^2-p_3-p_4+5},
       QQ[p_0, p_1, p_2, p_3, p_4, p_5, p_6],
matrix {{8,   6,   5,   4,   3,   2,   7}})

i7 : wtRbar=matrix{{8,7,6,5,4,3,2}};

i8 : Rbar=QQ[f8,f7,f6,f5,f4,f3,f2,Weights=>entries weightGrevlex(wtRbar)];
```

```
i9 : phi=map(Rbar,ring(ic#1#1),matrix{{f8,f6,f5,f4,f3,f2,f7}});

i10 : Ibar=transpose gens gb ideal apply(ic#1,v->phi(v))

o10 = {-2} | f4-f2^2+f3-5                                  |
      {-2} | f5-f3f2-f2^2-2f3+9                            |
      {-3} | f6-f2^3+3f3f2+2f2^2-11f2-8                    |
      {-3} | f3^2-f2^3+6f3f2+4f2^2-5f3-23f2+2              |
      {-3} | f7-f3f2^2-2f2^3-3f3f2-2f2^2-2f3+8f2+8         |
      {-4} | f8-f2^4+2f3f2^2-f2^3+4f3f2-8f2^2+f3-20f2-5    |

i11 : for i to numRows(Ibar)-1 list
        if (degree(leadMonomial(Ibar_(i,0))))#0 !=1
        then Ibar_(i,0)
        else continue

         2     3                2
o11 = {f3  - f2  + 6f3*f2 + 4f2  - 5f3 - 23f2 + 2}
```

I used a manual minimization here instead of the built-in, first because it works better, and second because I can use names reflecting the weights involved. It should be possible to automate this step; but I don't know, among other things, how to produce the correct variable names from the weights and manipulate the weights and variables when writing Macaulay2 code.

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : R=QQ[w,z,MonomialOrder=>{Weights=>{11,7},Weights=>{1,0}}];

i2 : I=ideal(w^7+3*w^6*z+w^6+3*w^5*z^3+6*w^5*z^2+9*w^4*z^4+4*w^3*z^6-w^3*z^5-3*w^2*z^7-3

i3 : S=R/I

i4 : time ic2=icFractions S;toString ic2
     -- used 1154.54 seconds

o5 = {(480283443759700624527742127489090366563851437985675334193034000715938555
     48717601230806796393203985859360245113996959727780809488310561976408247148
     47127877525494404304786089974990003055877492135791668855486642460944761735
     46761479267214575904128000000*w^4*z^4+
     54333603945328183472215340166201710186659296010071783960504256696377459148
     95950170008194504790153477217882088280886828691511063139133318972132985103
     98158862627725943402980229991176755962346033413149820266847859940725353253
     413242699218130484362240000*w^2*z^7-
     20292591246852171515302909684013412974972164090071687581622600676403167993
     12227264310857412736449433309072099970363852293261996790721641659628452596
     17879912474607164446833822071500693217305936957884921540279216593455856094
     9624527016708149166210560000*w^5*z^2+
     10287987028874213787503193043349398009633400283793996600008597747472284284
     91779019156190828999091459217819386065393004299326022210462177342887754754
     02142760591987690442019642215338777621984865424407395823014889500174706891
     0599795098278487682520320000*w^3*z^5-
     68937364085802892004502418288061172716621976218772196798795580517985745079
     79714906227788221259381060951512268111706644316136513575976041120514437103
     18192612785459414580983431906382999014031032500008777886341325098671389908
     8765372662275847765509120000*w*z^8+
     27070385682223201889654607958579230527777633087204196308819589407334293011
     45141379136177754033532543324439839957092062262274342121170333440558743796
     33495572894578502899639484624140965685022218067483806239396077405078135482
     68648947109861390229395200000*w^4*z^3-
     23892489174331394229982523366624897001961756760730589902086610265426536893
     02812545076622660286039443996351079492989251858106276733687941490551921566
     14492732743782138926852478375332343828530538514685791320941512863898953253
     04352200225699981244460800000*w^2*z^6+
     14342211443602862652071199664870937571828347072767497553569922006606755951
     51497826659438192680188201377908957735291272325299894965559924819028293741
     56092375182618177780862369926447733134572938232777822705895665076207529539
     84421349113048212084499200000*z^9+
     29609001461690546700646309106143240013647810188804941111947339508064516476
```

```
17136083902131946394679682588825139658652291058699171025254955637508026432
11421834840400389267352347404748983467467128899518493587631153088168257168
0914466136161212167839488000*w^5*z-
87648093813597087424399469576716431585013840876066327222558521266406169054
68840894318727941386624979024217947280599236705102304521984172980206400024
40836794228854579569053306165413707863413984945335402812338454388583251306
3683279393462693312615488000*w^3*z^4+
49967813076740851268505479936653056494970357611453157357052965255476654072
44964470865741570236329561913422774053646881771560434905685897705684802376
56015871454057474973172609981309065233136339755618674778300066504804214376
3509646726183824284653760000*w*z^7-
17326479113352625419075961159200924786234918263688780517861451364745030790
27675280229562947444551456103111636318078728548954908187914562965513691116
23642445708628755290678337117915032354704104177870759692247290095713133871
74610453942406917968545344000*w^4*z^2+
66194001233917719606128402850005681513830215266129571177772225140223600066
71728251085455579444175031434153007736160806898351883585202450272802511658
09751522211551609332073587983527600618954974204767762950890256417293729055
5754239745454107632888960000*w^2*z^5-
69863686787928370443478730615327986329656574428932233209578492231660879258
63809605198836906897977870006000508491797652488977212210987363229217229552
86269433879737973630402798724280892321475309924005975678250318383808228229
638826199303424297006528000*z^8-
10767366807437335860573356797157909901742103628272907247877383706456201253
21898380264572730930465297821813242867735057511497029896814162847024493233
68115354941126528564873113419499754016642973811696307091191585586821742429
03747181809909931569866752000*w^5+
69811774568467497865391420965261320471252980904079063857458534124223744227
05378757834530893593082774414974201825877627281795469677475420994083171586
78056094648848931956788955813374836882921497822918923067093100450655176245
9408245525644418365866752000*w^3*z^3-
64853742505786987708767477462620250336349762061961190253493759629812055716
98923173271459063749507382769891510330708215057661961341126770615457843014
25580817608268428617140288846810617589918087943060512764955130247287279220
866227935535874437868960000*w*z^6+
13544321707120116177766662323706594426657745447934689675301287088298143309
10330481315543992874559506710338883760284115744287281195574306968794927391
01344659341104734240872460584210944857117282218435210753384762673483723102
91347590052173338631913920000*w^4*z+
23442681481093350682328859706500979790129122035420252994196213462447463638
13059991530912961458845961057574462188041913681506073923189267867062172585
67074270273115755132527434116200592110707452052893063272281080822743103723
7218097110272458236816608000*w^2*z^4+
22000297466755823525729664805943856706218506190542153853059774095457241771
45107365021404826775857350360327236310275012680241952191688205273288498397
98567326307300168564993310127977553199976447846657408266142971876625912256
```

```
  24953548973455951896266880000*z^7-
  12351335926805508246713676795063308860803389135822870127185814035810462553
  34621650623651292747119272519272221491276734424377261211470301572634324569
  78509299006053195380247689043971736826042461063155443166417635552669348550
  008521090059263814138066880000*w^3*z^2+
  69623109186388416837063192284349288094327026844177222720037774373331066339
  85579664347480372237560054518656464926082379650205537691497968233019959950
  81483495976158730066149806583096412595126364359075071931048118800551435913
  13986365122508829702650240000*w*z^5-
  13019630629969949786123198964688840195298236508555116511224557913812958166
  91859210257060964534144465340340190223863101067168349044594647549293526970
  73410936493641953793576149827267918793396906408028377906768604324456097318
  767574379726268584923745600000*w^4-
  14390375492544622154897727869698122519948039340167139553903414953400019708
  71046272180462908466856054841712104509174490453039355858004730290346138528
  57648974107554284060369572412618885133983601659669938159568121719967735828
  53223548683931998157774720000*w^2*z^3+
  14448383513421088855277409752082576510636995050453768096277243052953288630
  50808332832129719151491440218652384146869627965341589086063373003024178982
  13294135615535577968667320286053329696197737646142016534211685584612437497
  61283241179756369688768000000*z^6+
  14799505900503039381931424435723336751899131343487675384831363659224326356
  20208897996296765880918480019811053088148060136524568787560101897233280573
  18091826464351970220871921910703405366442626404406919668208209477356019346
  512649981662523990670547520000*w^3*z+
  96187887112135006694103330413117556178591118559727478993413773311429836637
  51298220307006860018330468217886946712213147530171327371605185619344765770
  73246316100735936077253999030691224266196357345895720386180755924450987282
  812785622905181311670761600000*w*z^4+
  93102148749263672483140997057593866081110265890633827675409296390109814079
  45364713989975931574573818793388534487164724219258204048098523426846960184
  63802874265580761920954233392606224284761483549104540269752227447333565915
  058428348414808664758933120000*w^2*z^2+
  55807818907137625423347942894369556408974431645397275239423227556296104375
  29492340452129890508806883941028370892603513423919221775063227241933746601
  94124228568033656400885814693861694089858017262154748396636342732964828876
  611747435828604631611048640000*w^3+
  91176524161443781221228094240517099070374757265888503571557542285661727902
  82125984337154592705486733499805870158820787275288097927602915335584081466
  74992642801837264771275124665139537047855823415979195339142841881696786828
  685778934648525709209704640000*w*z^3+
  13025217737349111603032584891502442724339251037984071938793934612237389700
  40303712048164941815069533357115124308402969610755442561086130762226297352
  39284663257405323538753589237877076721122260487997027905591834554528112404
  0979684192355036727442435200000*w^2*z)
/(48028344375970062452774212748909036656385143798567533419303400071593855548
```

```
     71760123080679639320398585936024511399695972778080948831056197640824714847
     12787752549440430478608997499000305587749213579166885548664246094476173546
     761479267214575904128000*w^3*z+
     16871495332071534759051454222052661594678883847189107893447604640534251820
     95977274005264386120242580187782969389123969924607923050909484812289707523
     22184415639162407629665211941956517603901646821399649538889747987033937886
     939288870790761279142400000*w*z^4-
     45712438234451393805781826240438439466728342052705125855635772592931377933
     08532281538471332583426263694645050768821651247848682663461324195273993317
     59212656198468796645161568570442399990864681921625436905795410992086138785
     611489219211530999328000000*w^2*z^2-
     16981581079332349794327991770658676876576080614640663443248400886586032547
     30274675625322230407262532104324502027759707735361222156736212556560826784
     26559082162335970176863339424084900232452770577195991019766546542626595680
     075602865477151451766400000*z^5-
     33093982404052804006439685967795535056161972483857435307096672231904275700
     52127890130311288253811018843795781035528703248740618401073707218537836490
     53476863953801346118096700354326184944489255505293444603404788799931135297
     732472638768680974028800000*w^3-
     16411098852881742619071505663148288616734917244499481835186933531932672858
     25498653011709561070276647574004958262729494514634077892822982349036736271
     02942250615810702261111857516507714064688881910752807415324222919569205891
     250615386370038742039680000*w*z^3-
     10050467841707435115145165819774690101296123523821759739285543215197037439
     86764930147701408503172072456861366699559354895061021051295413189430824312
     02217566094745812276579604855843291018716628489099749813868584613515175656
     655659862201672816032960000*w^2*z-
     16856447432324603664490311377429672595743160892196062778990116895112170068
     92609721637484672343406680255094448171347899292898520600407268503528208812
     48843158218124840963445207000395551312230693920499019289913633182047843747
     214971147097157646368960000*z^4-
     24080639189035148092129016253470960851061658417422946827128738421588814384
     18013888053549531919152400364420640244782713275569315143438955005040298303
     55490226025892629947778867143422216160329562743570027557019475974354062496
     021387352995939494812800000*w*z^2), 0, 0, 0, z}

i6 : G=transpose gens gb presentation integralClosure S

o6 = {-1} | w          |
     {-3} | z3         |
     {-1} | w_(4,4)    |
     {-1} | w_(4,3)    |
     {-2} | w_(6,1)z   |
     {-2} | w_(6,1)^2  |
```

Well, the fraction is unreadable and the presentation is wrong.

8 Timings

The original example: (* means that the integral closure mod q is larger than in characteristic 0, — means the method is not applicable, ** means it ran out of (time and/or storage) resources, and **** here meant the this was not separable as given, though the variables could have been interchanged to make it separable.)

$char$	$normal$	$normalP$	$icFract.$	$icFRacP$	$Normalis.$	$Int.Clos.$	$QthPower$	Qth
0	0	$---$	1.5	$---$	0.2	0.4	$---$	$---$
2	0	0	0.9	0.2	0.2	0.3	$---$	$---$
3	1	0	0.8	0.3	0.2	0.3	$---$	$---$
5	0	35	1.1	1.4	0.1	0.4	$---$	$---$
7	0	0	1.2	2.3	0.2	0.4	$---$	$---$
$*11$	0	4	3.3	152	0.1	0.3	$---$	$---$
13	1	7	1.2	356	0.1	0.3	$---$	$---$
17	0	26	1.2		0.1	0.3	$---$	$---$
19	0	37	1.3		0.1	0.4	$---$	$---$
23	0	145	1.3		0.1	0.3	$---$	$---$
29	0	639	1.3		0.2	0.4	$---$	$---$
31	0		1.4		0.2	0.3	$---$	$---$
37	1		1.3		0.1	0.3	$---$	$---$
41	0		1.4		0.1	0.3	$---$	$---$
101	0		1.5		0.2	0.3	$---$	$---$
4001	0		1.7		0.1	0.3	$---$	$---$

9 Timings

The type I version: (* means that the integral closure mod q is larger than in characteristic 0, — means the method is not applicable, ** means it ran out of (time and/or storage) resources, and **** here meant the this was not separable as given, though the variables could have been interchanged to make it separable.)

char	*normal*	*normalP*	*icFract.*	*icFRacP*	*Normalis.*	*Int.Clos.*	*QthPower*	*Qth*
0	11	−−−		−−−		0.1	(23)	
2	1	0	20.	0.6	***	0.1	0.4	0.1
3	0	0	2.5	0.5		0.1	0.3	0.1
5	1	0	***	2.5		0.1	0.5	0.1
*7	1	1		58.		0.1	0.6	0.1
*11	2	3		***		0.1	1.9	0.2
13	2	30				0.1	1.4	0.2
17	1	835				0.1	3.0	0.2
19	1					0.1	3.2	0.3
23	1					0.1	4.4	0.4
29	2					0.1	6.1	0.7
31	1					0.1	7.1	0.9
37	2					0.1	13.	0.8
41	1					0.1	15.	0.9
101	1					0.1	87.	6.0
4001	2					0.1		