Leonard 2009 and Quillen-Suslin

1 Example

Consider this characteristic-independent version of the example from my 2009 paper:

$$A := \mathbf{F}[z, y_1, y_2; x_2, x_1]/I$$

for

$$I := \langle y_1^2 - y_2 x_2 x_1,$$
$$y_2 y_1 - y_1 - x_2^2 x_1 - x_2 x_1^2,$$
$$y_2^2 - y_2 - y_1(x_2 + x_1),$$
$$z^3 - (y_2 + y_1)(x_2 + 1)x_1 z - (y_2(x_2 x_1 + 1) + y_1(x_2 + x_1))x_2^2 x_1\rangle,$$

a degree 3 integral extension of the integral closure of a degree 3 integral extension of $P := \mathbf{F}[x_2, x_1]$. This is an example that was made up to test weighted extensions of weighted extensions, so there is a natural weight function defined by $wt(z) := (19, 12)$, $wt(y_1) := (15, 9)$, $wt(y_2) := (12, 9)$, $wt(x_2) := (9, 9)$, and $wt(x_1) := (9, 0)$ extending the grevlex weight on P. A has (standard) P-module basis

$$(1, y_2, y_1, z, zy_2, zy_1, z^2, z^2 y_2, z^2 y_1).$$

It was expected (by me) that the integral closure would have a similar P-module basis. One surprising result of my 2009 paper was that the integral closure found naturally (in the paper, only in characteristic 2) had a larger P-module generating set. Yet Eisenbud pointed out to me at the MSRI 2010 workshop that the *Quillen-Suslin theorem* was meant to guarantee a free presentation, one with exactly the same number (9) of P-module basis elements as the input.

Which is correct? Well, both are *technically* correct. It depends on what information one expects from an integral closure as to which should be used. I have no idea what advantages the free presentation has other than the word *free*. Perhaps this type of mathematics predates Gröbner bases and hasn't caught up; or perhaps there is a lot of mathematics related to freeness that I don't appreciate. But when a free presentation fails to provide even membership information, and when I seem to be the only one testing correcctness of implementations using basic membership to test isomorphism, ...

2 Canonical forms

This example has a slightly larger integral closure in characteristic 2 than in other characteristics, but the flavor is the same.

In characteristic 2 there should be 10 elements in a P-module generating set rather than the 9 that would form a P-module basis, and in other characteristics there should be 11. That is, despite the fact that there is the possibility of giving a P-algebra presentation (with $9 - 1 = 8$ non-trivial dependent variables, 2 independent variables, and 36 relations), it is better to give a strict affine P-algebra presentation that allows for membership to be decided directly by use of *canonical forms* and either the 11 (or 10) fractions c_i/δ (in canonical form) or the $55 + 2$ (or $45 + 1$) relations of the minimal, reduced Gröbner basis $\overline{B}$ for the ideal $\overline{I}$ of induced relations defining the strict affine P-algebra presentation $\overline{A} = \overline{R}/\overline{I}$.

That is, one should be able to rewrite a fraction $a/b \in Q(A)$ in its canonical form $c/\delta \in Q(A)$ for δ the canonical conductor element found. Then one should have the choice of checking membership either from the fractions by reducing c modulo the numerators to produce 0 to decide membership; or from the relations by mapping c into the presentation ring $\overline{R}$ and reducing that modulo $\overline{I}$ to a multiple of δ.

Neither of these methods is really possible for most examples and most implementations because there is no well-thought-out canonical form and no well-thought-out presentation. This example shows that such canonical forms can affect even the freeness of the presentation. So one must choose between a free presentation in which membership (and hence isomorphism) is hard to check, versus a presentation which is almost free, but allows direct methods for checking membership.

So I will be writing a paper discussing *canonical forms*, *membership* of $a/b \in Q(A)$ in the integral closure, and *isomorphism* between different supposed presentations of the same integral closure.

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";
> intmat A[5][5]=19,15,12,9,9,12,9,9,9,0,1,1,1,0,0,1,1,0,0,0,1,0,0,0,0;
> ring r=2,(z,y1,y2,x2,x1),M(A);
> poly f1 = y1^2-y2*x2*x1;
> poly f2 = y2*y1-y1-x2^2*x1-x2*x1^2;
> poly f3 = y2^2-y2-y1*(x2+x1);
> poly f4 = z^3-(y2+y1)*(x2+1)*x1*z-(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1;
> ideal i = f1,f2,f3,f4;
> list nor=normal(i);nor;
//   characteristic : 2
//   number of vars : 12
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4) T(5) T(6) T(7)
//        block   2 : ordering M
//                  : names    z y1 y2 x2 x1
//                  : weights  19 15 12  9  9
//                  : weights  12  9  9  9  0
//                  : weights   1  1  1  0  0
//                  : weights   1  1  0  0  0
//                  : weights   1  0  0  0  0
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=z*y1^2*y2+z*y1*x2*x1^2+z*y1^2*x2+z*y1^2
      _[2]=z*y1*y2*x2^2+z*y1*x2^3+z*y1^2*x2+z*y1*x2^2
      _[3]=z*y1^2*x2^2+z*y1*x2^3+z*y1^2*x2+z*y1*x2^2
      _[4]=z*y1^2*y2*x2+z*y1^2*x2*x1+z*y1*x2^3+z*y1*x2^2
      _[5]=z*y1^3*x2+z*y1^2*x2*x1+z*y1*x2^3+z*y1*x2^2
      _[6]=z*y1^4+z*y1^3*y2+z*y1^2*x2*x1+z*y1^3
      _[7]=y1^3*x2^3+y1^2*x2^4+z*y1^2*x2+y1^3*x2+z*y1*x2^2
          +z*y1^2+y1^2*x2^2+z*y1*y2+z*y1*x1+z*y1
      _[8]=z^2
> def R=nor[1][1];
> setring R;
> option(redSB);
> ideal s=std(norid);s;
s[1]=y2^2+y1*x2+y1*x1+y2
s[2]=y1*y2+x2^2*x1+x2*x1^2+y1
s[3]=y1^2+y2*x2*x1
```

```
s[4]=z^3+y2*x2^3*x1^2+z*y1*x2*x1+y1*x2^3*x1+y1*x2^2*x1^2+z*y2*x2*x1+z*y1*x1
    +z*y2*x1+y2*x2^2*x1
s[5]=T(7)*y2*x1+T(7)*y1+z*x2^3*x1+z*x2^2*x1^2+z^2*y2+z*y2*x2^2+z*y2*x2*x1
    +z^2*x2+z^2*x1+x2^3*x1^2+x2^2*x1^3+z*y1*x2+z*y1*x1+y2*x2^2*x1+y2*x2*x1^2
    +z^2+z*x2^2+z*x2*x1+x2^2*x1^2+x2*x1^3+y2*x2*x1+y2*x1^2
s[6]=T(7)*y1*x1+T(7)*y1+z*y2*x2^2*x1+z^2*y1+z*y1*x2*x1+z^2*y2+z*y2*x2^2
    +y2*x2^2*x1^2+z^2*x2+z*x2^2*x1+z*x2*x1^2+z*y1*x2+y1*x2*x1^2+y2*x2^2*x1
    +y2*x2*x1^2+z^2+z*x2^2+z*x2*x1+y1*x2*x1+y1*x1^2+y2*x2*x1+y1*x1
s[7]=T(7)*x2^2*x1+T(7)*x2*x1^2+T(7)*y1*x2+T(7)*y1+z*y1*x2^3+z*y1*x2^2*x1
    +z*y2*x2^3+z^2*x2^2+z^2*x2*x1+z*x2^3*x1+z*x2^2*x1^2+z^2*y1+z*y1*x2^2
    +y1*x2^3*x1+y1*x2^2*x1^2+z*y2*x2*x1+y2*x2^3*x1+z^2*x2+z*x2^3+z*x2^2*x1
    +x2^3*x1^2+x2^2*x1^3+z*y1*x2+y1*x2*x1^2+x2^2*x1^2+x2*x1^3+y2*x2*x1+y1*x1
s[8]=T(7)*z*x1+z^2*y1*x2+y2*x2^3*x1^2+z^2*x2*x1+x2^4*x1^2+x2^3*x1^3
    +y1*x2^2*x1^2+z^2*y2+y2*x2^2*x1^2+y1*x2^2*x1+z^2
s[9]=T(7)*z*y2+z^2*x2^3+z^2*x2^2*x1+y1*x2^4*x1+y1*x2^3*x1^2+z^2*y2*x2
    +y2*x2^4*x1+y2*x2^3*x1^2+x2^4*x1^2+x2^3*x1^3+z^2*y1+y1*x2^3*x1
    +y1*x2^2*x1^2+y2*x2^3*x1+x2^4*x1+x2^3*x1^2+y1*x2^2*x1+y2*x2^2*x1+y1*x2^2
s[10]=T(7)*z*y1+z^2*y2*x2^2+x2^5*x1^2+x2^4*x1^3+z^2*y1*x2+y1*x2^4*x1
     +y1*x2^3*x1^2+y2*x2^3*x1^2+z^2*x2^2+z^2*x2*x1+x2^4*x1^2+x2^3*x1^3
     +y1*x2^3*x1+y2*x2^3*x1+y1*x2^2*x1
s[11]=T(7)*z^2+x2^6*x1^2+x2^5*x1^3+y2*x2^5*x1+y1*x2^4*x1+z*y2*x2^2*x1
     +x2^4*x1^2+x2^3*x1^3+z*y1*x2^2+z*y2*x2*x1+y2*x2^3*x1+z*x2^2*x1
     +z*x2*x1^2+z*y1*x1+y1*x2^2*x1
s[12]=T(6)*x2+T(6)+T(7)*x2*x1+T(7)*x1^2+z*y1*x2^2+z*y1*x2*x1+z^2*y2
     +z*x2^2*x1+z*x2*x1^2+x2^3*x1^2+x2^2*x1^3+y1*x2^2*x1+y1*x2*x1^2+z*y2*x2
     +z*y2*x1+z^2+x2^2*x1^2+x2*x1^3+y1*x2*x1+y1*x1^2+z*x2+z*x1
s[13]=T(6)*y2+T(6)*x1+T(7)*x2*x1^2+T(7)*x1^3+z*y1*x2^2*x1+z*y1*x2*x1^2
     +z^2*x2*x1+z^2*x1^2+z*x2^2*x1^2+z*x2*x1^3+y1*x2^2*x1^2+y1*x2*x1^3
     +z*y2*x2*x1+z*y2*x1^2+y2*x2^2*x1^2+y2*x2*x1^3+y1*x2*x1^2+y1*x1^3
     +y2*x2*x1^2+y2*x1^3+z*x2*x1+z*x1^2
s[14]=T(6)*y1+T(6)*x1+T(7)*x2*x1^2+T(7)*x1^3+z*y1*x2^2*x1+z*y1*x2*x1^2
     +z*x2^2*x1^2+z*x2*x1^3+z*y2*x2*x1+z*y2*x1^2+z*x2*x1+z*x1^2
s[15]=T(6)*z+y1*x2^3*x1^2+y1*x2^2*x1^3+y2*x2^3*x1^2+y2*x2^2*x1^3
s[16]=T(5)*y1+T(5)*y2+T(7)*y1*x2+T(7)*y2*x2+T(7)*z+T(7)*y1+T(7)*y2
     +z*y2*x2^3+z*x2^4+z*x2^3*x1+z*y1*x2^2+z^2*y2+z^2*x2+z^2*x1+x2^3*x1^2
     +x2^2*x1^3+z*y2*x2+y2*x2*x1^2+z*x2^2+z*x2*x1+x2^2*x1^2+x2*x1^3+z*y1
     +y1*x2^2+y1*x2*x1+y2*x2*x1+y2*x1^2+x2^2*x1+x2*x1^2+y2*x1
s[17]=T(5)*z+x2^4*x1^2+x2^3*x1^3+y2*x2^2*x1^2+y1*x2^3+y1*x2^2*x1+y1*x2^2
s[18]=T(5)*y2*x1+T(5)*x1^2+T(7)*x2*x1^2+T(7)*x1^3+T(7)*x1^2+T(7)*y1
     +T(7)*x1+z*y1*x2^2*x1+z*y1*x2*x1^2+z*x2^2*x1^2+z*x2*x1^3+z^2*y1
     +z*y1*x2*x1+z*y2*x2^2+z*y2*x2*x1+z*y2*x1^2+y2*x2^2*x1^2+z^2*x2
     +z^2*x1+z*x2*x1^2+x2^3*x1^2+x2^2*x1^3+y1*x2^2*x1+y1*x2*x1^2+z*y2*x1
     +y2*x2^2*x1+z*x2^2+z*x2*x1+z*x1^2+x2^2*x1^2+x2*x1^3+y1*x1^2
     +z*y2+y2*x2*x1+y2*x1^2+z*x1+y1*x1+z
s[19]=T(5)*y2*x2+T(5)*y2+T(7)*y1*x2^2+T(7)*y2*x2^2+T(7)*z*x2+T(7)*y2
     +z*y2*x2^4+z*x2^5+z*x2^4*x1+z*y1*x2^3+z*y2*x2^3+z^2*x2^2+z^2*x2*x1
```

```
     +z*x2^4+z*x2^3*x1+z*y1*x2^2+y1*x2^3*x1+y1*x2^2*x1^2+y2*x2^2*x1^2
     +z^2*x2+z^2*x1+z*x2^3+z*x2^2*x1+y1*x2^3+z*y2*x2+y2*x2^2*x1+z^2
     +x2^3*x1+x2^2*x1^2+z*y1+y1*x2*x1+y1*x1^2+y2*x1^2+y1*x1+y2*x1
s[20]=T(5)*x1^3+T(5)*x1^2+T(6)+T(7)*x2*x1^3+T(7)*x1^4+T(7)*x2*x1^2
     +T(7)*x2*x1+T(7)*x1^2+T(7)*y1+T(7)*x1+z*y1*x2^2*x1^2+z*y1*x2*x1^3
     +z^2*x2*x1^2+z^2*x1^3+z*x2^2*x1^3+z*x2*x1^4+z^2*y1*x1+z*y1*x2^2*x1
     +y1*x2^2*x1^3+y1*x2*x1^4+z^2*y2*x1+z*y2*x2*x1^2+z*y2*x1^3
     +y2*x2*x1^4+z*x2^2*x1^2+z*y1*x2^2+z*y1*x2*x1+y1*x2^2*x1^2+y1*x1^4
     +z*y2*x2^2+z*y2*x2*x1+y2*x1^4+z^2*x2+z*x2^2*x1+z*x1^3+y1*x2^2*x1
     +y1*x2*x1^2+y1*x1^3+z*y2*x2+z*y2*x1+y2*x2^2*x1+y2*x2*x1^2+z*x2^2
     +z*x2*x1+y1*x2*x1+z*y2+y2*x2*x1+y2*x1^2+z*x2+z*x1+z
s[21]=T(5)*x2*x1^2+T(5)*x1^2+T(7)*x1^3+T(7)*x2*x1+T(7)*x1^2+T(7)*y1
     +T(7)*x1+z^2*y1*x1+z*y1*x2*x1^2+y2*x2^2*x1^3+z^2*x2*x1+z^2*x1^2
     +z*x2*x1^3+x2^3*x1^3+x2^2*x1^4+z*y1*x2^2+z*y1*x2*x1+y1*x2^2*x1^2
     +y1*x2*x1^3+z*y2*x2^2+z*y2*x1^2+y2*x2^2*x1^2+z^2*x2+z^2*x1
     +z*x2^2*x1+z*x2*x1^2+x2^2*x1^3+x2*x1^4+y1*x2^2*x1+y1*x2*x1^2
     +y1*x1^3+z*y2*x2+z*y2*x1+y2*x2^2*x1+y2*x1^3+z*x2^2+z*x1^2+y1*x2*x1
     +z*y2+y2*x2*x1+y2*x1^2+z*x2+z*x1+z
s[22]=T(5)*x2^2*x1+T(5)*y2+T(7)*x2*x1^2+T(7)*y1*x2+T(7)*y2*x2+T(7)*z
     +T(7)*x2*x1+T(7)*y2+z^2*y1*x2+z*y1*x2^2*x1+z*y2*x2^3+y2*x2^3*x1^2
     +z^2*x2^2+z^2*x2*x1+z*x2^4+z*x2^3*x1+z*x2^2*x1^2+x2^4*x1^2
     +x2^3*x1^3+z^2*y1+y1*x2^3*x1+y1*x2^2*x1^2+z*y2*x2^2+z*y2*x2*x1
     +y2*x2^3*x1+y2*x2^2*x1^2+z^2*x2+z^2*x1+z*x2^2*x1+z*y1*x2
     +y1*x2^2*x1+y2*x2^2*x1+y2*x2*x1^2+z^2+z*x2*x1+x2^2*x1^2+x2*x1^3
     +z*y1+y1*x2^2+y1*x1^2+y2*x2*x1+y2*x1^2+z*x2+x2^2*x1+x2*x1^2
     +y1*x1+y2*x1
s[23]=T(4)+T(5)*x1^2+T(5)+T(7)*x2*x1^2+T(7)*x1^3+T(7)*x1^2+T(7)*y1
     +T(7)*x1+z*y1*x2^2*x1+z*y1*x2*x1^2+z^2*x2*x1+z^2*x1^2+z*x2^2*x1^2
     +z*x2*x1^3+z^2*y1+z*y1*x2*x1+y1*x2^2*x1^2+y1*x2*x1^3+z^2*y2
     +z*y2*x2^2+z*y2*x2*x1+z*y2*x1^2+y2*x2*x1^3+z^2*x2+z^2*x1
     +z*x2*x1^2+y1*x2*x1^2+y1*x1^3+z*y2*x1+y2*x2^2*x1+y2*x2*x1^2
     +y2*x1^3+z*x2^2+z*x2*x1+z*x1^2+z*y2+y2*x1^2+z*x1+y2*x1+z
s[24]=T(3)+T(5)*x2*x1+T(5)+T(7)*x2*x1+T(7)*y1+T(7)*x1+z^2*y1
     +z*y1*x2^2+z*y2*x2^2+y2*x2^2*x1^2+z*x2^2*x1+x2^3*x1^2+x2^2*x1^3
     +z*y2*x2+y2*x2*x1^2+z^2+z*x2^2+x2^2*x1^2+x2*x1^3+z*y2+y2*x2*x1
     +z*x2+y2*x1+z
s[25]=T(2)+T(5)*x2*x1+T(5)*y2+T(5)+T(7)*y1*x2+T(7)*y2*x2+T(7)*z
     +T(7)*x2*x1+T(7)*y1+T(7)*x1+z*y2*x2^3+z*x2^4+z*x2^3*x1+z^2*y1
     +y2*x2^2*x1^2+z^2*x2+z^2*x1+z*x2^3+x2^3*x1^2+x2^2*x1^3+z*y1*x2
     +y1*x2^2*x1+y1*x2*x1^2+z*y2*x2+z^2+z*x2^2+x2^2*x1^2+x2*x1^3
     +y1*x2^2+y1*x2*x1+y1*x1^2+z*y2+y2*x2*x1+y2*x1^2+z*x2+x2^2*x1
     +x2*x1^2+y1*x1+y2*x1+z
s[26]=T(1)+T(5)*x1+T(7)*x2*x1+T(7)*x1^2+T(7)*y1+T(7)*x1+z*y1*x2^2
     +z*y1*x2*x1+z*y2*x2^2+z^2*x2+z^2*x1+z*x2^2*x1+z*x2*x1^2
     +y1*x2^2*x1+y1*x2*x1^2+z*y2*x2+z*y2*x1+y2*x2^2*x1+y2*x2*x1^2
     +z^2+z*x2^2+y1*x1^2+z*y2+y2*x1^2+z*x2+z*x1+y1*x1+y2*x1+z
```

```
s[27]=T(7)^2+T(5)*x2^3+T(5)*x2^2+T(5)*x2*x1+T(5)*y2+T(7)*y1*x2^4
     +T(7)*y2*x2^4+T(7)*z*x2^3+T(7)*y1*x2^2+T(7)*y2*x2^2
     +T(7)*x2*x1^2+T(7)*y1*x2+T(7)*y2*x2+T(7)*z+T(7)*y2+T(7)*x1
     +z*y2*x2^6+z*x2^7+z*x2^6*x1+z*y1*x2^5+z*y2*x2^5+z*x2^6
     +z*x2^5*x1+z*y1*x2^4+z*y2*x2^4+y2*x2^5*x1+z*x2^5+z*x2^4*x1
     +x2^6*x1+x2^5*x1^2+z*y1*x2^3+z*y1*x2^2*x1+y1*x2^4*x1+z*y2*x2^3
     +y2*x2^4*x1+z^2*x2*x1+z*x2^2*x1^2+y1*x2^3*x1+y1*x2^2*x1^2
     +z*y2*x2^2+z*y2*x2*x1+y2*x2^4+y2*x2^3*x1+y2*x2^2*x1^2+z^2*x1
     +z*x2^2*x1+y1*x2^3+y1*x2^2*x1+z*y2*x2+y2*x2^2*x1+z^2+x2^4
     +x2^2*x1^2+z*y1+y1*x2^2+y1*x2*x1+y1*x1^2+z*y2+y2*x2^2+y2*x2*x1
     +y2*x1^2+y2*x1+z+x2^2+x2*x1
s[28]=T(6)*T(7)+T(7)*x2*x1^2+T(7)*x1^3+T(7)*y1*x2+T(7)*y1
     +y2*x2^5*x1^2+y2*x2^4*x1^3+x2^6*x1^2+x2^4*x1^4+z*y1*x2^3*x1
     +z*y1*x2^2*x1^2+y1*x2^4*x1^2+y1*x2^3*x1^3+y2*x2^4*x1^2
     +y2*x2^3*x1^3+z^2*x2^2*x1+z^2*x2*x1^2+z*x2^3*x1^2+z*x2^2*x1^3
     +x2^5*x1^2+x2^3*x1^4+z^2*y1*x2+z^2*y1*x1+z*y1*x2^2*x1
     +z*y1*x2*x1^2+z^2*y2*x2+z*y2*x2^3+z^2*x2^2+z^2*x2*x1
     +z*x2^2*x1^2+z*x2*x1^3+x2^3*x1^3+x2^2*x1^4+z^2*y1+z*y1*x2^2
     +y1*x2^2*x1^2+y1*x2*x1^3+z^2*y2+z*y2*x2^2+z*y2*x2*x1+z*y2*x1^2
     +y2*x2^3*x1+z*x2^3+z*x2^2*x1+x2^2*x1^3+x2*x1^4+z*y1*x2
     +y1*x2^2*x1+z^2+z*x2^2+z*x1^2+y2*x2*x1+y1*x1
s[29]=T(5)*T(7)+T(5)*x2*x1+T(5)*x1+T(7)*z+T(7)*y1+y1*x2^5*x1
     +y1*x2^4*x1^2+z*x2^4*x1+z*x2^3*x1^2+z*y1*x2^3+z*y1*x2^2*x1
     +z^2*y2*x2+z*y2*x2^3+z*y2*x2^2*x1+y2*x2^5+z*x2^4+z*x2^2*x1^2
     +x2^4*x1^2+x2^3*x1^3+z*y1*x2^2+y1*x2^4+z^2*y2+y2*x2^4
     +z*x2^2*x1+x2^5+x2^4*x1+z*y1*x2+y1*x2^3+y1*x2^2*x1+z^2+z*x2*x1
     +x2^4+x2^3*x1+x2^2*x1^2+x2*x1^3+y1*x2^2+y1*x1^2+x2^2*x1
     +x2*x1^2+y1*x1
s[30]=T(6)^2+T(7)*x2*x1^4+T(7)*x1^5+T(7)*y1*x2+T(7)*y1+z*x2^4*x1^3
     +z*x2^2*x1^5+z*y1*x2^2*x1^3+z*y1*x2*x1^4+z^2*y2*x2*x1^2
     +z^2*y2*x1^3+z^2*x2^2*x1^2+z^2*x2*x1^3+z*x2^2*x1^4+z*x2*x1^5
     +x2^4*x1^4+x2^2*x1^6+z^2*y1*x2*x1+z^2*y1*x1^2+y1*x2^2*x1^4
     +y1*x2*x1^5+z^2*y2*x2*x1+z*y2*x2*x1^3+z*y2*x1^4+z^2*x2^2*x1
     +z^2*x1^3+x2^2*x1^5+x2*x1^6+z^2*y1*x2+z^2*y1*x1+y1*x2*x1^4
     +y1*x1^5+z^2*y2*x2+z^2*y2*x1+z*y2*x2^3+z^2*x2^2+z*x2*x1^3
     +z*x1^4+x2^2*x1^4+x2*x1^5+z^2*y1+z*y1*x2^2+z^2*y2+z*y2*x2^2
     +y2*x2^3*x1+z^2*x1+z*x2^3+z*x2^2*x1+z*y1*x2+y1*x2^2*x1+z^2
     +z*x2^2+z*x2*x1+y2*x2*x1+y1*x1
s[31]=T(5)*T(6)+T(7)*x2*x1^3+T(7)*x1^4+T(7)*x2*x1^2+T(7)*x1^3
     +T(7)*x2*x1+T(7)*x1^2+z*y2*x2^3*x1^2+z*y2*x2^2*x1^3
     +z*x2^4*x1^2+z*x2^2*x1^4+z^2*y1*x2*x1+z^2*y1*x1^2+z*y1*x2^3*x1
     +z*y1*x2*x1^3+z*y2*x2^3*x1+z*y2*x2^2*x1^2+y2*x2^3*x1^3
     +y2*x2^2*x1^4+z^2*x2^2*x1+z^2*x1^3+z*x2^3*x1^2+z*x2*x1^4
     +x2^4*x1^3+x2^2*x1^5+z*y1*x2^2*x1+z*y1*x2*x1^2+y1*x2^3*x1^2
     +y1*x2*x1^4+z*y2*x2*x1^2+z*y2*x1^3+y2*x2^3*x1^2+y2*x2^2*x1^3
     +z^2*x2*x1+z^2*x1^2+z*x2^3*x1+z*x2*x1^3+x2^3*x1^3+x2*x1^5
```

```
      +z*y1*x2^2+z*y1*x2*x1+y1*x2^2*x1^2+y1*x1^4+z^2*y2+z*y2*x2*x1
      +z*y2*x1^2+y2*x2*x1^3+y2*x1^4+z*x2^2*x1+z*x1^3+x2^3*x1^2
      +x2^2*x1^3+y1*x2^2*x1+y1*x2*x1^2+z*y2*x2+z*y2*x1+y2*x2*x1^2
      +y2*x1^3+z^2+z*x2*x1+z*x1^2+x2^2*x1^2+x2*x1^3+y1*x2*x1
      +y1*x1^2+z*x2+z*x1
s[32]=T(5)^2+T(5)*y2+T(5)*x2+T(5)*x1+T(5)+T(7)*x2*x1^2+T(7)*x1^3
      +T(7)*y2*x2+T(7)*z+T(7)*x2^2+T(7)*x2*x1+T(7)*x1^2+T(7)*x1
      +z*y1*x2^3*x1+z*y1*x2^2*x1^2+z^2*x2^2*x1+z^2*x2*x1^2
      +z*x2^4*x1+z*x2^2*x1^3+z^2*y1*x2+z^2*y1*x1+z*y1*x2^2*x1
      +z*y1*x2*x1^2+y1*x2^3*x1^2+y1*x2^2*x1^3+z*y2*x2^3
      +z*y2*x2^2*x1+z*x2^4+z*x2^3*x1+z*x2^2*x1^2+z*x2*x1^3
      +x2^4*x1^2+x2^2*x1^4+z^2*y1+z*y1*x2*x1+y1*x2^2*x1^2
      +y1*x2*x1^3+z*y2*x2^2+z*y2*x2*x1+z*y2*x1^2+y2*x2^2*x1^2
      +z^2*x2+z^2*x1+z*x2^3+z*x2*x1^2+x2^2*x1^3+x2*x1^4+z*y2*x2
      +z*y2*x1+z*x1^2+x2^2*x1^2+x2*x1^3+y1*x2^2+y1*x2*x1+z*y2
      +y2*x1^2+z*x2+z*x1+x2^2*x1+x2*x1^2+z
```

Since the numerators of the fractions given in `nor[2][1]` are not even reduced modulo the input ideal of relations, let alone interreduced, it is hard to take this result seriously mathematically. (Moreover the relations s[26], s[25], s[24], s[23] are clearly admissions that T(1), T(2), T(3), T(4) are extraneous variables.) But were we to take this seriously and expect it to be consistent with canonical forms, we would have to think that elements of $Q(A)$ have a meaningful canonical form with denominator of the form $\delta(z, a)$ for some second fixed element $a \in A$ independent of z, since the denominators of all these fractions generating this strict affine A-algebra have denominator z^2. Since there is a relation involving only z, x_2, x_1, I suppose that means that I can take my canonical version and invert either x_2 or x_1 to get one such. But is it unique? Unless the relation involving z, x_2, x_1 has a monomial that is a power of either x_2 or x_1, it is probably clear that it is not unique.

We would also have to believe that it makes more sense (for example from `s[5]`) that $T(7)(y_2x_1+y_1)$ reduces to $zx_2^3 * x_1 + zx_2^2x_1^2 + z^2y_2 + zy_2x_2^2 + zy_2x_2x_1 + z^2x_2 + z^2x_1 + x_2^3x_1^2 + x_2^2x_1^3 + zy_1x_2 + zy_1x_1 + y_2x_2^2x_1 + y_2x_2x_1^2 + z^2 + zx_2^2 + zx_2x_1 + x_2^2x_1^2 + x_2x_1^3 + y_2x_2x_1 + y_2x_1^2$ rather than the other way around.

4 Macaulay2

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

i1 : loadPackage "QthPower";

i2 : wtr=matrix{{19,15,12,9,9},{12,9,9,9,0}};

i3 : R=QQ[z,y1,y2,x2,x1,Weights=>entries weightGrevlex(wtr)];

i4 : I={y1^2-y2*x2*x1,
        y2*y1-y1-x2^2*x1-x2*x1^2,
        y2^2-y2-y1*(x2+x1),
        z^3-(y2+y1)*(x2+1)*x1*z-(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1};

i5 : A=R/ideal(I);

i6 : time ic0=icFractions A;
     -- used 2.74919 seconds

i7 : toString ic0

o7 = {(z*y1*x2-z*y2+z)/(x1),
      (z^2*y1*x2+z^2*y1*x1-z^2*y2+z^2)/(y1*x1),
      (z^2*x2)/(y1),
       z, y1, y2, x2, x1}

i8 : G=gens gb presentation integralClosure A

o8 = | y2^2-y1x2-y1x1-y2
       y1y2-x2^2x1-x2x1^2-y1
       y1^2-y2x2x1
       z3-y2x2^3x1^2-zy1x2x1-y1x2^3x1-y1x2^2x1^2-zy2x2x1-zy1x1-zy2x1-y2x2^2x1
       w_(0,0)y1-z2x2
       w_(0,0)x2x1+w_(0,0)x1^2-z2y2+z2
       w_(0,0)z-y1x2^3x1-zx2^2x1-x2^4x1-x2^3x1^2-zy1x2-zx2x1-zy1-y1x2^2
       w_(0,0)y2x1-z2y1
       w_(0,0)^2-w_(0,0)x1^3+w_(0,0)x1^2-zx2^4x1-z2y2x2+z2y2x1
          -zy2x2^3-z2x2^2-z2y2-z2x1+z2
       w_(1,1)y2-w_(0,0)x2^2+w_(0,0)x1^2-2z2y2+2z2
       w_(1,1)y1-w_(0,0)y2x2+w_(0,0)x2+w_(0,0)x1-z2y1
       w_(1,1)x2x1+w_(0,0)y2-w_(0,0)-z2x2^2-z2x2x1
       w_(1,1)z-y2x2^4x1-y2x2^3x1^2-zy1x2^2-zy1x2x1-y1x2^4-2y1x2^3x1-y1x2^2x1^2
          -zy2x2^2-zy2x2x1+x2^4x1+x2^3x1^2-zy1x2-zy1x1-zy2x1+zx2^2+zx2x1+zy2
```

```
   +zx1-z
w_(1,1)w_(0,0)-w_(0,0)y2x2^2+w_(0,0)x2^2-w_(0,0)x1^2+w_(0,0)y2+w_(0,0)x1
   -w_(0,0)-zy1x2^4-zy1x2^3x1-z2x2^3-z2x2^2x1-zx2^5-2zx2^4x1-zx2^3x1^2
   -z2y1x2+zy2x2^3-z2x2^2-z2x2x1-z2y1+z2y2-zx2^3-z2
w_(1,0)x1-zy1x2+zy2-z
w_(1,0)y2-zx2^3-zx2^2x1+zy1
w_(1,0)y1-zy2x2^2+zx2^2+zx2x1
w_(1,0)z-w_(0,0)y2x2+w_(0,0)x2+w_(0,0)x1
w_(1,0)w_(0,0)-y2x2^5x1-zy1x2^3-y1x2^5-y1x2^4x1-zy2x2^3+x2^5x1+x2^4x1^2
   -zy1x2^2+y2x2^3x1+zx2^3+zx2^2x1+zy2x2+zx2x1-zx2
w_(1,1)^2-w_(1,0)x2^4-w_(1,0)x2^2+w_(1,1)x2^2+w_(1,1)x1-w_(1,1)
   -w_(0,0)y2x2^3-w_(0,0)x2^4+w_(0,0)x1^4-w_(0,0)y2x2^2+w_(0,0)x2^3
   +w_(0,0)x1^3+2w_(0,0)x2^2-2w_(0,0)x1^2-w_(0,0)y2+w_(0,0)-zy2x2^5
   -2zy2x2^4x1-zy2x2^3x1^2-2z2y1x2^2-z2y1x2x1-3zy1x2^4-3zy1x2^3x1
   -zy1x2^2x1^2-3z2y2x2^2-z2y2x1^2+2zx2^5+4zx2^4x1+2zx2^3x1^2-2z2y1x2
   -z2y1x1+z2y2x2-2z2y2x1+2zy2x2^3+zy2x2^2x1+4z2x2^2+z2x2x1+z2x1^2
   -zy1x2^2+4z2y2-z2x2+2z2x1-2zx2^3-zx2^2x1-4z2
w_(1,0)w_(1,1)+w_(1,0)x2^2-w_(1,0)-x2^7x1-2x2^6x1^2-x2^5x1^3-zy2x2^4
   -zy2x2^3x1-y2x2^6-2y2x2^5x1-y2x2^4x1^2-zx2^5-2zx2^4x1-zx2^3x1^2
   +y1x2^5+2y1x2^4x1+y1x2^3x1^2-zy2x2^3-zy2x2^2x1+zx2^4+zx2^3x1+x2^6
   +3x2^5x1+3x2^4x1^2+x2^3x1^3+2zy1x2^2+zy1x2x1-y2x2^4-y2x2^3x1+2zx2^3
   +3zx2^2x1+zx2x1^2+zy1x2+zy1x1-zy2x2+x2^4+x2^3x1-zy1-zy2+zx2+z
w_(1,0)^2+w_(1,1)x2^2-w_(0,0)x2^4+w_(0,0)x1^4-w_(0,0)y2+w_(0,0)-z2y2x2^2
   +z2y2x2x1-z2y2x1^2+2z2x2^2-z2x2x1+z2x1^2 |
```

Here, judging from the denominators produced, the induced canonical form on elements of $Q(A)$ would have denominators of the form $\delta(y_1, x_1)$.

And one would have to buy into, say,

```
w_(1,0)x1
```

reducing to `-zy1x2-zy2+z` rather than the other way around.

```
i10 : R2=ZZ/2[z,y1,y2,x2,x1,Weights=>entries weightGrevlex(wtr)];

i11 : I2={y1^2-y2*x2*x1,
         y2*y1-y1-x2^2*x1-x2*x1^2,
         y2^2-y2-y1*(x2+x1),
         z^3-(y2+y1)*(x2+1)*x1*z-(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1};

i12 : A2=R2/ideal(I2);

i13 : time ic2=icFractions A2;
     -- used 2.15463 seconds

i14 : toString ic2

o14 = {(z^2*y2+z^2*x1+x2^3*x1^2+x2^2*x1^3+y1*x2^2*x1+y2*x2*x1^2+z^2+x2^2*x1^2
       +x2*x1^3+y1*x2*x1+y2*x1^2)
      /(y1*x1+y2*x1),
       (z*x2^3*x1+z*x2^2*x1^2+z*y1*x2^2+z*y1*x1^2+z^2*y2+z*x2^2*x1+z*x1^3
       +x2^3*x1^2+x2^2*x1^3+z*y1*x2+z*y1*x1+y1*x2^2*x1+y1*x2*x1^2+z*y2*x2
       +z*y2*x1+z^2+z*x2*x1+z*x1^2+x2^2*x1^2+x2*x1^3+y1*x2*x1+y1*x1^2+z*y2
       +z*x2+z*x1+z)
      /(y2*x1+x2*x1+x1^2+x1),
       (y1*x2^2*x1+y2*x2^2*x1)
      /(z),
        z, y1, y2, x2, x1}

i15 : G2=gens gb presentation integralClosure A2

o15 = | y2^2+y1x2+y1x1+y2
        y1y2+x2^2x1+x2x1^2+y1
        y1^2+y2x2x1
        z3+y2x2^3x1^2+zy1x2x1+y1x2^3x1+y1x2^2x1^2+zy2x2x1+zy1x1+zy2x1+y2x2^2x1
        w_(0,0)y1+w_(0,0)y2+z2+y1x2x1+y2x2x1+y1x1+y2x1
        w_(0,0)z+y1x2^2x1+y2x2^2x1
        w_(0,0)x2^2x1+w_(0,0)x2x1^2+w_(0,0)y2x2+w_(0,0)y2x1+z2y2+z2x2+z2x1
           +x2^3x1^2+x2^2x1^3+y2x2^2x1+y2x2x1^2+z2+x2^2x1^2+x2x1^3+y2x2x1
           +y2x1^2
        w_(0,0)y2x2x1+w_(0,0)y2x2+w_(0,0)y2x1+w_(0,0)y2+z2y1+z2y2+y2x2^2x1^2
           +z2x2+z2x1+y2x2^2x1+y2x1^2+y2x1
        w_(0,0)^2+w_(0,0)x2x1+w_(0,0)x1
           +zx2^2x1
        w_(1,0)x1+w_(0,0)y2+w_(0,0)x1+w_(0,0)
        w_(1,0)y1+w_(0,0)x2^2+w_(0,0)x2x1+w_(0,0)y2+z2+y1x2x1+y2x2x1+y1x1+y2x1
        w_(1,0)z+x2^4x1+x2^3x1^2+y1x2^3+y2x2^2x1
        w_(1,0)w_(0,0)+w_(0,0)y2x2+w_(0,0)x2x1+w_(0,0)y2+w_(0,0)x2+w_(0,0)x1
```

```
   +w_(0,0)+zy2x2^2+zx2^2x1+zx2^2
w_(1,1)x1+w_(0,0)y2+zy1x2+zy2x1+z2+zx2x1+zx1^2+y1x2x1+zy2+y2x2x1
   +y1x1+y2x1+z
w_(1,1)y1+w_(0,0)y2x2+zy2x2^2+zx2^2x1+zx2x1^2+zy1x2+zy1x1+zx2^2
   +zx2x1+zy1
w_(1,1)z+w_(0,0)y2x2^2+w_(0,0)x2^3+w_(0,0)x2x1^2+w_(0,0)y2x2
   +w_(0,0)y2x1+w_(0,0)x2^2+w_(0,0)x2x1+w_(0,0)y2+x2^3x1^2+x2^2x1^3
   +y2x2x1^2+x2^2x1^2+x2x1^3+y1x2^2+y2x1^2+x2^2x1+x2x1^2+y2x1
w_(1,1)y2x2+w_(1,0)y2+w_(0,0)x2^3+w_(0,0)x2x1^2+w_(0,0)y2x2
   +w_(0,0)y2x1+zx2^4+zx2^3x1+zy1x2^2+zy1x2x1+z2y2+zy2x2^2+zy2x2x1+z2x2
   +z2x1+x2^3x1^2+x2^2x1^3+zy1x2+zy2x2+y2x2^2x1+y2x2x1^2+x2^2x1^2
   +x2x1^3+y1x2x1+y2x1^2+y1x1+y2x1
w_(1,1)w_(0,0)+w_(0,0)y2x2+w_(0,0)y2+y2x2^4x1+x2^5x1+x2^3x1^3
   +zy1x2^2+y2x2^3x1+y2x2^2x1^2+z2x2+x2^4x1+x2^3x1^2+y1x2^2x1+z2+y1x1
   +y2x1
w_(1,0)^2+w_(1,1)x2^2+w_(1,0)y2+w_(1,0)x2+w_(1,0)+w_(0,0)y2x2
   +w_(0,0)x2x1+w_(0,0)y2+w_(0,0)x2+w_(0,0)x1+w_(0,0)+zy1x2^2+zy2x2^2
   +z2x2+zx2^3+y1x2^2x1+y2x2^2x1+y1x2x1+y2x2x1
w_(1,0)w_(1,1)+w_(0,0)x2^3+w_(0,0)x2x1^2+w_(0,0)y2x1+w_(0,0)x2^2
   +w_(0,0)x2x1+w_(0,0)y2+y1x2^5+y1x2^4x1+y2x2^5+y2x2^4x1+y2x2^3x1^2
   +zx2^4+zx2^3x1+x2^5x1+x2^3x1^3+zy1x2^2+y1x2^4+y1x2^2x1^2+z2y2+y2x2^4
   +y2x2^2x1^2+z2x1+x2^5+x2^4x1+x2^3x1^2+x2^2x1^3+y1x2^3+y2x2^2x1
   +y2x2x1^2+x2^4+x2^3x1+x2^2x1^2+x2x1^3+y2x2x1+y2x1^2+y1x1+y2x1
w_(1,1)^2+w_(1,0)y2x2^3+w_(1,0)y2x2^2+w_(1,0)x2^3+w_(1,0)x2^2
   +w_(0,0)x2^5+w_(0,0)x2x1^4+w_(0,0)y2x1^3+w_(0,0)y2x2^2+w_(0,0)x2^3
   +w_(0,0)x2^2+w_(0,0)x2x1+w_(0,0)x2+z2y2x2^2+z2y2x2x1+z2y2x1^2
   +z2x2^3+z2x1^3+x2^6x1+x2^2x1^5+z2y1x2+z2y1x1+y1x2^5+y1x2^4x1+zy2x2^3
   +y2x2^4x1+y2x2x1^4+z2x2^2+z2x2x1+x2^5x1+x2x1^5+z2y2+y2x2^4+y2x1^4
   +y1x2^3+y1x2^2x1+x2^4+x2^2x1^2+y2x2^2+y2x2x1+x2^2x1+x2x1^2+x2^2+x2x1 |
```

But now what would the denominators be? They would involve y1+y2, x1, and z. Oops.

```
i16 : R3=ZZ/3[z,y1,y2,x2,x1,Weights=>entries weightGrevlex(wtr)];

i17 : I3={y1^2-y2*x2*x1,
          y2*y1-y1-x2^2*x1-x2*x1^2,
          y2^2-y2-y1*(x2+x1),
          z^3-(y2+y1)*(x2+1)*x1*z-(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1};

i18 : A3=R3/ideal(I3);

i19 : time ic3=icFractions A3;
     -- used 0.68462 seconds

i20 : toString ic3

o20 = {(z*y1*x2-z*y2+z)
      /(x1),
       (z^2*y1*x2+z^2*y1*x1-z^2*y2+z^2)
      /(y1*x1),
       (z^2*x2)
      /(y1),
        z, y1, y2, x2, x1}

i21 : G3=gens gb presentation integralClosure A3

o21 = | y2^2-y1x2-y1x1-y2
        y1y2-x2^2x1-x2x1^2-y1
        y1^2-y2x2x1
        z3-y2x2^3x1^2-zy1x2x1-y1x2^3x1-y1x2^2x1^2-zy2x2x1-zy1x1-zy2x1-y2x2^2x1
        w_(0,0)y1-z2x2 w_(0,0)x2x1+w_(0,0)x1^2-z2y2+z2
        w_(0,0)z-y1x2^3x1-zx2^2x1-x2^4x1-x2^3x1^2-zy1x2-zx2x1-zy1-y1x2^2
        w_(0,0)y2x1-z2y1 w_(0,0)^2-w_(0,0)x1^3+w_(0,0)x1^2-zx2^4x1-z2y2x2+z2y2x1
           -zy2x2^3-z2x2^2-z2y2-z2x1+z2
        w_(1,1)y2-w_(0,0)x2^2+w_(0,0)x1^2+z2y2-z2
        w_(1,1)y1-w_(0,0)y2x2+w_(0,0)x2+w_(0,0)x1-z2y1
        w_(1,1)x2x1+w_(0,0)y2-w_(0,0)-z2x2^2-z2x2x1
        w_(1,1)z-y2x2^4x1-y2x2^3x1^2-zy1x2^2-zy1x2x1-y1x2^4+y1x2^3x1-y1x2^2x1^2
           -zy2x2^2-zy2x2x1+x2^4x1+x2^3x1^2-zy1x2-zy1x1-zy2x1+zx2^2+zx2x1
           +zy2+zx1-z
        w_(1,1)w_(0,0)-w_(0,0)y2x2^2+w_(0,0)x2^2-w_(0,0)x1^2+w_(0,0)y2+w_(0,0)x1
           -w_(0,0)-zy1x2^4-zy1x2^3x1-z2x2^3-z2x2^2x1-zx2^5+zx2^4x1-zx2^3x1^2
           -z2y1x2+zy2x2^3-z2x2^2-z2x2x1-z2y1+z2y2-zx2^3-z2 w_(1,0)x1-zy1x2+zy2-z
        w_(1,0)y2-zx2^3-zx2^2x1+zy1 w_(1,0)y1-zy2x2^2+zx2^2+zx2x1
        w_(1,0)z-w_(0,0)y2x2+w_(0,0)x2+w_(0,0)x1
        w_(1,0)w_(0,0)-y2x2^5x1-zy1x2^3-y1x2^5-y1x2^4x1-zy2x2^3+x2^5x1+x2^4x1^2
           -zy1x2^2+y2x2^3x1+zx2^3+zx2^2x1+zy2x2+zx2x1-zx2
```

```
w_(1,1)^2-w_(1,0)x2^4-w_(1,0)x2^2+w_(1,1)x2^2+w_(1,1)x1-w_(1,1)
   -w_(0,0)y2x2^3-w_(0,0)x2^4+w_(0,0)x1^4-w_(0,0)y2x2^2+w_(0,0)x2^3
   +w_(0,0)x1^3-w_(0,0)x2^2+w_(0,0)x1^2-w_(0,0)y2+w_(0,0)-zy2x2^5
   +zy2x2^4x1-zy2x2^3x1^2+z2y1x2^2-z2y1x2x1-zy1x2^2x1^2-z2y2x1^2-zx2^5
   +zx2^4x1-zx2^3x1^2+z2y1x2-z2y1x1+z2y2x2+z2y2x1-zy2x2^3+zy2x2^2x1
   +z2x2^2+z2x2x1+z2x1^2-zy1x2^2+z2y2-z2x2-z2x1+zx2^3-zx2^2x1-z2
w_(1,0)w_(1,1)+w_(1,0)x2^2-w_(1,0)-x2^7x1+x2^6x1^2-x2^5x1^3-zy2x2^4
   -zy2x2^3x1-y2x2^6+y2x2^5x1-y2x2^4x1^2-zx2^5+zx2^4x1-zx2^3x1^2+y1x2^5
   -y1x2^4x1+y1x2^3x1^2-zy2x2^3-zy2x2^2x1+zx2^4+zx2^3x1+x2^6+x2^3x1^3
   -zy1x2^2+zy1x2x1-y2x2^4-y2x2^3x1-zx2^3+zx2x1^2+zy1x2+zy1x1-zy2x2+x2^4
   +x2^3x1-zy1-zy2+zx2+z
w_(1,0)^2+w_(1,1)x2^2-w_(0,0)x2^4+w_(0,0)x1^4-w_(0,0)y2+w_(0,0)-z2y2x2^2
   +z2y2x2x1-z2y2x1^2-z2x2^2-z2x2x1+z2x1^2 |
```

```
i81 : R2=ZZ/2[z,y1,y2,x2,x1,Weights=>entries weightGrevlex(wtr)];

i82 : I2={y1^2-y2*x2*x1,
          y2*y1-y1-x2^2*x1-x2*x1^2,
          y2^2-y2-y1*(x2+x1),
          z^3-(y2+y1)*(x2+1)*x1*z-(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1};

i83 : A2=R2/ideal(I2);

i84 : time icp2=icFracP A2;
     -- used 85.2053 seconds

i85 : toString icp2

o85 = {1,
       (z^2*y2+z^2)
      /(y1*x1+y2*x1),
       (z^2*y1+z^2*y2+z^2)
      /(y1*x1+y2*x1),
        y1*x2*x1+z*x1,
        x2^3*x1+x2^2*x1^2+y1*x2^2+y1*x2*x1+z*y2+y1*x2,
       (z^2*x2*x1+z^2*x1^2+z^2*y1+z^2*y2+z^2)
      /(y1*x1+y2*x1),
       (z*y1*x2+z*y1*x1+z*x2*x1+z*y2+z)
      /(x1)}
```

5 MAGMA

This example has two independent variables, so MAGMA's `IntegralClosure` function does not apply. But even its `Normalisation` function runs out of storage for me on this in any characteristic.

6 My qth-power algortihm in Macaulay2

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

i1 : loadPackage "QthPower";

i2 : wtR=matrix{{19,15,12,9,9},{12,9,9,9,0}};

i3 : Rq=QQ[z,y1,y2,x2,x1,Weights=>entries weightGrevlex(wtR)];

i4 : Iq={y1^2+y2*x2*x1,
        y2*y1+y1+x2^2*x1+x2*x1^2,
        y2^2+y2+y1*(x2+x1),
        z^3+(y2+y1)*(x2+1)*x1*z+(y2*(x2*x1+1)+y1*(x2+x1))*x2^2*x1};

i5 : time ic0=rationalIntegralClosure(wtR,Rq,Iq);
3
5
7
11
     -- used 569.06 seconds
i6 : toString ic0

o6 = ({   x2^2*x1^2+x2*x1^3,
       y2*x2^2*x1^2+y2*x2*x1^3,
       y1*x2^2*x1^2+y1*x2*x1^3,
        z*x2^2*x1^2+z*x2*x1^3,
     z*y2*x2^2*x1^2+z*y2*x2*x1^3,
   z^2*y2*x2*x1+z^2*x2*x1,
     z*y1*x2^2*x1^2+z*y1*x2*x1^3,
     z*y1*x2^3*x1-z*y1*x2^2*x1^2-2*z*y1*x2*x1^3-z*y2*x2^2*x1-z*y2*x2*x1^2
         -z*x2^2*x1-z*x2*x1^2,
   z^2*y1*x2*x1+z^2*y1*x1^2-z^2*y2*x1-z^2*x1,
      z^2*x2^2*x1^2+z^2*x2*x1^3,
      z^2*x2^3*x1-z^2*x2^2*x1^2-2*z^2*x2*x1^3+z^2*y1*x2+z^2*y1*x1-z^2*y2-z^2},

     p_0^2+p_0*p_10^2-p_0+4*p_1*p_10^2-8*p_1*p_10*p_11+p_1*p_10-8*p_1*p_11
        -3*p_1+p_2*p_10^4-4*p_2*p_10^3*p_11+4*p_2*p_10^2*p_11^2+p_2*p_10^3
        -4*p_2*p_10^2*p_11+4*p_2*p_10*p_11^2+p_2*p_10+p_2+p_3*p_10^4
        +p_3*p_10^2-p_4*p_10^4+4*p_4*p_10^2*p_11^2+3*p_4*p_10^2+p_5*p_10^4
        -3*p_5*p_10^3*p_11+4*p_5*p_10*p_11^3-5*p_5*p_10^2*p_11
        +8*p_5*p_10*p_11^2+4*p_5*p_11^3-p_5*p_10^2-2*p_5*p_10*p_11
        +8*p_5*p_11^2+p_6*p_10^5-4*p_6*p_10^4*p_11+4*p_6*p_10^3*p_11^2
        -2*p_6*p_10^3+8*p_6*p_10^2*p_11+2*p_7*p_10^5-2*p_7*p_10^4*p_11
```

```
   -4*p_7*p_10^3*p_11^2-2*p_7*p_10^3-p_7*p_10^2*p_11,

p_0*p_1+3*p_1*p_10*p_11+3*p_1*p_11+p_2*p_10^3*p_11-2*p_2*p_10^2*p_11^2
   +p_2*p_10^2*p_11-2*p_2*p_10*p_11^2+p_4*p_10^4-p_4*p_10^3*p_11
   -2*p_4*p_10^2*p_11^2+p_5*p_10^3*p_11-p_5*p_10^2*p_11^2
   -2*p_5*p_10*p_11^3+p_5*p_10^2*p_11-4*p_5*p_10*p_11^2-2*p_5*p_11^3
   -3*p_5*p_11^2+p_6*p_10^4*p_11-2*p_6*p_10^3*p_11^2-3*p_6*p_10^2*p_11
   +p_7*p_10^4*p_11+p_7*p_10^3*p_11^2,

p_0*p_2+p_1*p_10^3-p_1*p_10^2*p_11-2*p_1*p_10*p_11^2-2*p_1*p_10*p_11
   -2*p_1*p_11^2-p_1*p_10-p_1*p_11+p_2*p_10^2+p_2*p_10*p_11+p_2*p_11
   -p_2+p_4*p_10^3+p_4*p_10^2*p_11-p_5*p_10^3*p_11+p_5*p_10^2*p_11^2
   +2*p_5*p_10*p_11^3-p_5*p_10^2*p_11+p_5*p_10*p_11^2+2*p_5*p_11^3
   -p_6*p_10^4+p_6*p_10^3*p_11+2*p_6*p_10^2*p_11^2-p_6*p_10^2
   +p_7*p_10^5*p_11-p_7*p_10^4*p_11^2-2*p_7*p_10^3*p_11^3-p_7*p_10^4
   -2*p_7*p_10^3*p_11-p_7*p_10^2*p_11^2-p_7*p_10^2,

p_0*p_3+p_3*p_10^2-p_3+4*p_4*p_10^2-8*p_4*p_10*p_11+p_4*p_10-8*p_4*p_11
   -3*p_4-p_6*p_10^4+4*p_6*p_10^3*p_11-4*p_6*p_10^2*p_11^2-p_6*p_10^3
   +4*p_6*p_10^2*p_11-4*p_6*p_10*p_11^2-p_6*p_10-p_6+p_7*p_10^5
   -3*p_7*p_10^4*p_11+4*p_7*p_10^2*p_11^3-p_7*p_10^4-p_7*p_10^3*p_11
   +4*p_7*p_10^2*p_11^2+4*p_7*p_10*p_11^3-2*p_7*p_10^3
   +2*p_7*p_10^2*p_11+4*p_7*p_10*p_11^2-p_7*p_10-p_7+p_8*p_10^5
   +2*p_8*p_10^4*p_11-8*p_8*p_10^3*p_11^2-9*p_8*p_10^2*p_11-p_9*p_10^6
   +3*p_9*p_10^5*p_11-4*p_9*p_10^3*p_11^3-p_9*p_10^4-p_9*p_10^3*p_11
   +p_10^7*p_11-3*p_10^6*p_11^2+4*p_10^4*p_11^4-p_10^6-3*p_10^5*p_11
   +6*p_10^4*p_11^2+8*p_10^3*p_11^3-p_10^4-p_10^3*p_11,

p_0*p_4+3*p_4*p_10*p_11+3*p_4*p_11-p_6*p_10^3*p_11+2*p_6*p_10^2*p_11^2
   -p_6*p_10^2*p_11+2*p_6*p_10*p_11^2+p_7*p_10^4*p_11-p_7*p_10^3*p_11^2
   -2*p_7*p_10^2*p_11^3-2*p_7*p_10^2*p_11^2-2*p_7*p_10*p_11^3
   -p_7*p_10^2*p_11-p_7*p_10*p_11^2+3*p_8*p_10^3*p_11^2+3*p_8*p_10^2*p_11
   -p_9*p_10^5*p_11+p_9*p_10^4*p_11^2+2*p_9*p_10^3*p_11^3+p_10^6*p_11^2
   -p_10^5*p_11^3-2*p_10^4*p_11^4,

p_0*p_5+p_1*p_10^3-2*p_1*p_10^2*p_11+p_1*p_10^2-2*p_1*p_10*p_11-p_2*p_10^3
   +2*p_2*p_10^2*p_11+2*p_2*p_10*p_11+p_2*p_10-p_4*p_10^4
   +2*p_4*p_10^3*p_11+3*p_4*p_10^2+3*p_5*p_10*p_11+3*p_5*p_11+p_6*p_10^3
   +p_7*p_10^5-p_7*p_10^4*p_11-2*p_7*p_10^3*p_11^2+p_7*p_10^3,

p_0*p_6-p_4*p_10^3+p_4*p_10^2*p_11+2*p_4*p_10*p_11^2-p_4*p_10^2
   +4*p_4*p_10*p_11+2*p_4*p_11^2+3*p_4*p_11+3*p_6*p_10*p_11+3*p_6*p_11
   +p_7*p_10^4*p_11-p_7*p_10^3*p_11^2-2*p_7*p_10^2*p_11^3+p_7*p_10^3*p_11
   -p_7*p_10^2*p_11^2-2*p_7*p_10*p_11^3-p_8*p_10^5*p_11+p_8*p_10^4*p_11^2
   +2*p_8*p_10^3*p_11^3-p_8*p_10^4+4*p_8*p_10^3*p_11+5*p_8*p_10^2*p_11^2
   +3*p_9*p_10^3*p_11^2+3*p_9*p_10^2*p_11+p_10^6*p_11-3*p_10^4*p_11^3
```

```
   -2*p_10^3*p_11^4,

p_0*p_7+p_4*p_10^2-2*p_4*p_10*p_11+p_4*p_10-2*p_4*p_11+p_6*p_10^2
   -2*p_6*p_10*p_11-2*p_6*p_11-p_6+p_7*p_10^2+p_7*p_10*p_11+p_7*p_11-p_7
   +p_8*p_10^4-p_8*p_10^3*p_11-2*p_8*p_10^2*p_11^2+p_9*p_10^4*p_11
   -2*p_9*p_10^3*p_11^2-3*p_9*p_10^2*p_11+p_10^4*p_11+p_10^3*p_11^2,

p_0*p_8-p_2*p_10^2+2*p_2*p_10*p_11,

p_0*p_9-3*p_1-p_5*p_10^2+p_5*p_10*p_11+2*p_5*p_11^2,

p_1^2-p_1*p_10*p_11-p_1*p_11+p_2*p_10^2*p_11^2+p_2*p_10*p_11^2
   +p_4*p_10^3*p_11+p_4*p_10^2*p_11^2+p_5*p_10^2*p_11^2+p_5*p_10*p_11^3
   +2*p_5*p_10*p_11^2+p_5*p_11^3+p_5*p_11^2+p_6*p_10^3*p_11^2
   +p_6*p_10^2*p_11,

p_1*p_2+p_1*p_10^2*p_11+p_1*p_10*p_11^2+p_1*p_10*p_11+p_1*p_11^2
   -p_5*p_10^2*p_11^2-p_5*p_10*p_11^3-p_5*p_10*p_11^2-p_5*p_11^3
   -p_6*p_10^3*p_11-p_6*p_10^2*p_11^2+p_7*p_10^4*p_11^2+p_7*p_10^3*p_11^3,

p_1*p_3+3*p_4*p_10*p_11+3*p_4*p_11-p_6*p_10^3*p_11+2*p_6*p_10^2*p_11^2
   -p_6*p_10^2*p_11+2*p_6*p_10*p_11^2+p_7*p_10^4*p_11-p_7*p_10^3*p_11^2
   -2*p_7*p_10^2*p_11^3-2*p_7*p_10^2*p_11^2-2*p_7*p_10*p_11^3
   -p_7*p_10^2*p_11-p_7*p_10*p_11^2+3*p_8*p_10^3*p_11^2+3*p_8*p_10^2*p_11
   -p_9*p_10^5*p_11+p_9*p_10^4*p_11^2+2*p_9*p_10^3*p_11^3+p_10^6*p_11^2
   -p_10^5*p_11^3-2*p_10^4*p_11^4,

p_1*p_4-p_4*p_10*p_11-p_4*p_11-p_6*p_10^2*p_11^2-p_6*p_10*p_11^2
   +p_7*p_10^3*p_11^2+p_7*p_10^2*p_11^3+p_7*p_10^2*p_11^2+p_7*p_10*p_11^3
   -p_8*p_10^3*p_11^2-p_8*p_10^2*p_11-p_9*p_10^4*p_11^2-p_9*p_10^3*p_11^3
   +p_10^5*p_11^3+p_10^4*p_11^4+p_10^4*p_11^2+p_10^3*p_11^3,

p_1*p_5+p_1*p_10^2*p_11+p_1*p_10*p_11-p_2*p_10^2*p_11-p_2*p_10*p_11
   -p_4*p_10^3*p_11-p_4*p_10^2-p_5*p_10*p_11-p_5*p_11+p_7*p_10^4*p_11
   +p_7*p_10^3*p_11^2,

p_1*p_6-p_4*p_10^2*p_11-p_4*p_10*p_11^2-2*p_4*p_10*p_11-p_4*p_11^2-p_4*p_11
   -p_6*p_10*p_11-p_6*p_11+p_7*p_10^3*p_11^2+p_7*p_10^2*p_11^3
   +p_7*p_10^2*p_11^2+p_7*p_10*p_11^3-p_8*p_10^4*p_11^2-p_8*p_10^3*p_11^3
   -2*p_8*p_10^3*p_11-2*p_8*p_10^2*p_11^2-p_9*p_10^3*p_11^2
   -p_9*p_10^2*p_11+p_10^5*p_11^2+2*p_10^4*p_11^3+p_10^3*p_11^4,

p_1*p_7+p_4*p_10*p_11+p_4*p_11+p_6*p_10*p_11+p_6*p_11+p_8*p_10^3*p_11
   +p_8*p_10^2*p_11^2+p_9*p_10^3*p_11^2+p_9*p_10^2*p_11,

p_1*p_8-p_2*p_10*p_11-p_5*p_11,
```

```
p_1*p_9+p_1-p_5*p_10*p_11-p_5*p_11^2,

p_2^2-p_1*p_10^2*p_11-p_1*p_10*p_11^2-p_1*p_10*p_11-p_1*p_11^2
   +p_2*p_10^2*p_11+p_2*p_10*p_11^2+p_2*p_11^2-p_2*p_11+p_4*p_10^3*p_11
   +p_4*p_10^2*p_11^2-p_6*p_10^2*p_11-p_7*p_10^4*p_11-2*p_7*p_10^3*p_11^2
   -p_7*p_10^2*p_11^3-p_7*p_10^2*p_11,

p_2*p_3+p_4*p_10^3-p_4*p_10^2*p_11-2*p_4*p_10*p_11^2-2*p_4*p_10*p_11
   -2*p_4*p_11^2-p_4*p_10-p_4*p_11-p_6*p_10^2-p_6*p_10*p_11-p_6*p_11+p_6
   -p_7*p_10^4*p_11+p_7*p_10^3*p_11^2+2*p_7*p_10^2*p_11^3-p_7*p_10^3*p_11
   +p_7*p_10^2*p_11^2+2*p_7*p_10*p_11^3-p_7*p_10^2-p_7*p_10*p_11-p_7*p_11
   +p_7+p_8*p_10^5*p_11-p_8*p_10^4*p_11^2-2*p_8*p_10^3*p_11^3
   -3*p_8*p_10^3*p_11-3*p_8*p_10^2*p_11^2-p_9*p_10^4*p_11
   -p_9*p_10^3*p_11^2-p_10^6*p_11+3*p_10^4*p_11^3+2*p_10^3*p_11^4
   -p_10^4*p_11-p_10^3*p_11^2,

p_2*p_4+p_4*p_10^2*p_11+p_4*p_10*p_11^2+p_4*p_10*p_11+p_4*p_11^2
   -p_7*p_10^3*p_11^2-p_7*p_10^2*p_11^3-p_7*p_10^2*p_11^2-p_7*p_10*p_11^3
   +p_8*p_10^4*p_11^2+p_8*p_10^3*p_11^3+p_8*p_10^3*p_11+p_8*p_10^2*p_11^2
   -p_10^5*p_11^2-2*p_10^4*p_11^3-p_10^3*p_11^4,

p_2*p_5+p_2*p_10^2*p_11+p_2*p_10*p_11+p_4*p_10^3+p_4*p_10^2*p_11
   +p_5*p_10^2*p_11+p_5*p_10*p_11^2+p_5*p_10*p_11+p_5*p_11^2
   +p_6*p_10^3*p_11+p_7*p_10^3*p_11,

p_2*p_6+p_4*p_10^2*p_11+p_4*p_10*p_11^2+p_4*p_10*p_11+p_4*p_11^2
   +p_6*p_10^2*p_11+p_6*p_10*p_11^2+p_6*p_10*p_11+p_6*p_11^2
   +p_8*p_10^4*p_11+2*p_8*p_10^3*p_11^2+p_8*p_10^2*p_11^3
   +p_9*p_10^4*p_11^2+p_9*p_10^3*p_11^3+p_9*p_10^3*p_11+p_9*p_10^2*p_11^2,

p_2*p_7-p_6*p_10*p_11-p_6*p_11+p_7*p_10^2*p_11+p_7*p_10*p_11^2+p_7*p_11^2
   -p_7*p_11-p_9*p_10^3*p_11-p_9*p_10^2*p_11^2+p_10^4*p_11^2+p_10^3*p_11^3,

p_2*p_8+p_5*p_10*p_11+p_5*p_11^2,

p_2*p_9-p_1*p_10-p_1*p_11,

p_3^2+p_0*p_10^2+4*p_1*p_10^2-8*p_1*p_10*p_11+p_2*p_10+p_5*p_10^4
   -3*p_5*p_10^3*p_11+4*p_5*p_10*p_11^3,

p_3*p_4+3*p_1*p_10*p_11+p_5*p_10^3*p_11-p_5*p_10^2*p_11^2-2*p_5*p_10*p_11^3,

p_3*p_5+p_4*p_10^3-2*p_4*p_10^2*p_11+p_4*p_10^2-2*p_4*p_10*p_11+p_6*p_10^3
   -2*p_6*p_10^2*p_11-2*p_6*p_10*p_11-p_6*p_10+p_7*p_10^3+p_7*p_10^2*p_11
   +p_7*p_10*p_11-p_7*p_10+p_8*p_10^5-p_8*p_10^4*p_11-2*p_8*p_10^3*p_11^2
```

```
   +p_9*p_10^5*p_11-2*p_9*p_10^4*p_11^2-3*p_9*p_10^3*p_11+p_10^5*p_11
   +p_10^4*p_11^2,

p_3*p_6-p_1*p_10^3+p_1*p_10^2*p_11+2*p_1*p_10*p_11^2,

p_3*p_7-p_2*p_10^2+2*p_2*p_10*p_11,

p_3*p_8+p_6*p_10^2-2*p_6*p_10*p_11+p_7*p_10^2+p_7*p_10*p_11,

p_3*p_9-3*p_4-p_7*p_10^3+p_7*p_10^2*p_11+2*p_7*p_10*p_11^2,

p_4^2-p_1*p_10*p_11+p_5*p_10^2*p_11^2+p_5*p_10*p_11^3,

p_4*p_5+p_4*p_10^2*p_11+p_4*p_10*p_11+p_6*p_10^2*p_11+p_6*p_10*p_11
   +p_8*p_10^4*p_11+p_8*p_10^3*p_11^2+p_9*p_10^4*p_11^2+p_9*p_10^3*p_11,

p_4*p_6-p_1*p_10^2*p_11-p_1*p_10*p_11^2+p_2*p_10*p_11+p_5*p_11,

p_4*p_7-p_2*p_10*p_11-p_5*p_11,

p_4*p_8+p_6*p_10*p_11,

p_4*p_9+p_4-p_7*p_10^2*p_11-p_7*p_10*p_11^2,

p_5^2-p_1*p_10^2-p_1*p_10+p_5*p_10^2*p_11+p_5*p_10*p_11+p_6*p_10^3
   -p_7*p_10^4*p_11,

p_5*p_6+p_4*p_10+p_4+p_6*p_10^2*p_11+p_6*p_10*p_11-p_7*p_10^3*p_11
   -p_7*p_10^2*p_11^2-p_7*p_10^2*p_11-p_7*p_10*p_11^2+p_8*p_10^3*p_11
   +p_8*p_10^2+p_9*p_10^4*p_11+p_9*p_10^3*p_11^2-p_10^5*p_11^2
   -p_10^4*p_11^3-p_10^4*p_11-p_10^3*p_11^2,

p_5*p_7-p_4*p_10-p_4+p_7*p_10^2*p_11+p_7*p_10*p_11-p_8*p_10^3*p_11
   -p_8*p_10^2+p_10^4*p_11+p_10^3*p_11^2,

p_5*p_8-p_1*p_10,

p_5*p_9+p_2*p_10+p_5,

p_6^2-p_1+p_2*p_10^2*p_11+p_2*p_10*p_11^2+2*p_5*p_10*p_11+2*p_5*p_11^2,

p_6*p_7+p_1-p_5*p_10*p_11-p_5*p_11^2,

p_6*p_8+p_4-p_7*p_10^2*p_11-p_7*p_10*p_11^2,

p_6*p_9+p_4*p_10+p_4*p_11+p_6,
```

```
     p_7^2-p_1,

     p_7*p_8-p_4,

     p_7*p_9-p_6,

     p_8^2+p_9*p_10*p_11,

     p_8*p_9+p_8-p_10^2*p_11-p_10*p_11^2,

     p_9^2+p_8*p_10+p_8*p_11+p_9,

     p_0*p_11-p_1*p_10+2*p_1*p_11-p_2,

     p_3*p_11-p_4*p_10+2*p_4*p_11+p_6+p_7},

     QQ[p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_10, p_11],

matrix {{38,  38,  35,  34,  34,  32,  31,  19,  15,  12,   9,   9},
        {33,  24,  24,  30,  21,  24,  21,  12,   9,   9,   9,   0}})
```

The denominators are all elements of P. Since this is an integral extension, it suffices to use the extended euclidean algorithm in $\mathbf{F}(x_2, x_1)[z, y_1, y_2]$ to put $a/b \in A$ into canonical form c/δ for $c \in A$ and $\delta = (x_2 + x_1)x_2x_1^2$.

[Note that it is easy enough to use the P-linear relations to reduce the number of variables needed so that this can indeed be written as a free presentation; but again the cost is not being able to determine memebership directly.]

7 My qth-power algorithm in Magma

My output on this example in Magma in characteristic 2 is:

```
WT_MATRIX_T= [
    [ 34, 34, 31, 29, 26, 23, 19, 15, 12, 9, 9 ],
    [ 30, 21, 21, 24, 24, 15, 12,  9,  9, 9, 0 ]
]
I=
 1 y_34_30*y_9_0+y_34_21*y_9_9+y_34_21*y_9_0+y_31_21+y_19_12
 2 y_12_9^2+y_15_9*y_9_9+y_15_9*y_9_0+y_12_9
 3 y_15_9^2+y_12_9*y_9_9*y_9_0
 4 y_15_9*y_12_9+y_15_9+y_9_9^2*y_9_0+y_9_9*y_9_0^2
 5 y_19_12^2+y_29_24*y_9_0+y_23_15
 6 y_19_12*y_15_9+y_34_21
 7 y_19_12*y_12_9+y_31_21
 8 y_23_15^2+y_23_15*y_9_9*y_9_0+y_23_15*y_9_0+y_19_12*y_9_9^2*y_9_0
 9 y_23_15*y_19_12+y_19_12*y_9_9*y_9_0+y_19_12*y_9_0+y_15_9*y_9_9^2*y_9_0
   +y_12_9*y_9_9^2*y_9_0
10 y_23_15*y_15_9+y_29_24*y_9_0+y_26_24*y_9_0+y_23_15*y_9_0
11 y_23_15*y_12_9+y_26_24*y_9_0+y_23_15*y_9_0+y_23_15
12 y_26_24^2+y_34_30*y_9_9^2+y_29_24*y_9_9^2+y_29_24*y_9_9*y_9_0
   +y_29_24*y_9_9+y_29_24*y_9_0+y_26_24*y_9_9^2+y_26_24*y_9_9*y_9_0
   +y_26_24*y_9_0+y_26_24+y_23_15*y_9_9^2+y_23_15+y_19_12*y_9_9^2*y_9_0
13 y_26_24*y_23_15+y_31_21*y_9_9^2+y_26_24*y_9_9*y_9_0+y_26_24*y_9_0
   +y_19_12*y_9_9^2*y_9_0+y_19_12*y_9_9^2
14 y_26_24*y_19_12+y_31_21*y_9_9+y_31_21+y_19_12*y_9_9*y_9_0
   +y_19_12*y_9_9+y_19_12*y_9_0+y_19_12+y_15_9*y_9_9^3
   +y_12_9*y_9_9^2*y_9_0+y_9_9^4*y_9_0+y_9_9^3*y_9_0^2
15 y_26_24*y_15_9+y_29_24*y_9_0+y_26_24*y_9_0+y_23_15*y_9_9^2
   +y_23_15*y_9_9*y_9_0+y_23_15*y_9_0
16 y_26_24*y_12_9+y_29_24*y_9_9+y_29_24*y_9_0+y_26_24*y_9_9
   +y_23_15*y_9_9+y_23_15
17 y_29_24^2+y_34_30*y_9_9^2+y_31_21*y_9_9^3+y_29_24*y_9_9^2
   +y_29_24*y_9_9*y_9_0+y_29_24*y_9_9+y_29_24*y_9_0+y_26_24*y_9_9^2*y_9_0
   +y_26_24*y_9_9^2+y_26_24*y_9_0+y_26_24+y_ 23_15*y_9_9^2*y_9_0
   +y_23_15*y_9_9+y_23_15*y_9_0+y_23_15
18 y_29_24*y_26_24+y_34_30*y_9_9^2+y_34_21*y_9_9^2+y_31_21*y_9_9^2
   +y_29_24*y_9_9^2+y_29_24*y_9_9+y_26_24*y_9_9^2+y_26_24*y_9_9*y_9_0
   +y_26_24*y_9_0+y_26_24+y_23_15*y_9_9^3+y_23_15*y_9_9^2*y_9_0
   +y_23_15*y_9_0+y_23_15+y_19_12*y_9_9^4+y_19_12*y_9_9^3*y_9_0
   +y_19_12*y_9_9^2
19 y_29_24*y_23_15+y_34_21*y_9_9^2+y_31_21*y_9_9^2+y_29_24*y_9_9*y_9_0
   +y_29_24*y_9_0+y_19_12*y_9_9^2
20 y_29_24*y_19_12+y_34_21*y_9_9+y_34_21+y_31_21*y_9_9+y_31_21
   +y_19_12*y_9_9+y_19_12+y_15_9*y_9_9^3+y_15_9*y_9_9^2*y_9_0
```

```
    +y_15_9*y_9_9^2+y_12_9*y_9_9^3*y_9_0
21  y_29_24*y_15_9+y_26_24*y_9_9*y_9_0+y_23_15*y_9_9^2+y_23_15*y_9_9
22  y_29_24*y_12_9+y_29_24*y_9_9+y_29_24*y_9_0+y_29_24+y_26_24*y_9_9
    +y_26_24*y_9_0+y_26_24+y_23_15*y_9_9^2+y_23_15*y_9_9*y_9_0
    +y_23_15*y_9_9+y_23_15*y_9_0+y_23_15
23  y_31_21^2+y_29_24*y_9_0+y_26_24*y_9_9^2*y_9_0^2+y_26_24*y_9_9*y_9_0^3
    +y_23_15*y_9_9^3*y_9_0+y_23_15*y_9_9^2*y_9_0^2+y_23_15
24  y_31_21*y_29_24+y_34_21*y_9_9^2+y_34_21*y_9_9*y_9_0+y_34_21*y_9_0
    +y_34_21+y_19_12*y_9_9^3*y_9_0+y_19_12*y_9_9^2*y_9_0^2
    +y_19_12*y_9_9^2*y_9_0+y_19_12*y_9_9*y_9_0^2+y_15_9*y_9_9^4*y_9_0
    +y_15_9*y_9_9^3*y_9_0^2+y_15_9*y_9_9^3+y_15_9*y_9_9^2*y_9_0
    +y_15_9*y_9_9^2+y_12_9*y_9_9^3*y_9_0+y_9_9^5*y_9_0+y_9_9^3*y_9_0^3
    +y_9_9^4*y_9_0+y_9_9^3*y_9_0^2
25  y_31_21*y_26_24+y_34_21*y_9_9^2+y_34_21*y_9_9*y_9_0+y_34_21*y_9_9
    +y_34_21*y_9_0+y_31_21*y_9_9*y_9_0+y_31_21*y_9_0+y_15_9*y_9_9^3*y_9_0
    +y_15_9*y_9_9^2*y_9_0^2+y_15_9*y_9_9^3+y_12_9*y_9_9^4*y_9_0
    +y_12_9*y_9_9^3*y_9_0^2+y_12_9*y_9_9^2*y_9_0+y_9_9^5*y_9_0
    +y_9_9^4*y_9_0^2
26  y_31_21*y_23_15+y_31_21*y_9_9*y_9_0+y_31_21*y_9_0+y_15_9*y_9_9^3*y_9_0
    +y_15_9*y_9_9^2*y_9_0^2+y_15_9*y_9_9^2*y_9_0+y_12_9*y_9_9^2*y_9_0
    +y_9_9^4*y_9_0^2+y_9_9^3*y_9_0^3
27  y_31_21*y_19_12+y_29_24*y_9_9*y_9_0+y_29_24*y_9_0^2+y_29_24*y_9_0
    +y_26_24*y_9_9*y_9_0+y_26_24*y_9_0^2+y_23_15*y_9_9^2*y_9_0
    +y_23_15*y_9_9*y_9_0^2+y_23_15*y_9_9*y_9_0+y_23_15*y_9_0^2+y_23_15
28  y_31_21*y_15_9+y_34_21+y_19_12*y_9_9^2*y_9_0+y_19_12*y_9_9*y_9_0^2
29  y_31_21*y_12_9+y_34_21*y_9_9+y_34_21*y_9_0+y_31_21
30  y_34_21^2+y_29_24*y_9_9^2*y_9_0^2+y_29_24*y_9_9*y_9_0^3
    +y_29_24*y_9_9*y_9_0^2+y_26_24*y_9_9^2*y_9_0^2+y_26_24*y_9_9*y_9_0^3
    +y_23_15*y_9_9^3*y_9_0^2+y_23_15*y_9_9^2*y_9_0^3+y_23_15*y_9_9^2*y_9_0^2
    +y_23_15*y_9_9*y_9_0^3+y_23_15*y_9_9*y_9_0
31  y_34_21*y_31_21+y_29_24*y_9_9^2*y_9_0^2+y_29_24*y_9_9*y_9_0^3
    +y_29_24*y_9_0+y_26_24*y_9_9*y_9_0^2+y_26_24*y_9_0
    +y_23_15*y_9_9*y_9_0^2+y_23_15*y_9_9*y_9_0+y_23_15*y_9_0
32  y_34_21*y_29_24+y_31_21*y_9_9^2*y_9_0+y_31_21*y_9_9*y_9_0
    +y_19_12*y_9_9^3*y_9_0+y_19_12*y_9_9^2*y_9_0^2+y_19_12*y_9_9^2*y_9_0
    +y_19_12*y_9_9*y_9_0^2+y_15_9*y_9_9^3*y_9_0+y_12_9*y_9_9^4*y_9_0
    +y_12_9*y_9_9^3*y_9_0^2+y_12_9*y_9_9^3*y_9_0+y_9_9^5*y_9_0^2
    +y_9_9^4*y_9_0^3
33  y_34_21*y_26_24+y_34_21*y_9_9*y_9_0+y_34_21*y_9_0+y_19_12*y_9_9^3*y_9_0
    +y_19_12*y_9_9^2*y_9_0^2+y_19_12*y_9_9^2*y_9_0+y_19_12*y_9_9*y_9_0^2
    +y_15_9*y_9_9^4*y_9_0+y_15_9*y_9_9^3*y_9_0^2+y_15_9*y_9_9^2*y_9_0
    +y_12_9*y_9_9^4*y_9_0+y_9_9^4*y_9_0^2+y_9_9^3*y_9_0^3
34  y_34_21*y_23_15+y_34_21*y_9_9*y_9_0+y_34_21*y_9_0+y_15_9*y_9_9^2*y_9_0
    +y_12_9*y_9_9^3*y_9_0^2+y_9_9^4*y_9_0^2+y_9_9^3*y_9_0^3
35  y_34_21*y_19_12+y_29_24*y_9_0+y_26_24*y_9_9*y_9_0^2+y_26_24*y_9_0
    +y_23_15*y_9_9^2*y_9_0+y_23_15*y_9_9*y_9_0+y_23_15*y_9_0
```

```
36 y_34_21*y_15_9+y_31_21*y_9_9*y_9_0
37 y_34_21*y_12_9+y_34_21+y_19_12*y_9_9^2*y_9_0+y_19_12*y_9_9*y_9_0^2
38 y_34_30^2+y_29_24*y_9_9^4+y_29_24*y_9_9^3*y_9_0+y_29_24*y_9_9^2*y_9_0^2
   +y_29_24*y_9_9*y_9_0^3+y_29_24*y_9_9^3+y_29_24*y_9_9*y_9_0^2
   +y_26_24*y_9_9^4+y_26_24*y_9_9^3*y_9_0+y_26_24*y_9_9^2*y_9_0^2
   +y_26_24*y_9_9*y_9_0^3+y_26_24*y_9_9^2+y_26_24*y_9_9*y_9_0
   +y_23_15*y_9_9^5+y_23_15*y_9_9^4*y_9_0+y_23_15*y_9_9^3*y_9_0^2
   +y_23_15*y_9_9^2*y_9_0^3+y_23_15*y_9_9^4+y_23_15*y_9_9^3*y_9_0
   +y_23_15*y_9_9^2*y_9_0^2+y_23_15*y_9_9*y_9_0^3+y_23_15*y_9_9^2
   +y_23_15*y_9_9*y_9_0
39 y_34_30*y_34_21+y_29_24*y_9_9^3*y_9_0+y_29_24*y_9_9*y_9_0^3
   +y_26_24*y_9_9^3*y_9_0+y_26_24*y_9_9*y_9_0^3+y_23_15*y_9_9^4*y_9_0
   +y_23_15*y_9_9^2*y_9_0^3+y_23_15*y_9_9^3*y_9_0+y_23_15*y_9_9*y_9_0^3
40 y_34_30*y_31_21+y_29_24*y_9_9^3*y_9_0+y_29_24*y_9_9*y_9_0^3
   +y_23_15*y_9_9^3+y_23_15*y_9_9*y_9_0^2
41 y_34_30*y_29_24+y_34_30*y_9_9+y_34_30+y_31_21*y_9_9^3
   +y_31_21*y_9_9^2*y_9_0+y_31_21*y_9_9^2+y_31_21*y_9_9*y_9_0
   +y_19_12*y_9_9^4+y_19_12*y_9_9^2*y_9_0^2+y_19_12*y_9_9^2*y_9_0
   +y_19_12*y_9_9*y_9_0^2+y_19_12*y_9_9^2+y_19_12*y_9_9*y_9_0
   +y_12_9*y_9_9^5+y_12_9*y_9_9^3*y_9_0^2+y_12_9*y_9_9^4
   +y_12_9*y_9_9^3*y_9_0+y_9_9^6*y_9_0+y_9_9^4*y_9_0^3+y_9_9^5
   +y_9_9^3*y_9_0^2+y_9_9^4+y_9_9^3*y_9_0
42 y_34_30*y_26_24+y_34_30*y_9_9+y_34_30+y_34_21*y_9_9^2
   +y_34_21*y_9_9*y_9_0+y_34_21*y_9_9+y_34_21*y_9_0+y_31_21*y_9_9
   +y_31_21+y_19_12*y_9_9^4+y_19_12*y_9_9^2*y_9_0^2+y_19_12*y_9_9^3
   +y_19_12*y_9_9*y_9_0^2+y_19_12*y_9_9+y_19_12+y_15_9*y_9_9^5
   +y_15_9*y_9_9^3*y_9_0^2+y_12_9*y_9_9^5+y_12_9*y_9_9^4*y_9_0
   +y_12_9*y_9_9^4+y_12_9*y_9_9^3*y_9_0+y_9_9^5*y_9_0+y_9_9^3*y_9_0^3
   +y_9_9^5+y_9_9^4*y_9_0+y_9_9^4+y_9_9^3*y_9_0
43 y_34_30*y_23_15+y_34_21*y_9_9^2+y_34_21*y_9_9*y_9_0+y_34_21*y_9_9
   +y_34_21*y_9_0+y_31_21*y_9_9+y_31_21+y_19_12*y_9_9+y_19_12
   +y_12_9*y_9_9^4*y_9_0+y_12_9*y_9_9^3*y_9_0^2+y_9_9^5*y_9_0
   +y_9_9^3*y_9_0^3+y_9_9^4*y_9_0+y_9_9^3*y_9_0^2
44 y_34_30*y_19_12+y_26_24*y_9_9^2*y_9_0+y_26_24*y_9_9*y_9_0^2
   +y_23_15*y_9_9^3+y_23_15*y_9_9^2*y_9_0
45 y_34_30*y_15_9+y_31_21*y_9_9^2+y_31_21*y_9_9*y_9_0+y_19_12*y_9_9^2
   +y_19_12*y_9_9*y_9_0
46 y_34_30*y_12_9+y_19_12*y_9_9^3+y_19_12*y_9_9*y_9_0^2

delta= x_9_9^3*x_9_0^3+x_9_9^2*x_9_0^4+x_9_9^3*x_9_0^2+x_9_9*x_9_0^4+
    x_9_9^2*x_9_0^2+x_9_9*x_9_0^3

phi=
1 x_19_12*x_15_9*x_9_9^4*x_9_0^2+x_19_12*x_15_9*x_9_9^2*x_9_0^4+
    x_19_12*x_15_9*x_9_9^4*x_9_0+x_19_12*x_15_9*x_9_9^3*x_9_0^2+
    x_19_12*x_15_9*x_9_9^2*x_9_0^3+x_19_12*x_15_9*x_9_9*x_9_0^4+
```

```
    x_19_12*x_12_9*x_9_9^3*x_9_0^2+x_19_12*x_12_9*x_9_9^2*x_9_0^3+
    x_19_12*x_15_9*x_9_9^3*x_9_0+x_19_12*x_15_9*x_9_9*x_9_0^3+
    x_19_12*x_12_9*x_9_9^3*x_9_0+x_19_12*x_12_9*x_9_9*x_9_0^3+
    x_19_12*x_9_9^3*x_9_0^2+x_19_12*x_9_9^2*x_9_0^3+
    x_19_12*x_12_9*x_9_9^2*x_9_0+x_19_12*x_12_9*x_9_9*x_9_0^2+
    x_19_12*x_9_9^3*x_9_0+x_19_12*x_9_9*x_9_0^3+x_19_12*x_9_9^2*x_9_0+
    x_19_12*x_9_9*x_9_0^2
2 x_19_12*x_15_9*x_9_9^3*x_9_0^3+x_19_12*x_15_9*x_9_9^2*x_9_0^4+
    x_19_12*x_15_9*x_9_9^3*x_9_0^2+x_19_12*x_15_9*x_9_9*x_9_0^4+
    x_19_12*x_15_9*x_9_9^2*x_9_0^2+x_19_12*x_15_9*x_9_9*x_9_0^3
3 x_19_12*x_12_9*x_9_9^3*x_9_0^3+x_19_12*x_12_9*x_9_9^2*x_9_0^4+
    x_19_12*x_12_9*x_9_9^3*x_9_0^2+x_19_12*x_12_9*x_9_9*x_9_0^4+
    x_19_12*x_12_9*x_9_9^2*x_9_0^2+x_19_12*x_12_9*x_9_9*x_9_0^3
4 x_19_12^2*x_9_9^3*x_9_0^2+x_19_12^2*x_9_9^2*x_9_0^3+
    x_19_12^2*x_9_9^3*x_9_0+x_19_12^2*x_9_9*x_9_0^3+
    x_19_12^2*x_12_9*x_9_9*x_9_0+x_19_12^2*x_15_9*x_9_9+
    x_19_12^2*x_15_9*x_9_0+x_19_12^2*x_9_9*x_9_0+x_19_12^2*x_12_9+
    x_19_12^2
5 x_19_12^2*x_15_9*x_9_9^2*x_9_0+x_19_12^2*x_15_9*x_9_9*x_9_0^2+
    x_19_12^2*x_12_9*x_9_9^2*x_9_0+x_19_12^2*x_9_9^3*x_9_0+
    x_19_12^2*x_9_9^2*x_9_0^2+x_19_12^2*x_15_9*x_9_9*x_9_0+
    x_19_12^2*x_15_9*x_9_0^2+x_19_12^2*x_12_9*x_9_9*x_9_0+
    x_19_12^2*x_9_9^2*x_9_0+x_19_12^2*x_15_9*x_9_9+x_19_12^2*x_15_9*x_9_0+
    x_19_12^2*x_12_9*x_9_0+x_19_12^2*x_9_9*x_9_0+x_19_12^2*x_12_9+
    x_19_12^2*x_9_0+x_19_12^2
6 x_19_12^2*x_12_9*x_9_9*x_9_0^2+x_19_12^2*x_9_9^2*x_9_0^2+
    x_19_12^2*x_9_9*x_9_0^3+x_19_12^2*x_15_9*x_9_9*x_9_0+
    x_19_12^2*x_15_9*x_9_0^2+x_19_12^2*x_9_9*x_9_0^2+x_19_12^2*x_12_9*x_9_0
   +x_19_12^2*x_9_0
7 x_19_12*x_9_9^3*x_9_0^3+x_19_12*x_9_9^2*x_9_0^4+x_19_12*x_9_9^3*x_9_0^2+
    x_19_12*x_9_9*x_9_0^4+x_19_12*x_9_9^2*x_9_0^2+x_19_12*x_9_9*x_9_0^3
8 x_15_9*x_9_9^3*x_9_0^3+x_15_9*x_9_9^2*x_9_0^4+x_15_9*x_9_9^3*x_9_0^2+
    x_15_9*x_9_9*x_9_0^4+x_15_9*x_9_9^2*x_9_0^2+x_15_9*x_9_9*x_9_0^3
9 x_12_9*x_9_9^3*x_9_0^3+x_12_9*x_9_9^2*x_9_0^4+x_12_9*x_9_9^3*x_9_0^2+
    x_12_9*x_9_9*x_9_0^4+x_12_9*x_9_9^2*x_9_0^2+x_12_9*x_9_9*x_9_0^3
10 x_9_9^4*x_9_0^3+x_9_9^3*x_9_0^4+x_9_9^4*x_9_0^2+x_9_9^2*x_9_0^4+
    x_9_9^3*x_9_0^2+x_9_9^2*x_9_0^3
11 x_9_9^3*x_9_0^4+x_9_9^2*x_9_0^5+x_9_9^3*x_9_0^3+x_9_9*x_9_0^5+
    x_9_9^2*x_9_0^3+x_9_9*x_9_0^4

psi=
1 f_19_12
2 f_15_9
3 f_12_9
4 f_9_9
5 f_9_0
```

At least for output purposes it was easier here to assign proper names to all of the variables; that is, ones that reflect the weights involved.

```
loading file "intclos_input_19_15_12.mag"
Loading "intclos_func70109.mag"
-----------------------------------------
I:
1 y_34_30*y_9_0 + 22*y_34_21*y_9_9 + y_34_21*y_9_0 + y_31_21 + 22*y_19_12
2 y_38_33*y_9_0 + 22*y_38_24*y_9_9 + y_38_24*y_9_0 + y_35_24
3 y_12_9^2 + 22*y_15_9*y_9_9 + 22*y_15_9*y_9_0 + 22*y_12_9
4 y_15_9^2 + 22*y_12_9*y_9_9*y_9_0
5 y_15_9*y_12_9 + 22*y_9_9^2*y_9_0 + 22*y_9_9*y_9_0^2 + 22*y_15_9
6 y_19_12^2 + 22*y_38_24
7 y_19_12*y_15_9 + 22*y_34_21
8 y_19_12*y_12_9 + 22*y_31_21
9 y_31_21^2 + 22*y_35_24*y_9_9^2*y_9_0 + 22*y_35_24*y_9_9*y_9_0^2 +
    21*y_32_24*y_9_9*y_9_0 + 21*y_32_24*y_9_0^2 + 22*y_38_24
10 y_31_21*y_19_12 + 22*y_32_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0^2 + 22*y_38_24
11 y_31_21*y_15_9 + 22*y_19_12*y_9_9^2*y_9_0 + 22*y_19_12*y_9_9*y_9_0^2 +
    22*y_34_21
12 y_31_21*y_12_9 + 22*y_34_21*y_9_9 + 22*y_34_21*y_9_0 + 22*y_31_21
13 y_32_24^2 + 22*y_19_12*y_9_9^4*y_9_0 + 22*y_32_24*y_9_9^2*y_9_0 +
    22*y_31_21*y_9_9^3 + 22*y_38_24*y_9_9^2 + 22*y_32_24*y_9_9*y_9_0 +
    22*y_38_24*y_9_9
14 y_32_24*y_31_21 + 22*y_9_9^5*y_9_0^2 + 22*y_9_9^4*y_9_0^3 +
    22*y_31_21*y_9_9^2*y_9_0 + 22*y_12_9*y_9_9^4*y_9_0 +
    22*y_12_9*y_9_9^3*y_9_0^2 + 22*y_19_12*y_9_9^3*y_9_0 +
    22*y_19_12*y_9_9^2*y_9_0^2 + 22*y_15_9*y_9_9^3*y_9_0 +
    22*y_31_21*y_9_9*y_9_0 + 22*y_19_12*y_9_9^2*y_9_0 + 22*y_19_12*y_9_9*y_9_0^2
    + 22*y_9_9^4*y_9_0 + 22*y_9_9^3*y_9_0^2 + 22*y_34_21*y_9_9 + 22*y_34_21 +
    22*y_15_9*y_9_9^2
15 y_32_24*y_19_12 + 22*y_15_9*y_9_9^3*y_9_0 + 22*y_19_12*y_9_9^2*y_9_0 +
    22*y_9_9^4*y_9_0 + 22*y_9_9^3*y_9_0^2 + 22*y_34_21*y_9_9 +
    22*y_19_12*y_9_9*y_9_0 + 22*y_34_21 + 22*y_15_9*y_9_9^2
16 y_32_24*y_15_9 + 22*y_38_24*y_9_9
17 y_32_24*y_12_9 + 22*y_35_24*y_9_9 + 22*y_32_24
18 y_34_21^2 + 22*y_32_24*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9*y_9_0^3 +
    22*y_38_24*y_9_9*y_9_0
19 y_34_21*y_32_24 + 22*y_12_9*y_9_9^4*y_9_0^2 + 22*y_34_21*y_9_9^2*y_9_0 +
    22*y_15_9*y_9_9^4*y_9_0 + 22*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_31_21*y_9_9^2*y_9_0 + 22*y_34_21*y_9_9*y_9_0 + 22*y_31_21*y_9_9*y_9_0 +
    22*y_12_9*y_9_9^3*y_9_0
20 y_34_21*y_31_21 + 22*y_38_24*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0^2 +
    22*y_35_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0
21 y_34_21*y_19_12 + 22*y_35_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0
22 y_34_21*y_15_9 + 22*y_31_21*y_9_9*y_9_0
23 y_34_21*y_12_9 + 22*y_19_12*y_9_9^2*y_9_0 + 22*y_19_12*y_9_9*y_9_0^2 +
    22*y_34_21
24 y_34_30^2 + 22*y_32_24*y_9_9^4 + y_32_24*y_9_9^3*y_9_0 +
```

```
    y_32_24*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9*y_9_0^3 + y_38_33*y_9_9^2 +
    3*y_38_24*y_9_9^2 + 20*y_38_24*y_9_9*y_9_0 + 22*y_35_24*y_9_9
25 y_34_30*y_34_21 + 22*y_32_24*y_9_9^3*y_9_0 + y_32_24*y_9_9*y_9_0^3 +
    2*y_38_24*y_9_9*y_9_0
26 y_34_30*y_32_24 + 22*y_12_9*y_9_9^5*y_9_0 + y_12_9*y_9_9^4*y_9_0^2 +
    22*y_34_21*y_9_9^3 + y_34_21*y_9_9^2*y_9_0 + 22*y_15_9*y_9_9^5 +
    y_15_9*y_9_9^3*y_9_0^2 + 22*y_31_21*y_9_9^3 + y_31_21*y_9_9^2*y_9_0 +
    y_9_9^5*y_9_0 + y_9_9^4*y_9_0^2 + 22*y_34_21*y_9_9^2 + y_34_21*y_9_9*y_9_0 +
    y_31_21*y_9_9*y_9_0 + 2*y_12_9*y_9_9^3*y_9_0 + y_19_12*y_9_9^3 +
    y_19_12*y_9_9^2*y_9_0 + y_31_21*y_9_9 + y_19_12*y_9_9*y_9_0 +
    22*y_19_12*y_9_9
27 y_34_30*y_31_21 + 22*y_38_24*y_9_9^3 + y_38_24*y_9_9*y_9_0^2 +
    2*y_35_24*y_9_9*y_9_0 + 2*y_32_24*y_9_0
28 y_34_30*y_19_12 + 22*y_35_24*y_9_9^2 + y_35_24*y_9_9*y_9_0 + 2*y_32_24*y_9_0
29 y_34_30*y_15_9 + 22*y_31_21*y_9_9^2 + y_31_21*y_9_9*y_9_0 + y_19_12*y_9_9^2 +
    y_19_12*y_9_9*y_9_0
30 y_34_30*y_12_9 + 22*y_19_12*y_9_9^3 + y_19_12*y_9_9*y_9_0^2 + 2*y_34_21
31 y_35_24^2 + 22*y_34_21*y_9_9^3*y_9_0 + 22*y_34_21*y_9_9^2*y_9_0^2 +
    22*y_38_24*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0^2 +
    22*y_19_12*y_9_9^4*y_9_0 + 21*y_19_12*y_9_9^3*y_9_0^2 +
    22*y_19_12*y_9_9^2*y_9_0^3 + 22*y_35_24*y_9_9^2*y_9_0 +
    22*y_35_24*y_9_9*y_9_0^2 + y_31_21*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0 +
    22*y_38_24*y_9_0^2 + 22*y_35_24*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0 +
    y_35_24*y_9_0
32 y_35_24*y_34_30 + 22*y_15_9*y_9_9^5*y_9_0 + y_15_9*y_9_9^3*y_9_0^3 +
    22*y_19_12*y_9_9^4*y_9_0 + y_19_12*y_9_9^2*y_9_0^3 + 22*y_9_9^6*y_9_0 +
    22*y_9_9^5*y_9_0^2 + y_9_9^4*y_9_0^3 + y_9_9^3*y_9_0^4 + 22*y_34_21*y_9_9^3
    + y_34_21*y_9_9*y_9_0^2 + y_12_9*y_9_9^4*y_9_0 + y_12_9*y_9_9^3*y_9_0^2 +
    22*y_19_12*y_9_9^3*y_9_0 + y_19_12*y_9_9*y_9_0^3 + y_34_21*y_9_9*y_9_0 +
    y_34_21*y_9_0^2 + 2*y_15_9*y_9_9^3*y_9_0 + 2*y_15_9*y_9_9^2*y_9_0^2 +
    y_31_21*y_9_9^2 + y_31_21*y_9_9*y_9_0 + 22*y_9_9^4*y_9_0 +
    22*y_9_9^3*y_9_0^2 + y_34_21*y_9_9 + y_34_21*y_9_0 + y_31_21*y_9_0 +
    22*y_19_12*y_9_9^2 + 22*y_19_12*y_9_9*y_9_0 + 22*y_31_21 + 22*y_19_12*y_9_0
    + y_19_12
33 y_35_24*y_34_21 + 22*y_15_9*y_9_9^4*y_9_0^2 + 22*y_15_9*y_9_9^3*y_9_0^3 +
    22*y_19_12*y_9_9^3*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0^3 + 22*y_9_9^5*y_9_0^2
    + 21*y_9_9^4*y_9_0^3 + 22*y_9_9^3*y_9_0^4 + 22*y_34_21*y_9_9^2*y_9_0 +
    22*y_34_21*y_9_9*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0^2 +
    22*y_19_12*y_9_9*y_9_0^3 + 22*y_34_21*y_9_9*y_9_0 + 22*y_34_21*y_9_0^2 +
    22*y_15_9*y_9_9^3*y_9_0 + 22*y_15_9*y_9_9^2*y_9_0^2
34 y_35_24*y_32_24 + 22*y_31_21*y_9_9^3*y_9_0 + 22*y_35_24*y_9_9^2*y_9_0 +
    22*y_34_21*y_9_9^3 + 22*y_34_21*y_9_9^2*y_9_0 + 22*y_32_24*y_9_9^2*y_9_0 +
    22*y_32_24*y_9_9*y_9_0^2 + y_19_12*y_9_9^3*y_9_0 + 22*y_35_24*y_9_9*y_9_0 +
    22*y_32_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0^2
35 y_35_24*y_31_21 + 22*y_12_9*y_9_9^4*y_9_0^2 + 22*y_12_9*y_9_9^3*y_9_0^3 +
    22*y_34_21*y_9_9^2*y_9_0 + 22*y_34_21*y_9_9*y_9_0^2 +
```

```
    22*y_15_9*y_9_9^4*y_9_0 + 21*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_15_9*y_9_9^2*y_9_0^3 + 22*y_31_21*y_9_9^2*y_9_0 +
    22*y_31_21*y_9_9*y_9_0^2 + 22*y_34_21*y_9_9*y_9_0 + 22*y_34_21*y_9_0^2 +
    22*y_31_21*y_9_9*y_9_0 + 22*y_31_21*y_9_0^2 + 22*y_12_9*y_9_9^3*y_9_0 +
    22*y_12_9*y_9_9^2*y_9_0^2
36 y_35_24*y_19_12 + 22*y_9_9^4*y_9_0^2 + 22*y_9_9^3*y_9_0^3 +
    22*y_31_21*y_9_9*y_9_0 + 22*y_12_9*y_9_9^3*y_9_0 + 22*y_12_9*y_9_9^2*y_9_0^2
    + 22*y_19_12*y_9_9^2*y_9_0 + 22*y_19_12*y_9_9*y_9_0^2 + 22*y_31_21*y_9_0 +
    22*y_19_12*y_9_0^2 + y_19_12*y_9_0
37 y_35_24*y_15_9 + 22*y_32_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0^2
38 y_35_24*y_12_9 + 22*y_38_24*y_9_9 + 22*y_38_24*y_9_0
39 y_38_24^2 + 22*y_31_21*y_9_9^3*y_9_0^2 + 22*y_35_24*y_9_9^2*y_9_0^2 +
    22*y_34_21*y_9_9^3*y_9_0 + 22*y_34_21*y_9_9^2*y_9_0^2 +
    22*y_32_24*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9*y_9_0^3 +
    22*y_35_24*y_9_9*y_9_0^2 + 21*y_32_24*y_9_9*y_9_0^2 + 22*y_32_24*y_9_0^3 +
    22*y_31_21*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0^2 +
    22*y_38_24*y_9_0
40 y_38_24*y_35_24 + 22*y_19_12*y_9_9^4*y_9_0^2 + 22*y_19_12*y_9_9^3*y_9_0^3 +
    22*y_32_24*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9*y_9_0^3 +
    22*y_31_21*y_9_9^3*y_9_0 + 22*y_31_21*y_9_9^2*y_9_0^2 +
    22*y_38_24*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0^2 +
    22*y_32_24*y_9_9*y_9_0^2 + 22*y_32_24*y_9_0^3 + 22*y_38_24*y_9_9*y_9_0 +
    22*y_38_24*y_9_0^2
41 y_38_24*y_34_30 + 22*y_9_9^6*y_9_0^2 + y_9_9^4*y_9_0^4 +
    22*y_31_21*y_9_9^3*y_9_0 + y_31_21*y_9_9^2*y_9_0^2 + 22*y_12_9*y_9_9^5*y_9_0
    + y_12_9*y_9_9^3*y_9_0^3 + 22*y_19_12*y_9_9^4*y_9_0 +
    y_19_12*y_9_9^2*y_9_0^3 + 2*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_31_21*y_9_9^2*y_9_0 + y_31_21*y_9_9*y_9_0^2 + y_19_12*y_9_9^2*y_9_0^2 +
    y_19_12*y_9_9*y_9_0^3 + 2*y_9_9^4*y_9_0^2 + 2*y_9_9^3*y_9_0^3 +
    2*y_34_21*y_9_9*y_9_0 + y_19_12*y_9_9^2*y_9_0 + y_19_12*y_9_9*y_9_0^2 +
    2*y_34_21*y_9_0 + 2*y_15_9*y_9_9^2*y_9_0
42 y_38_24*y_34_21 + 22*y_9_9^5*y_9_0^3 + 22*y_9_9^4*y_9_0^4 +
    22*y_31_21*y_9_9^2*y_9_0^2 + 22*y_12_9*y_9_9^4*y_9_0^2 +
    22*y_12_9*y_9_9^3*y_9_0^3 + 22*y_19_12*y_9_9^3*y_9_0^2 +
    22*y_19_12*y_9_9^2*y_9_0^3 + 22*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_31_21*y_9_9*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0^2 +
    22*y_19_12*y_9_9*y_9_0^3 + 22*y_9_9^4*y_9_0^2 + 22*y_9_9^3*y_9_0^3 +
    22*y_34_21*y_9_9*y_9_0 + 22*y_34_21*y_9_0 + 22*y_15_9*y_9_9^2*y_9_0
43 y_38_24*y_32_24 + 22*y_34_21*y_9_9^3*y_9_0 + 22*y_38_24*y_9_9^2*y_9_0 +
    22*y_19_12*y_9_9^4*y_9_0 + 22*y_19_12*y_9_9^3*y_9_0^2 +
    22*y_35_24*y_9_9^2*y_9_0 + 22*y_38_24*y_9_9*y_9_0 + 22*y_35_24*y_9_9*y_9_0 +
    22*y_34_21*y_9_9^2 + 22*y_32_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0
44 y_38_24*y_31_21 + 22*y_15_9*y_9_9^4*y_9_0^2 + 22*y_15_9*y_9_9^3*y_9_0^3 +
    22*y_19_12*y_9_9^3*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0^3 + 22*y_9_9^5*y_9_0^2
    + 21*y_9_9^4*y_9_0^3 + 22*y_9_9^3*y_9_0^4 + 22*y_34_21*y_9_9^2*y_9_0 +
    22*y_34_21*y_9_9*y_9_0^2 + 22*y_12_9*y_9_9^3*y_9_0^2 +
```

```
    22*y_19_12*y_9_9^2*y_9_0^2 + 22*y_19_12*y_9_9*y_9_0^3 +
    21*y_34_21*y_9_9*y_9_0 + 22*y_34_21*y_9_0^2 + 21*y_15_9*y_9_9^3*y_9_0 +
    21*y_15_9*y_9_9^2*y_9_0^2 + 22*y_31_21*y_9_9*y_9_0 + 22*y_34_21*y_9_0 +
    22*y_31_21*y_9_0 + 22*y_12_9*y_9_9^2*y_9_0
45 y_38_24*y_19_12 + 22*y_12_9*y_9_9^3*y_9_0^2 + 22*y_34_21*y_9_9*y_9_0 +
    22*y_15_9*y_9_9^3*y_9_0 + 22*y_15_9*y_9_9^2*y_9_0^2 + 22*y_31_21*y_9_9*y_9_0
    + 22*y_34_21*y_9_0 + 22*y_31_21*y_9_0 + 22*y_12_9*y_9_9^2*y_9_0
46 y_38_24*y_15_9 + 22*y_35_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0
47 y_38_24*y_12_9 + 22*y_32_24*y_9_9*y_9_0 + 22*y_32_24*y_9_0^2 + 22*y_38_24
48 y_38_33^2 + 22*y_31_21*y_9_9^5 + 2*y_31_21*y_9_9^4*y_9_0 +
    22*y_31_21*y_9_9^3*y_9_0^2 + 22*y_35_24*y_9_9^4 + 2*y_35_24*y_9_9^3*y_9_0 +
    22*y_35_24*y_9_9^2*y_9_0^2 + 22*y_34_30*y_9_9^4 + y_34_21*y_9_9^3*y_9_0 +
    22*y_34_21*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9^4 + y_32_24*y_9_9^3*y_9_0 +
    y_32_24*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9*y_9_0^3 + 2*y_19_12*y_9_9^5 +
    21*y_19_12*y_9_9^3*y_9_0^2 + 22*y_35_24*y_9_9^3 + 2*y_35_24*y_9_9^2*y_9_0 +
    22*y_35_24*y_9_9*y_9_0^2 + 3*y_32_24*y_9_9^2*y_9_0 +
    21*y_32_24*y_9_9*y_9_0^2 + 22*y_32_24*y_9_0^3 + 2*y_31_21*y_9_9^3 +
    20*y_31_21*y_9_9^2*y_9_0 + y_38_33*y_9_9^2 + 3*y_38_24*y_9_9^2 +
    20*y_38_24*y_9_9*y_9_0 + 22*y_34_30*y_9_9^2 + 21*y_34_21*y_9_9^2 +
    y_32_24*y_9_9^2 + 2*y_32_24*y_9_9*y_9_0 + 20*y_32_24*y_9_0^2 + y_38_24*y_9_9
    + 20*y_38_24*y_9_0 + 21*y_19_12*y_9_9^3 + 22*y_19_12*y_9_9^2*y_9_0 +
    22*y_35_24*y_9_9 + 22*y_38_33 + 21*y_38_24 + 22*y_35_24
49 y_38_33*y_38_24 + 22*y_31_21*y_9_9^4*y_9_0 + y_31_21*y_9_9^3*y_9_0^2 +
    22*y_35_24*y_9_9^3*y_9_0 + y_35_24*y_9_9^2*y_9_0^2 + 22*y_34_21*y_9_9^4 +
    y_34_21*y_9_9^2*y_9_0^2 + 22*y_32_24*y_9_9^3*y_9_0 + y_32_24*y_9_9*y_9_0^3 +
    y_19_12*y_9_9^4*y_9_0 + y_19_12*y_9_9^3*y_9_0^2 + 22*y_35_24*y_9_9^2*y_9_0 +
    y_35_24*y_9_9*y_9_0^2 + 22*y_32_24*y_9_9^2*y_9_0 + 2*y_32_24*y_9_9*y_9_0^2 +
    y_32_24*y_9_0^3 + 2*y_31_21*y_9_9^2*y_9_0 + 2*y_38_24*y_9_9*y_9_0 +
    2*y_32_24*y_9_0^2 + 2*y_38_24*y_9_0
50 y_38_33*y_35_24 + 22*y_19_12*y_9_9^5*y_9_0 + y_19_12*y_9_9^3*y_9_0^3 +
    22*y_32_24*y_9_9^3*y_9_0 + y_32_24*y_9_9*y_9_0^3 + 22*y_31_21*y_9_9^4 +
    y_31_21*y_9_9^2*y_9_0^2 + 22*y_38_24*y_9_9^3 + y_38_24*y_9_9*y_9_0^2 +
    y_34_21*y_9_9^3 + y_34_21*y_9_9^2*y_9_0 + 22*y_32_24*y_9_9^2*y_9_0 +
    y_32_24*y_9_0^3 + y_38_24*y_9_9*y_9_0 + y_38_24*y_9_0^2 + y_19_12*y_9_9^4 +
    2*y_19_12*y_9_9^3*y_9_0 + y_19_12*y_9_9^2*y_9_0^2 + y_35_24*y_9_9^2 +
    y_35_24*y_9_9*y_9_0 + 22*y_31_21*y_9_9^2 + y_38_24*y_9_9 + y_38_24*y_9_0 +
    y_35_24*y_9_0 + y_19_12*y_9_9^2 + 22*y_35_24
51 y_38_33*y_34_30 + 22*y_9_9^7*y_9_0 + y_9_9^6*y_9_0^2 + y_9_9^5*y_9_0^3 +
    22*y_9_9^4*y_9_0^4 + 22*y_31_21*y_9_9^4 + 2*y_31_21*y_9_9^3*y_9_0 +
    22*y_31_21*y_9_9^2*y_9_0^2 + 22*y_12_9*y_9_9^6 + y_12_9*y_9_9^5*y_9_0 +
    y_12_9*y_9_9^4*y_9_0^2 + 22*y_12_9*y_9_9^3*y_9_0^3 + 22*y_19_12*y_9_9^5 +
    y_19_12*y_9_9^4*y_9_0 + y_19_12*y_9_9^3*y_9_0^2 + 22*y_19_12*y_9_9^2*y_9_0^3
    + y_15_9*y_9_9^5 + 2*y_15_9*y_9_9^4*y_9_0 + 20*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_31_21*y_9_9^3 + 2*y_31_21*y_9_9^2*y_9_0 + 22*y_31_21*y_9_9*y_9_0^2 +
    y_19_12*y_9_9^4 + y_19_12*y_9_9^3*y_9_0 + 22*y_19_12*y_9_9^2*y_9_0^2 +
    22*y_19_12*y_9_9*y_9_0^3 + y_9_9^6 + 3*y_9_9^5*y_9_0 + 22*y_9_9^4*y_9_0^2 +
```

```
    20*y_9_9^3*y_9_0^3 + y_34_30*y_9_9^2 + 3*y_34_21*y_9_9^2 +
    20*y_34_21*y_9_9*y_9_0 + 22*y_12_9*y_9_9^4 + 22*y_12_9*y_9_9^3*y_9_0 +
    2*y_19_12*y_9_9^3 + 21*y_19_12*y_9_9*y_9_0^2 + y_34_21*y_9_9 +
    20*y_34_21*y_9_0 + 19*y_15_9*y_9_9^2*y_9_0 + 22*y_31_21*y_9_9 + y_9_9^4 +
    y_9_9^3*y_9_0 + 22*y_34_30 + 21*y_34_21 + 22*y_31_21 + y_19_12*y_9_9 +
    y_19_12
52 y_38_33*y_34_21 + 22*y_9_9^6*y_9_0^2 + y_9_9^4*y_9_0^4 +
    22*y_31_21*y_9_9^3*y_9_0 + y_31_21*y_9_9^2*y_9_0^2 + 22*y_12_9*y_9_9^5*y_9_0
    + y_12_9*y_9_9^3*y_9_0^3 + 22*y_19_12*y_9_9^4*y_9_0 +
    y_19_12*y_9_9^2*y_9_0^3 + 2*y_15_9*y_9_9^3*y_9_0^2 +
    22*y_31_21*y_9_9^2*y_9_0 + y_31_21*y_9_9*y_9_0^2 + y_19_12*y_9_9^2*y_9_0^2 +
    y_19_12*y_9_9*y_9_0^3 + 2*y_9_9^4*y_9_0^2 + 2*y_9_9^3*y_9_0^3 +
    2*y_34_21*y_9_9*y_9_0 + y_19_12*y_9_9^2*y_9_0 + y_19_12*y_9_9*y_9_0^2 +
    2*y_34_21*y_9_0 + 2*y_15_9*y_9_9^2*y_9_0
53 y_38_33*y_32_24 + 22*y_34_21*y_9_9^4 + y_34_21*y_9_9^3*y_9_0 +
    22*y_38_24*y_9_9^3 + y_38_24*y_9_9^2*y_9_0 + 22*y_19_12*y_9_9^5 +
    y_19_12*y_9_9^3*y_9_0^2 + 22*y_35_24*y_9_9^3 + y_35_24*y_9_9^2*y_9_0 +
    y_31_21*y_9_9^3 + 22*y_38_24*y_9_9^2 + y_38_24*y_9_9*y_9_0 +
    y_35_24*y_9_9*y_9_0 + 2*y_34_21*y_9_9^2 + 2*y_32_24*y_9_9*y_9_0 +
    22*y_19_12*y_9_9^3 + y_35_24*y_9_9 + 2*y_32_24*y_9_0
54 y_38_33*y_31_21 + 22*y_15_9*y_9_9^5*y_9_0 + y_15_9*y_9_9^3*y_9_0^3 +
    22*y_19_12*y_9_9^4*y_9_0 + y_19_12*y_9_9^2*y_9_0^3 + 22*y_9_9^6*y_9_0 +
    22*y_9_9^5*y_9_0^2 + y_9_9^4*y_9_0^3 + y_9_9^3*y_9_0^4 + 22*y_34_21*y_9_9^3
    + y_34_21*y_9_9*y_9_0^2 + 2*y_12_9*y_9_9^3*y_9_0^2 +
    22*y_19_12*y_9_9^3*y_9_0 + y_19_12*y_9_9*y_9_0^3 + 22*y_34_21*y_9_9^2 +
    2*y_34_21*y_9_9*y_9_0 + y_34_21*y_9_0^2 + 22*y_15_9*y_9_9^4 +
    2*y_15_9*y_9_9^3*y_9_0 + 3*y_15_9*y_9_9^2*y_9_0^2 + 2*y_31_21*y_9_9*y_9_0 +
    2*y_34_21*y_9_0 + 2*y_31_21*y_9_0 + 2*y_12_9*y_9_9^2*y_9_0
55 y_38_33*y_19_12 + 22*y_12_9*y_9_9^4*y_9_0 + y_12_9*y_9_9^3*y_9_0^2 +
    22*y_34_21*y_9_9^2 + y_34_21*y_9_9*y_9_0 + 22*y_15_9*y_9_9^4 +
    y_15_9*y_9_9^2*y_9_0^2 + 22*y_31_21*y_9_9^2 + y_31_21*y_9_9*y_9_0 +
    y_9_9^4*y_9_0 + y_9_9^3*y_9_0^2 + 22*y_34_21*y_9_9 + y_34_21*y_9_0 +
    y_31_21*y_9_0 + 2*y_12_9*y_9_9^2*y_9_0 + y_19_12*y_9_9^2 +
    y_19_12*y_9_9*y_9_0 + y_31_21 + y_19_12*y_9_0 + 22*y_19_12
56 y_38_33*y_15_9 + 22*y_35_24*y_9_9^2 + y_35_24*y_9_9*y_9_0 + 2*y_32_24*y_9_0
57 y_38_33*y_12_9 + 22*y_32_24*y_9_9^2 + y_32_24*y_9_0^2 + 2*y_38_24
-----------------------------------------
delta= x2^2*x1^3 + x2*x1^4
-----------------------------------------
817.870
```

8 Timings

Here are rough timing comparisons, all done on the same departmental computer. Times vary somewhat, even on the same machine with the same problem, and certainly are machine dependent. But this clearly points out how dependent certain implementations are on the characteristic, and whether they are competitive with each other time-wise. (— means not applicable, ** means time and/or space resources were exhausted.)

char	*normal*	*normalP*	*icFrac −tions*	*icFracP*	*Normal− isation*	*Integral Closure*	*M2 Qth*	*Magma Qth*
0	**	−−−	2.7	−−−	**	−−−	570.	818.
2	15	**	2.2	85.	**	−−−	1.5	0.3
3	(129)	**	0.7	**	**	−−−	1.9	0.3
5	**		1.2		**	−−−	5.8	0.5
7			0.8			−−−	18.5	1.7
11			1.0			−−−		11.
13			1.0			−−−		19.
17			1.1			−−−		39.
19			1.3			−−−		82.
23			1.1			−−−		397.
29			1.2			−−−		
31			1.2			−−−		
37						−−−		
41						−−−		
101			1.1			−−−		
4001			1.3			−−−		

The problems with `icFracP` and `normalP` are usually due to poorly thought-out computational implemetentations of the crucial step $f^q(mod\ I)$ of the qth-power method. But what causes `Normalisation` to run out of memory and `normal` to head to never-never land is unknown to me, especially since `icFractions` has no problem producing a technically correct answer.