Addendum to
A weighted module view of integral closures of affine domains of type I

Douglas A. Leonard

Department of Mathematics and Statistics
Auburn University
Auburn, AL 36830

The paper above, given in its initial form in Linz in 2006, but not published until 2009 in Advances in Mathematics of Communication, claimed that certain material left out for various reasons would be available on my website.

The example from Section 3 should be in the download entitled *example with larger generating set* below. The example with weights 25,21 alluded to in the paper should be in the download entitled *example in two variables with minimization.*

The example immediately below is a disguised version of a familiar *Hermitian curve* defined by $y^4 + y = x^5$ in characteristic 2, but left out of the final version of the paper. It is given to exhibit various shortcomings of the currently available implementations in MAGMA, SINGULAR, and MACAULAY2, in hopes that these will be improved along the lines I am suggesting.

As this happens, I will try to update this page to reflect said improvements. I encourage correspondence, both positive and negative, on this subject.

Instead of using $f_5^4 + f_5 - f_4^5 = 0$ to describe the Hermitian curve in characteristic 2, which has many points over $\mathbf{F}_{16}$, disguise it by using $y := f_5 f_4$ and $x := f_4^2$ to get the defining equation $y^8 + y^2 x^3 + x^9 = 0$ instead.

MAGMA's *IntegralClosure* function is an implementation of algorithms designed for number fields, then extended to functions fields in one independent variable.

```
F:=GF(2);
FF<x>:=FunctionField(F);
P<y>:=PolynomialRing(FF);
f:=y^8+y^2*x^3+x^9;
Ff:=FunctionField(f);
RER<Y>:=RationalExtensionRepresentation(Ff);
C<X>:=CoefficientRing(RER);
INT:=Integers(C);
IC:=IntegralClosure(INT,RER);
B:=Basis(IC);B:
```

gives an error because it only works for separable extensions.

```
F:=GF(2);
FF<y>:=FunctionField(F);
P<x>:=PolynomialRing(FF);
f:=x^9+x^3*y^2+y^8;
Ff:=FunctionField(f);
RER<X>:=RationalExtensionRepresentation(Ff);
C<Y>:=CoefficientRing(RER);
INT:=Integers(C);
IC:=IntegralClosure(INT,RER);
B:=Basis(IC);B:
```

is separable and gives output

```
[ 1, X, X^2, 1/Y*X^3, 1/Y^3*X^4 + 1/Y^2*X, 1/Y^4*X^5 + 1/Y^3*X^2, 1/Y^4*X^6 +
    1/Y^3*X^3, 1/Y^4*X^7 + 1/Y^2*X, 1/Y^6*X^8 + 1/Y^4*X^2 ]
```

allows for no weights to be placed on $\mathbf{F}_2(Y)[X]$ (since Y is a unit), produces an $\mathbf{F}_2(Y)$-module basis of size $d = s = 9$ as elements of $\mathbf{F}_2(Y)[X]$ with distinct degrees in X, and treats the integral closure as being a subset of $\mathbf{F}_2(Y)[X]/\langle x^9 + x^3y^2 + y^8\rangle$.

SINGULARs *normal* function (as of version 3.0.4) is an implementation of deJong's algorithm for producing an increasing sequence of rings starting at S and terminating at $ic(S)$. (It has a bug supposedly to be fixed in 3.1 that causes premature termination.)

```
LIB ‘‘normal.lib’’;
ring r=2,(y,x),wp(9,8);
```

```
ideal i=y8+y2x3+x9;
list nor=normal(i);
def R=nor[1];
setring R;
normap;
norid;
```

allows n to be arbitrary, uses a primary decomposition to break a general problem into parts, uses Jacobians (but seems to recompute them for every new ring produced) to avoid most separability problems, *ignores* the monomial ordering given, choosing instead to impose a grevelex ordering on a list of variables gotten by tacking on variables defining ring R_{i+1} over R_i at the end, minimizes this list of variables by possibly removing variables of any type that are noted to be products of other variables. This often produces completely unreadable (though often correct) presentations of the integral closure, with structure having no relation to the structure of the original ring. Curiously, the individual ring extensions are based on quadratic and linear relations, but over the previous ring; and no attempt is made either theoretically or practically to try for a presentation of the same form.

```
normap[1]=T(1)
normap[2]=T(1)^2*T(4)^2+T(2)
> norid;
norid[1]=T(1)^2*T(4)^3+T(1)^2+T(2)*T(4)
norid[2]=T(2)^5+T(1)*T(3)
norid[3]=T(1)*T(2)^4*T(4)^3+T(1)*T(2)^4+T(3)*T(4)
norid[4]=T(4)^13+T(1)^4*T(2)^3*T(4)+T(1)^2*T(2)^4*T(4)^2
         +T(1)*T(3)+T(4)
norid[5]=T(1)*T(4)^12+T(1)^5*T(2)^3+T(1)^3*T(2)^4*T(4)
         +T(1)^2*T(3)*T(4)^2+T(2)*T(3)+T(1)
norid[6]=T(1)^9*T(3)*T(4)+T(2)^4*T(4)^6+T(1)^5*T(2)^2*T(3)
         +T(1)^3*T(2)^3*T(3)*T(4)+T(1)*T(2)^4*T(3)*T(4)^2
         +T(2)^4+T(3)^2
norid[7]=T(1)^10*T(4)+T(1)^6*T(2)^2+T(1)^4*T(2)^3*T(4)
         +T(1)^2*T(2)^4*T(4)^2+T(4)^4+T(1)*T(3)+T(4)
```

Ignoring the bug, consider the form of this answer. From normap, it is possible to see that $y \to T(1)$, that $x \to T(1)^2 * T(4)^2 + T(2)$. To figure out what $T(2)$, $T(3)$, and $T(4)$ are is hopeless from this presentation, as is any idea of how one might effectively use such a presentation, despite the fact that it was generated from quadratic and linear relations at every recursion. It is possible to produce a slightly better presentation:

```
ring r1=2,(t4,t3,t2,t1,y,x),(lp(4),wp(9,8));
map phi=R,t1,t2,t3,t4;
ideal i1=phi(norid),y-t1,x-t1^2*t4^2-t2;
option(redSB);
```

```
 ideal j1=std(i1);j1;
j1[1]=y^8+x^9+y^2*x^3
j1[2]=t1+y
j1[3]=t2*x^2+y^6+x^3
j1[4]=t2*y^2+x^7
j1[5]=t2^2+t2*x+y^4*x^5
j1[6]=t3*y+y^6*x^17+y^2*x^14+y^4*x^8+y^6*x^2+x^5
j1[7]=t3*x^2+t2*y+y^5*x^19+y*x^16+y^3*x^10+y^5*x^4
j1[8]=t3*t2+t3*x+y^3*x^24+y^5*x^18+y^7*x^12+y*x^15
j1[9]=t3^2+y^2*x^43+y^4*x^37+y^2*x^28+y^6*x^16+y^2*x^13+y^4*x^7+y^6*x+x^4
j1[10]=t4*x+y^2
j1[11]=t4*y^4+t2*x+x^2
j1[12]=t4*t2+x^6
j1[13]=t4*t3+y^7*x^16+y^3*x^13+y^5*x^7+y^7*x+y*x^4
j1[14]=t4^2*y^2+t2+x
j1[15]=t4^3*y+t3*x+y^5*x^18+y*x^15+y^3*x^9+y^5*x^3+y
j1[16]=t4^4+t4+x^5
```

from which it can be seen that $T(2)$ was $\frac{y^6+x^3}{x^2} = \frac{x^7}{y^2}$, that $T(3) = \frac{y^6x^{17}+y^2x^{14}+y^4x^8+y^6x^2+x^5}{y}$, and that $T(4) = \frac{y^2}{x}$. So had this been produced, it would have been clear that these were functions of weights 38, 181, and 10 respectively, not the 4, 5, 10, and 15 expected. It is left to the intereseted reader to decide whether the following is correct or not:

```
 ring r=2,(x,y),wp(8,9);
 ideal i=x9+x3y2+y8;
 list nor=normal(i);
 def R=nor[1];
 setring R;
> normap;
normap[1]=T(1)
normap[2]=T(1)^3*T(3)+T(2)
> norid;
norid[1]=T(7)^2+T(6)*T(8)
norid[2]=T(6)*T(7)+T(5)*T(8)
norid[3]=T(6)^2+T(5)*T(7)
norid[4]=T(1)*T(5)*T(7)+T(4)*T(8)
norid[5]=T(3)^2*T(7)+T(1)*T(8)
norid[6]=T(1)*T(5)*T(6)+T(4)*T(7)
norid[7]=T(3)^2*T(6)+T(1)*T(7)
norid[8]=T(2)*T(5)^2+T(1)*T(3)*T(7)+T(3)*T(6)+T(1)*T(7)+T(6)
norid[9]=T(1)*T(5)^2+T(4)*T(6)
norid[10]=T(3)^2*T(5)+T(1)*T(6)
norid[11]=T(3)*T(4)^2+T(3)*T(5)+T(5)
norid[12]=T(3)^2*T(4)+T(1)^2*T(5)
norid[13]=T(1)^2*T(4)+T(3)^2+T(3)
```

```
norid[14]=T(2)^3+T(4)^3+T(3)*T(4)*T(5)+T(2)*T(5)*T(6)+T(4)*T(5)
norid[15]=T(3)^3*T(8)+T(2)*T(5)*T(7)+T(3)^2*T(8)+T(3)*T(8)+T(8)
norid[16]=T(5)^3*T(7)+T(1)*T(2)*T(3)*T(8)+T(2)^2*T(4)*T(8)
          +T(4)*T(5)*T(6)*T(8)+T(1)*T(2)*T(8)
norid[17]=T(2)^2*T(5)*T(7)+T(1)^3*T(8)+T(5)^3*T(8)+T(1)*T(3)*T(6)
          +T(2)*T(4)*T(6)+T(1)^2*T(7)+T(1)*T(6)
norid[18]=T(1)*T(2)*T(4)*T(7)+T(1)^3*T(8)+T(1)^2*T(7)+T(2)*T(8)
norid[19]=T(2)*T(5)*T(6)+T(1)*T(3)*T(8)+T(3)*T(7)+T(1)*T(8)+T(7)
norid[20]=T(5)^3*T(6)+T(1)*T(2)*T(3)*T(7)+T(2)^2*T(4)*T(7)
          +T(4)*T(5)^2*T(8)+T(1)*T(2)*T(7)
norid[21]=T(1)^3*T(7)+T(2)^2*T(4)*T(8)+T(4)*T(5)*T(6)*T(8)+T(1)*T(3)*T(5)
          +T(2)*T(4)*T(5)+T(1)^2*T(6)+T(2)*T(3)*T(7)+T(1)*T(5)+T(2)*T(7)
norid[22]=T(1)*T(2)*T(4)*T(6)+T(2)^2*T(4)*T(8)+T(4)*T(5)*T(6)*T(8)
          +T(1)*T(3)*T(5)+T(2)*T(4)*T(5)+T(2)*T(3)*T(7)+T(1)*T(5)
norid[23]=T(2)^2*T(3)*T(6)+T(1)*T(2)^2*T(7)+T(4)^2*T(5)*T(7)
          +T(3)*T(4)*T(6)*T(8)+T(1)*T(4)*T(7)*T(8)+T(2)*T(3)*T(4)
          +T(2)*T(4)+T(8)^2+T(1)
norid[24]=T(1)*T(2)^2*T(6)+T(4)^2*T(5)*T(6)+T(4)^3*T(8)+T(3)*T(4)*T(5)*T(8)
          +T(1)*T(4)*T(6)*T(8)+T(1)*T(2)*T(4)+T(2)^2*T(5)+T(5)^2*T(6)
          +T(4)*T(5)*T(8)+T(1)^2+T(7)*T(8)
norid[25]=T(4)^2*T(5)^2+T(4)^3*T(7)+T(3)*T(4)*T(5)*T(7)+T(1)*T(4)*T(5)*T(8)
          +T(1)^3+T(2)*T(3)+T(6)*T(8)
norid[26]=T(1)*T(2)*T(4)*T(5)+T(1)^3*T(6)+T(1)^2*T(5)+T(2)*T(6)
norid[27]=T(2)^2*T(3)*T(5)+T(4)^3*T(8)+T(1)^2*T(3)+T(1)*T(2)*T(4)
          +T(4)*T(5)*T(8)+T(1)^2
norid[28]=T(1)*T(2)*T(3)*T(5)+T(2)^2*T(4)*T(5)+T(4)*T(5)^2*T(6)+T(3)^3
          +T(2)^2*T(7)+T(5)^2*T(8)+T(3)^2
norid[29]=T(1)*T(2)^2*T(5)+T(4)^3*T(7)+T(3)*T(4)*T(5)*T(7)+T(1)*T(4)*T(5)*T(8)
          +T(1)^3+T(2)*T(3)^2+T(5)^3+T(4)*T(5)*T(7)+T(2)*T(3)+T(6)*T(8)
norid[30]=T(1)^3*T(5)+T(2)*T(4)^2+T(1)*T(4)+T(2)*T(5)
norid[31]=T(3)^4+T(1)*T(2)*T(4)^2+T(3)^3+T(1)*T(2)*T(5)+T(3)^2+T(3)
norid[32]=T(1)*T(3)^3+T(1)*T(3)^2+T(1)*T(3)+T(2)*T(4)+T(1)
norid[33]=T(1)^4+T(4)^3*T(6)+T(3)*T(4)*T(5)*T(6)+T(1)*T(2)*T(3)
          +T(4)^2*T(8)+T(5)*T(8)
```

Macaulay2's *integralClosure* function is similar in flavor to normal in SINGULAR:

```
R=ZZ/2[y,x,MonomialOrder=>{Weights=>{1,0},Weights=>{9,8}}]
J=ideal(y^8+y^2*x^3+x^9)
S=R/J
integralClosure S

      ZZ
      -- [w  , w  ]
       2   21   19
```

```
o4 = ---------------
      5     4
     w   + w   + w
      21    19    19
```

gives a "correct" answer, while

```
R=ZZ/2[x,y,MonomialOrder=>{Weights=>{1,0},Weights=>{8,9}}]
J=ideal(y^8+y^2*x^3+x^9)
S=R/J
integralClosure S
```

```
                                                                                ZZ
                                                                                -- [w  , w  ,
                                                                                 2   22   13
o8 = -----------------------------------------------------------------------------------------
            2    4   6    2      2     8    4     4 2               3 5    3      6    3
     (w  x  + y , x  + w  x + y , x*y  + w   + x y  + w  , w  w   + x y  + w  y + y  + x  +
       13                13                13           13   22 13             13
```

is both unreadable because of poor I/O (with exponents on a separate line) and because it is wrong, with $wt(w_{13}) = 20$, and $wt(w_{22}) = 34$.

icFracP in Macaulay 2 now and a version of it called **normal_p** in SINGULAR which produce only fractions to define $ic(S)$ over S; based on a stripped-down view of the qth power algorithm (written by Singh ans Swanson) which ignores everything except the descending-chain-of-modules and the Frobenius map. These provide the fractions very quickly for small positive characteristic, but normal_p's "withring" option (probably the old normal ring output) does not necessarily have a finite running time. To be fair integralClosure may not have a finite running time on the same moderate-sized examples that normal and "withring" fail on.]

```
icFracP S

        4          5      2   6    4 6    2 12    18    3
       y          y      y   y  + y x  + y x   + x   + x
o6 = {1, --------, ---------, --, ---------------------------}
              4     2    5   x                3
         y*x + x   y*x  + x                   x
```

gives fractions which would have weights 0, 4, 5, 10, and curiously 120 respectively.