Affine desingularization using unimodular change-of-variables maps

Douglas A. Leonard, Auburn University

We'll use example 2.65 on page 107-108 of Kollár's *Lectures on Resolution of Singularities*, which we'll write as

$$b_0 := x^4 + x^2yz^9 + xy^9z + y^5z^5$$

as a reasonable example for affine desingularization. This has singularities when $x = 0 = yz$.

```
Macaulay2, version 1.10
R=QQ[x,y,z]
f=x^4+x^2*y*z^9+x*y^9*z+y^5*z^5;
radical(ideal(flatten entries jacobian ideal f)+ideal f)
-- ideal (x, y*z)
L=flatten entries gens monomialIdeal(terms(f)); toString L
--{x^4, x*y^9*z, y^5*z^5, x^2*y*z^9}
n=numColumns(vars(R))
M=id_(QQ^(n))||matrix{for i to n-1 list -1}||
     transpose matrix(for j from 1 to n list for i to n-1 list
         degree(R_i,L_j)-degree(R_i,L_0))
-- |  1  0  0  |
-- |  0  1  0  |
-- |  0  0  1  |
-- | -1 -1 -1  |
-- | -3 -4 -2  |
-- |  9  5  1  |
-- |  1  5  9  |
GG=transpose (gens gb M);
l=lcm(for i to n-1 list lcm(for j to n list denominator(GG_(i,j))));
G=l*GG
-- | 40 -80 40   0 120   0   0   |
-- | 26 -25 11 -12   0 120   0   |
-- | 6  -15 21 -12   0   0 120   |
B=for j to numRows(G) list gcd(flatten entries G_j);
C=for j to numRows(G) list apply(entries G_j,(x->x//B_j))
--{{20, 13, 3}, {-16, -5, -3}, {40, 11, 21}, {0, -1, -1}}
```

```
uni=(n,V)->(
      G1=gens gb (id_(ZZ^n)||V);
      M1=matrix(for i to n-1 list for j to n-1 list G1_(i,j));
      G2=gens gb (id_(ZZ^n)||M1);
      M2=transpose matrix(for i to n-1 list for j to n-1 list G2_(i,j));
      M3=matrix(for i to n-1 list for j to n-1 list abs(M2_(i,j)));
      M4=matrix(prepend(entries M3_(n-1),for j to n-2 list entries M3_j));
      map(R,R,matrix{for i to n-1 list product(for j to n-1 list R_j^((M4_i)_j))})
          );
phi=uni(3,matrix{{40,11,21}}); toString phi
--map(R,R,{x^40*y^11, x^11*y^3, x^21*y^6*z})
toString(factor phi f)
--(y)^44*(x)^160*(x^120*y^35*z^9+y*z^5+z+1)
phi=uni(3,matrix{{20,13,3}}); toString phi
--map(R,R,{x^20*y^3, x^13*y^2, x^3*z})
toString(factor phi f)
--(y)^8*(x)^80*(x^60*y^13*z+z^9+y^2*z^5+y^4)
phi=uni(3,matrix{{25,7,13}}); toString phi
--map(R,R,{x^25*y^7, x^7*y^2, x^13*y^4*z})
toString(factor phi f)
--(y)^28*(x)^100*(x^74*y^24*z^9+y^2*z^5+x*y*z+1)
phi=uni(3,matrix{{10,3,5}}); toString phi
--map(R,R,{x^10*y^3, x^3*y, x^5*y*z})
toString(factor phi f)
--(y)^10*(x)^40*(x^28*y^6*z^9+x^2*y^3*z+z^5+y^2)
phi=uni(3,matrix{{8,5,1}}); toString phi
--map(R,R,{x^8*y^3, x^5*y^2, x*z})
toString(factor phi f)
--(y)^8*(x)^30*(x^24*y^13*z+z^9+y^2*z^5+x^2*y^4)
phi=uni(3,matrix{{14,9,2}}); toString phi
--map(R,R,{x^14*y^3, x^9*y^2, x^2*z})
toString(factor phi f)
--(y)^8*(x)^55*(x^42*y^13*z+z^9+y^2*z^5+x*y^4)
phi=uni(3,matrix{{7,5,1}}); toString phi
--map(R,R,{x^7*y^3, x^5*y^2, x*z})
toString(factor phi f)
--(y)^8*(x)^28*(x^25*y^13*z+x^2*y^2*z^5+z^9+y^4)
phi=uni(3,matrix{{5,2,2}}); toString phi
--map(R,R,{x^5*y^2, x^2*y, x^2*y*z})
toString(factor phi f)
```

```
--(y)^8*(x)^20*(x^10*y^6*z^9+x^5*y^4*z+y^2*z^5+1)
phi=uni(3,matrix{{5,3,1}}); toString phi
--map(R,R,{x^5*y^2, x^3*y, x*z})
toString(factor phi f)
--(y)^5*(x)^20*(x^13*y^6*z+x^2*z^9+z^5+y^3)
phi=uni(3,matrix{{4,1,3}}); toString phi
--map(R,R,{x^4*y, x, x^3*y*z})
toString(factor phi f)
--(y)^2*(x)^16*(x^20*y^9*z^9+x^4*y^3*z^5+y^2+z)
phi=uni(3,matrix{{4,1,2}}); toString phi
--map(R,R,{x^4*y, x, x^2*y*z})
toString(factor phi f)
--(y)^2*(x)^15*(x^12*y^9*z^9+y^3*z^5+x*y^2+z)
phi=uni(3,matrix{{4,0,1}}); toString phi
--map(R,R,{x^4*z, y, x})
toString(factor phi f)
--(x)^5*(x^12*y*z^2+x^11*z^4+y^9*z+y^5)
phi=uni(3,matrix{{2,1,0}}); toString phi
--map(R,R,{x^2*y, x, z})
toString(factor phi f)
--(x)^5*(y^2*z^9+x^6*y*z+x^3*y^4+z^5)
phi=uni(3,matrix{{1,0,3}}); toString phi
--map(R,R,{x, y, x^3*z})
toString(factor phi f)
--(x)^4*(x^25*y*z^9+x^11*y^5*z^5+y^9*z+1)
phi=uni(3,matrix{{1,2,0}}); toString phi
--map(R,R,{x, x^2*y, z})
toString(factor phi f)
--(x)^4*(x^15*y^9*z+x^6*y^5*z^5+y*z^9+1)
phi=uni(3,matrix{{11,3,6}}); toString phi
--map(R,R,{x^11*y^4, x^3*y, x^6*y^2*z})
toString(factor phi f)
--(y)^15*(x)^44*(x^35*y^12*z^9+x*z^5+y+z)
```

Compare this to the result from `resolve.lib` in `Singular 4.1.0`

```
 LIB "resolve.lib";
 ring R=0,(x,y,z),dp;
 ideal I=x4+x2yz9+xy9z+y5z5;
 list re=resolve(I);
 timer;
--24
size(re[1]);size(re[2]);
--364
--785
presentTree(re);
```

This has 785 blowups, leading to 364 leafs, visible one at a time using the `presentTree(re)` command. These have a similar enough flavor to our proposed desingularization tree, as to be comparable.

It should be noted that Kollár seems to have used this to suggest that a blowup approach did not lead to an obvious criterion for improvement.

Our suggestion is that one works on disjoint parts of the variety differentiated by valuations as opposed to blowing up (possibly overlapping) geometric objects, with our progress measured by getting closer to finding d explicit, independent local parameters and a local unit at each valuation.