Proposition 7.3 (Class 24)

June 19, 2015

```
# parametrize the Hadamard square root of rank five

#[
# [1,  0,  1,  0,  1,  0, 1, 0,  0,   0],
# [1,  0,  1,  0,  0,  1,y3, 1,  1,   0],
# [1,  0,  0,  1,  1,  0,y4,y7,  0,   1],
# [1,  0,  0,  1,  0,  1, 0,y8,  0,   0],
# [0,  1,  1,  0, y1,  0,y5, 0,y10, y14],
# [0,  1,  1,  0,  0, y1, 0, 0,y11,   0],
# [0,  1,  0,  1, y1,  0, 0, 0,  0, y15],
# [0,  1,  0,  1,  0, y1, 0,y9,y12, y16],
# [0,  0,  0,  0,  0,  0,y6, 1,y13, y17],
# [1,y18,y19,y20,y21,y22, 0 ,0, 0,    0]]

# the other possible embedding of the octahedral facet of the polar are \
   equivalent to the chosen parametrization
# for the verification please run the commented code to obtain all \
   possible parametrizations of the octahedron


# R.<a1,a2>=QQ[];

# M = matrix(R,[
#  [ 1, 0, 1, 0, 1, 0,a2, 0, 0, 0],
#  [ 1, 0, 1, 0, 0, 1,a2,a2,a2, 0],
#  [ 1, 0, 0, 1, 1, 0,a2,a2, 0,a2],
#  [ 1, 0, 0, 1, 0, 1, 0,a2, 0, 0],
#  [ 0, 1, 1, 0,a1, 0,a2, 0,a2,a2],
#  [ 0, 1, 1, 0, 0,a1, 0, 0,a2, 0],
#  [ 0, 1, 0, 1,a1, 0, 0, 0, 0,a2],
#  [ 0, 1, 0, 1, 0,a1, 0,a2,a2,a2],
#  [ 0, 0, 0, 0, 0, 0,a2,a2,a2,a2],
#  [a2,a2,a2,a2,a2,a2, 0 ,0, 0, 0]]); M

# M.matrix_from_rows_and_columns\
   ([0,1,4,5,2,3,6,7,8,9],[2,3,0,1,4,5,6,8,7,9])


# M.matrix_from_rows_and_columns\
   ([0,4,1,5,2,6,3,7,8,9],[2,3,4,5,0,1,6,8,9,7])
```

```
R.<y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20\
   ,y21>=QQ[];

M = matrix(R,[
 [1,  0,  1,  0,  1,  0, 1, 0,  0,   0],
 [1,  0,  1,  0,  0,  1,y3, 1,  1,   0],
 [1,  0,  0,  1,  1,  0,y4,y7,  0,   1],
 [1,  0,  0,  1,  0,  1, 0,y8,  0,   0],
 [0,  1,  1,  0, y1,  0,y5, 0,y10, y14],
 [0,  1,  1,  0,  0, y1, 0, 0,y11,   0],
 [0,  1,  0,  1, y1,  0, 0, 0,  0, y15],
 [0,  1,  0,  1,  0, y1, 0,y9,y12, y16],
 [0,  0,  0,  0,  0,  0,y6, 1,y13, y17],
 [1,y18,y19,y20,y21, y2, 0 ,0, 0,    0]]); M
J=ideal(M.matrix_from_columns([0,2,3,4,5,6,7]).minors(6));
JJ=ideal(y1*y2*y3*y4*y5*y6*y7*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*\
   y19*y20*y21)
KK=J.saturation(JJ)
KK
```

```
[  1   0   1   0   1   0   1   0   0   0]
[  1   0   1   0   0   1  y3   1   1   0]
[  1   0   0   1   1   0  y4  y7   0   1]
[  1   0   0   1   0   1   0  y8   0   0]
[  0   1   1   0  y1   0  y5   0 y10 y14]
[  0   1   1   0   0  y1   0   0 y11   0]
[  0   1   0   1  y1   0   0   0   0 y15]
[  0   1   0   1   0  y1   0  y9 y12 y16]
[  0   0   0   0   0   0  y6   1 y13 y17]
[  1 y18 y19 y20 y21  y2   0   0   0   0]
(Ideal (y5*y20 + y3*y21 - y6*y21 - y3 + y6 - y21, y1*y19 + y1*y20 - y1 - y2 - y21 + 1,
y6*y9 - y5, y6*y8 + y3 - y5 - y6, y5*y8 + y3*y9 - y5*y9 - y5, y6*y7 - y4 - y5 + 1, y5*y7 -
y4*y9 - y5*y9 + y9, y3*y7 + y4*y8 - y3*y9 - y4*y9 - y4 - y8 + y9 + 1, y2*y3 - y2*y6 +
y5*y19 - y2 - y5 + 1, y1*y3 - y1*y6 - y1 + y5, y4*y9*y20 - y4*y8*y21 + y4*y9*y21 + y4*y8 -
y4*y9 - y9*y20 - y7*y21 + y8*y21 - y8 + y9, y3*y9*y20 - y3*y8*y21 + y3*y9*y21 + y3*y8 -
y3*y9 + y8*y21 - y9*y21 - y21, y2*y9*y20 - y9*y19*y21 + y2*y8 - y2*y9 - y9*y20 + y8*y21 -
y8 + y9, y1*y9*y20 + y1*y8 - y1*y9 - y9*y21, y2*y4*y8 - y2*y4*y9 - y4*y9*y19 + y2*y7 -
y2*y8 + y4*y9 + y9*y19 - y7, y1*y4*y8 - y1*y4*y9 + y1*y7 - y1*y8 - y4*y9 + y9) of
Multivariate Polynomial Ring in y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13,
y14, y15, y16, y17, y18, y19, y20, y21 over Rational Field, 0)
```

```
# y6*y8 + y3 - y5 - y6 => y5 = y3*y8
# y6*y7 - y4 - y5 + 1, y6*y9 - y5 => y6*y7 - y4 - y6*y9 + 1 => y7 = y4*y9

R.<y1,y2,y3,y4,y6,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21>=\
   QQ[];

M = matrix(R,[
 [1,  0,  1,  0,  1,  0,    1,    0,  0,   0],
 [1,  0,  1,  0,  0,  1,   y3,    1,  1,   0],
```

```
 [1,  0,  0,  1,  1,  0,   y4,y4*y9,  0,   1],
 [1,  0,  0,  1,  0,  1,    0,   y8,  0,   0],
 [0,  1,  1,  0, y1,  0,y3*y8,    0,y10, y14],
 [0,  1,  1,  0,  0, y1,    0,    0,y11,   0],
 [0,  1,  0,  1, y1,  0,    0,    0,  0, y15],
 [0,  1,  0,  1,  0, y1,    0,   y9,y12, y16],
 [0,  0,  0,  0,  0,  0,   y6,    1,y13, y17],
 [1,y18,y19,y20,y21, y2,    0,    0, 0,   0]]); M
J=ideal(M.matrix_from_columns([0,1,2,3,4,5,6,7]).minors(6));
JJ=ideal(y1*y2*y3*y4*y6*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*y19*y20\
   *y21)
KK=J.saturation(JJ)
KK
```

```
[   1    0    1    0    1    0    1    0    0    0]
[   1    0    1    0    0    1   y3    1    1    0]
[   1    0    0    1    1    0   y4 y4*y9    0    1]
[   1    0    0    1    0    1    0   y8    0    0]
[   0    1    1    0   y1    0 y3*y8    0  y10  y14]
[   0    1    1    0    0   y1    0    0  y11    0]
[   0    1    0    1   y1    0    0    0    0  y15]
[   0    1    0    1    0   y1    0   y9  y12  y16]
[   0    0    0    0    0    0   y6    1  y13  y17]
[   1  y18  y19  y20  y21   y2    0    0    0    0]
```

```
(Ideal (y18 - y19 - y20 + 1, y1*y19 + y1*y20 - y1 - y2 - y21 + 1, y8^2 - y8*y9 - y8 + y9,
y6*y8 - y6*y9 + y3 - y6, y3*y8 - y6*y9, y1*y4 + y4*y8 - y4*y9 - y1 - y4 - y8 + y9 + 1,
y1*y3 - y1*y6 + y6*y9 - y1, y3*y4*y21 - y4*y6*y21 - y3*y4 + y4*y6 + y4*y20 - y3*y21 -
y4*y21 + y6*y21 + y3 - y6 - y20 + y21, y6*y9*y20 + y3*y21 - y6*y21 - y3 + y6 - y21,
y4*y9*y20 - y4*y8*y21 + y4*y8 - y4*y9 - y9*y20 + y8*y21 - y8 + y9, y3*y9*y20 + y3*y9*y21 -
y6*y9*y21 - y3*y9 + y6*y9 + y8*y21 - y9*y21 - y21, y2*y9*y20 - y9*y19*y21 + y2*y8 - y2*y9
- y9*y20 + y8*y21 - y8 + y9, y1*y9*y20 + y1*y8 - y1*y9 - y9*y21, y4*y8*y20 - y4*y8*y21 -
y4*y20 - y8*y20 + y4*y21 + y8*y21 + y20 - y21, y2*y8*y20 - y8*y19*y21 - y2*y20 - y8*y20 +
y8*y21 + y19*y21 + y20 - y21, y1*y8*y20 - y1*y20 - y8*y21 + y21, y2*y6*y20 - y6*y19*y21 -
y2*y3 + y2*y6 - y3*y19 + y6*y19 + y2*y20 - y3*y20 - y3*y21 + 2*y6*y21 - y19*y21 + 2*y3 -
2*y6 - y20 + y21, y1*y6*y20 + y6*y9 + y1*y20 - y6*y21 - y1 - y3 + y6 - y21, y3*y4*y20 -
y4*y6*y21 - y3*y4 + y4*y6 - y3*y20 + y6*y21 + y3 - y6, y2*y4*y20 - y4*y19*y21 - y2*y4 +
y4*y19 - y2*y20 + y19*y21 + y2 - y19, y3^2*y20 - y3*y6*y20 + y3^2*y21 - 2*y3*y6*y21 +
y6^2*y21 - y3^2 + 2*y3*y6 - y6^2 - y3*y21 + y6*y21, y2*y3*y20 - y3*y19*y21 - y2*y3 + y2*y6
- y3*y20 + y6*y21 + y3 - y6, y6*y9*y19 + y2*y3 - y2*y6 - y6*y9 - y2 + 1, y4*y8*y19 -
y4*y9*y19 + y2*y4 - y4*y19 - y8*y19 + y9*y19 - y2 + y19, y6*y9^2 - y1*y8 - y3*y9 + y1,
y4*y6*y9 - y6*y9 - y4 + 1, y3*y6*y9 - y6^2*y9 - y3^2 + y3*y6, y3*y4*y9 + y4*y8 - y3*y9 -
y4*y9 - y4 - y8 + y9 + 1, y2*y3*y9 - y2*y6*y9 + y3*y9*y19 + y2*y8 - y2*y9 - y3*y9 - y2 -
y8 + y9 + 1, y2*y4*y8 - y4*y9*y19 - y2*y8 + y9*y19, y2*y4*y6 - y3*y4*y19 - y2*y6 + y3*y19,
y3^2*y4 - y3*y4*y6 - y3^2 - y3*y4 + y3*y6 + y4*y6 + y3 - y6, y2*y3*y4 - y3*y4*y19 - y2*y3
- y2*y4 + y3*y19 + y4*y19 + y2 - y19, y2*y3^2 - 2*y2*y3*y6 + y2*y6^2 + y3^2*y19 -
y3*y6*y19 - y2*y3 - y3^2 + y2*y6 + y3*y6 + y3 - y6, y4*y6*y20*y21 - y4*y6*y21^2 -
y4*y6*y20 - y4*y20^2 + y4*y6*y21 + y4*y20*y21 - y6*y20*y21 + y6*y21^2 + y4*y20 + y6*y20 +
y20^2 - y4*y21 - y6*y21 - y20*y21 - y20 + y21, y2*y4*y9*y19 - y4*y9*y19^2 - y2^2*y4 +
y2*y4*y19 - y2*y9*y19 + y9*y19^2 + y2^2 - y2*y19) of Multivariate Polynomial Ring in y1,
y2, y3, y4, y6, y8, y9, y10, y11, y12, y13, y14, y15, y16, y17, y18, y19, y20, y21 over
Rational Field, 1)
```

```
# y18 - y19 - y20 + 1 => y18 = y19*y20
# y1*y3 - y1*y6 + y6*y9 - y1 => y1 = y3*y9

R.<y2,y3,y4,y6,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y19,y20,y21>=QQ[];

M = matrix(R,[
 [1,       0,  1,  0,     1,     0,     1,     0,  0,    0],
 [1,       0,  1,  0,     0,     1,    y3,     1,  1,    0],
 [1,       0,  0,  1,     1,     0,    y4,y4*y9,  0,    1],
 [1,       0,  0,  1,     0,     1,     0,    y8,  0,    0],
 [0,       1,  1,  0,y3*y9,     0,y3*y8,     0,y10, y14],
 [0,       1,  1,  0,     0,y3*y9,     0,     0,y11,    0],
 [0,       1,  0,  1,y3*y9,     0,     0,     0,  0, y15],
 [0,       1,  0,  1,     0,y3*y9,     0,    y9,y12, y16],
 [0,       0,  0,  0,     0,     0,    y6,     1,y13, y17],
 [1,y19*y20,y19,y20,  y21,   y2,     0,     0, 0,    0]]); M
J=ideal(M.matrix_from_columns([0,1,2,3,4,5,6,8]).minors(6));
JJ=ideal(y2*y3*y4*y6*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y19*y20*y21)
KK=J.saturation(JJ)
KK
```

```
[      1       0       1       0       1       0       1       0       0       0]
[      1       0       1       0       0       1      y3       1       1       0]
[      1       0       0       1       1       0      y4   y4*y9       0       1]
[      1       0       0       1       0       1       0      y8       0       0]
[      0       1       1       0   y3*y9       0   y3*y8       0     y10     y14]
[      0       1       1       0       0   y3*y9       0       0     y11       0]
[      0       1       0       1   y3*y9       0       0       0       0     y15]
[      0       1       0       1       0   y3*y9       0      y9     y12     y16]
[      0       0       0       0       0       0      y6       1     y13     y17]
[      1 y19*y20     y19     y20     y21      y2       0       0       0       0]
(Ideal (y19*y20 - y19 - y20 + 1, y3*y13 + y4*y13 - y6 - y13, y8*y11 - y8*y12 - y8 + y10 -
y11 + y12, y6*y11 - y6*y12 - y4*y13 + y13, y3*y11 + y4*y11 - y3*y12 - y4*y12 - y4 - y11 +
y12 + 1, y3*y8 - y3*y10 - y4*y10 + y4 + y10 - 1, y2*y4 + y4*y19 - y4 - y19, y2*y12*y20 +
y2*y11 - y2*y12 - y12*y20 + y11*y21 - y12*y21 - y11 + y12, y9*y11*y20 - y9*y12*y20 -
y2*y11 - y9*y11 + y9*y12 - y9*y20 + y12*y20 - y11*y21 + y12*y21 + y9 + y11 - y12,
y6*y12*y19 + y4*y13*y19 + y6*y12*y20 - y4*y13*y21 - y6*y12 - y13*y19, y3*y12*y19 +
y4*y12*y19 + y3*y12*y20 + y4*y12*y20 - y3*y12 - y4*y12 + y4*y19 - y12*y19 - y12*y20 -
y4*y21 + y12 - y19, y9*y11*y19 - y9*y12*y19 + y2*y12 - y9*y19 + y12*y19 - y12, y4*y11*y19
+ y4*y12*y20 - y4*y11*y21 + y4*y12*y21 - y4*y12 - y11*y19 - y12*y20 + y12, y2*y11*y19 +
y11*y19*y21 - y12*y19*y21 - y2*y11 - y11*y19 - y11*y21 + y12*y21 + y11, y9*y10*y19 -
y8*y12*y19 + y9*y10*y20 - y8*y12*y20 + y2*y8 - y2*y10 - y9*y10 + y8*y12 - y9*y19 + y12*y19
- y9*y20 + y12*y20 + y8*y21 - y10*y21 - y8 + y9 + y10 - y12, y3*y9*y19 + y3*y9*y20 - y3*y9
- y2 - y21 + 1, y4*y8*y13 - y6*y8 + y6*y10 - y4*y13 - y8*y13 + y13, y2*y8*y12 + y9*y10*y20
- y8*y12*y20 + y2*y8 - y2*y10 - y9*y10 - y2*y12 - y9*y20 + y12*y20 + y8*y21 - y10*y21 - y8
+ y9 + y10, y2*y3*y12 - y4*y12*y19 + y3*y9*y20 - y3*y12*y20 - y4*y12*y20 - y3*y9 - y2*y12
+ y4*y12 - y4*y19 + y12*y19 + y12*y20 + y4*y21 - y2 + y19 - y21 + 1, y4*y9*y11 - y4*y9*y12
- y4*y9 + y12, y4*y9*y10 - y4*y9 - y8*y12 + y12, y3*y4*y9 - y3*y12 - y4*y12 + y12,
y2*y6*y8 - y2*y6*y10 + y2*y8*y13 + y6*y8*y19 - y6*y10*y19 - y6*y8 + y6*y10 - y2*y13 -
y8*y13 + y13, y6*y12*y20^2 - y4*y13*y20*y21 - y6*y12*y20 + y4*y13*y20 + y4*y13*y21 -
y4*y13 - y13*y20 + y13, y4*y12*y20^2 - y4*y11*y20*y21 + y4*y12*y20*y21 + y4*y11*y20 -
```

```
2*y4*y12*y20 - y12*y20^2 + y4*y11*y21 - y4*y12*y21 - y4*y11 + y4*y12 - y11*y20 + 2*y12*y20
+ y11 - y12, y3*y12*y20^2 + y4*y11*y20*y21 - y4*y12*y20*y21 - y4*y11*y20 - y3*y12*y20 +
y4*y12*y20 - y4*y11*y21 + y4*y12*y21 - y4*y20*y21 + y4*y11 - y4*y12 + y4*y20 + y11*y20 -
y12*y20 + y4*y21 - y4 - y11 + y12 - y20 + 1, y9*y10*y20^2 - y8*y12*y20^2 + y2*y8*y20 -
y2*y10*y20 - y9*y10*y20 + y8*y12*y20 - y9*y20^2 + y12*y20^2 + y8*y20*y21 - y10*y20*y21 -
y2*y8 + y2*y10 - y8*y20 + y9*y20 + y10*y20 - y12*y20 - y8*y21 + y10*y21 + y8 - y10,
y3*y9*y20^2 - y3*y9*y20 - y2*y20 - y20*y21 + y2 + y20 + y21 - 1, y4*y9*y13*y20 - y2*y6*y12
- y4*y9*y13 + y4*y13*y19 - y6*y9*y20 + y6*y12*y20 - y9*y13*y20 - y4*y13*y21 + y6*y9 +
y2*y13 + y9*y13 - y13*y19 + y13*y21 - y13, y4*y9*y13*y19 + y2*y6*y12 - y6*y9*y19 -
y4*y13*y19 - y9*y13*y19 - y6*y12*y20 + y4*y13*y21 + y13*y19, y4*y9*y12*y19 + y4*y9*y12*y20
- y4*y9*y12 + y4*y9*y19 - y9*y12*y19 - y9*y12*y20 - y4*y9*y21 + y2*y12 + y9*y12 - y9*y19 +
y12*y21 - y12, y2*y9*y12*y19 - y2*y9*y11 - y2^2*y12 + y2*y9*y19 - y2*y12*y19 - y9*y12*y19
- y9*y11*y21 - y2*y12*y21 + y9*y12*y21 + y9*y19*y21 - y12*y19*y21 + y9*y11 + 2*y2*y12 -
y9*y19 + y12*y19 + y12*y21 - y12, y4*y8*y12*y19 + y4*y8*y12*y20 - y4*y8*y12 + y4*y8*y19 -
y4*y10*y19 - y4*y12*y19 - y8*y12*y19 - y4*y12*y20 - y8*y12*y20 - y4*y8*y21 + y4*y10*y21 +
y4*y12 + y8*y12 - y8*y19 + y10*y19 + y12*y19 + y12*y20 - y12, y2*y6*y9*y19 + y2*y9*y13*y19
+ y6*y9*y19^2 + y2*y6*y9*y20 + y2*y9*y13*y20 - y2*y6*y9 - y2^2*y13 - y2*y9*y13 - y6*y9*y19
- y2*y13*y19 - y9*y13*y19 - y9*y13*y20 - y2*y13*y21 - y13*y19*y21 + 2*y2*y13 + y9*y13 +
y13*y19 + y13*y21 - y13, y4^2*y9*y13 - y4*y6*y9 - y4*y9*y13 + y6*y12, y2*y6*y10*y12 -
y4*y10*y13*y19 + y6*y9*y10*y20 - y6*y10*y12*y20 + y9*y10*y13*y20 - y8*y12*y13*y20 +
y4*y10*y13*y21 - y6*y9*y10 - y2*y6*y12 - y2*y10*y13 - y9*y10*y13 + y8*y12*y13 + y4*y13*y19
+ y10*y13*y19 - y6*y9*y20 + y6*y12*y20 - y9*y13*y20 + y12*y13*y20 - y4*y13*y21 -
y10*y13*y21 + y6*y9 + y2*y13 + y9*y13 + y10*y13 - y12*y13 - y13*y19 + y13*y21 - y13,
y2^2*y6*y12 + y4*y13*y19^2 + y2*y6*y9*y20 + y2*y9*y13*y20 - y4*y13*y19*y21 - y2*y6*y9 -
y2^2*y13 - y2*y9*y13 + y2*y13*y19 - 2*y4*y13*y19 - y13*y19^2 - y6*y12*y20 - y9*y13*y20 -
y2*y13*y21 + 2*y4*y13*y21 + y13*y19*y21 + y9*y13 + y13*y19 - y13*y21 + y13, y2*y9*y11^2 +
y2^2*y11*y12 - y2*y9*y11*y12 + y9*y11^2*y21 + y2*y11*y12*y21 - 2*y9*y11*y12*y21 -
y2*y12^2*y21 + y9*y12^2*y21 - y2*y9*y11 - y9*y11^2 - y2*y11*y12 + y9*y11*y12 - y9*y11*y21
+ y9*y12*y21 + y9*y11, y6*y8*y9*y10 - y6*y9*y10^2 + y8*y9*y10*y13 - y8^2*y12*y13 -
y6*y8*y9 + y6*y9*y10 - y8*y9*y13 - y9*y10*y13 + 2*y8*y12*y13 + y9*y13 - y12*y13,
y2*y6*y9*y20^2 + y2*y9*y13*y20^2 - y2*y6*y9*y20 - y2^2*y13*y20 - y2*y9*y13*y20 -
y9*y13*y20^2 - y2*y13*y20*y21 + y2^2*y13 + y2*y13*y20 + y9*y13*y20 + y2*y13*y21 - y2*y13)
of Multivariate Polynomial Ring in y2, y3, y4, y6, y8, y9, y10, y11, y12, y13, y14, y15,
y16, y17, y19, y20, y21 over Rational Field, 1)
```

```
# y2*y4 + y4*y19 - y4 - y19 => y2*y4 = 1
# y6*y11 - y6*y12 - y4*y13 + y13 => y11 = y4*y12*y13
# y3*y13 + y4*y13 - y6 - y13 => y6 = y3*y4*y13

R.<y3,y4,y8,y9,y10,y12,y13,y14,y15,y16,y17,y19,y20,y21>=QQ[];

M = matrix(R,[
 [1,      0,  1,  0,    1,       0,        1,    0,     0,   0],
 [1,      0,  1,  0,    0,      y4,       y3,    1,     1,   0],
 [1,      0,  0,  1,    1,       0,       y4,y4*y9,     0,   1],
 [1,      0,  0,  1,    0,      y4,        0,   y8,     0,   0],
 [0,      1,  1,  0,y3*y9,       0,    y3*y8,    0,   y10, y14],
 [0,      1,  1,  0,    0,y3*y9*y4,        0,    0,y4*y12,   0],
 [0,      1,  0,  1,y3*y9,       0,        0,    0,     0, y15],
 [0,      1,  0,  1,    0,y3*y9*y4,        0,   y9,   y12, y16],
 [0,      0,  0,  0,    0,       0,y3*y4*y13,    1,   y13, y17],
```

```
 [1,y19*y20,y19,y20,  y21,       1,        0,    0, 0,    0]]); M
J=ideal(M.matrix_from_columns([0,1,2,3,4,5,6,8]).minors(6));
JJ=ideal(y3*y4*y8*y9*y10*y12*y13*y14*y15*y16*y17*y19*y20*y21)
KK=J.saturation(JJ)
KK
```

```
[       1        0        1        0        1        0        1        0        0
0]
[       1        0        1        0        0       y4       y3        1        1
0]
[       1        0        0        1        1        0       y4    y4*y9        0
1]
[       1        0        0        1        0       y4        0       y8        0
0]
[       0        1        1        0    y3*y9        0    y3*y8        0      y10
y14]
[       0        1        1        0        0 y3*y4*y9        0        0   y4*y12
0]
[       0        1        0        1    y3*y9        0        0        0        0
y15]
[       0        1        0        1        0 y3*y4*y9        0       y9      y12
y16]
[       0        0        0        0        0        0 y3*y4*y13        1      y13
y17]
[       1   y19*y20      y19      y20      y21        1        0        0        0
0]
(Ideal (y9 - y12, y19*y20 - y19 - y20 + 1, y8*y19 - y10*y19 - y8 + y10, y4*y19 - y4 - y19
+ 1, y4*y10 - y4 - y8 + 1, y3*y8 - y3*y10 - y8 + y10, y3*y4 - y3 - y4 + 1, y4*y8*y21 -
y4*y8 - y8*y20 + y10*y20 - y4*y21 - y8*y21 + y4 + 2*y8 - y10 + y21 - 1, y4^2*y21 - y4^2 -
y4*y20 - y4*y21 + 2*y4 + y20 - 1, y8*y12*y20 - y10*y12*y20 - y8*y12 + y10*y12 - y8*y21 +
y10*y21 + y8 - y10, y4*y12*y20 - y4*y12 - y12*y20 - y4*y21 + y4 + y12 + y21 - 1,
y8*y10*y20 - y10^2*y20 - y8^2*y21 + y8*y10*y21 + y8^2 - 2*y8*y10 + y10^2 - y8*y20 +
y10*y20 + y8*y21 - y10*y21, y3*y12*y19 + y3*y12*y20 - y3*y12 + y4*y12 - y12 - y21,
y8^2*y12 - y8*y10*y12 - y8*y10 + y10^2 - y8*y12 + y10*y12 + y8 - y10, y4*y8*y12 - y4*y12 -
y8*y12 - y8 + y10 + y12, y4^2*y12 - y4*y12 - y4 + 1, y3*y12*y20^2 - y3*y12*y20 + y4*y21 -
y20*y21 - y4 + 1) of Multivariate Polynomial Ring in y3, y4, y8, y9, y10, y12, y13, y14,
y15, y16, y17, y19, y20, y21 over Rational Field, 1)
```

```
# y9 - y12
# y4*y10 - y4 - y8 + 1 => y10 = y8


R.<y3,y4,y8,y9,y13,y14,y15,y16,y17,y19,y20,y21>=QQ[];

M = matrix(R,[
 [1,       0,  1,  0,     1,        0,         1,     0,      0,   0],
 [1,       0,  1,  0,     0,       y4,        y3,     1,      1,   0],
 [1,       0,  0,  1,     1,        0,        y4,y4*y9,      0,   1],
 [1,       0,  0,  1,     0,       y4,         0,    y8,      0,   0],
 [0,       1,  1,  0,y3*y9,        0,     y3*y8,     0,     y8, y14],
 [0,       1,  1,  0,     0,y3*y9*y4,         0,     0, y4*y9,   0],
 [0,       1,  0,  1,y3*y9,        0,         0,     0,      0, y15],
```

```
 [0,       1,  0,  1,     0,y3*y9*y4,         0,    y9,     y9, y16],
 [0,       0,  0,  0,     0,        0,y3*y4*y13,     1,    y13, y17],
 [1,y19*y20,y19,y20,  y21,        1,         0,     0, 0,     0]]); M
J=ideal(M.matrix_from_columns([0,1,2,3,4,5,6,7,8]).minors(6));
JJ=ideal(y3*y4*y8*y9*y13*y14*y15*y16*y17*y19*y20*y21)
KK=J.saturation(JJ)
KK
```

```
[       1        0        1        0        1        0        1        0        0
0]
[       1        0        1        0        0       y4       y3        1        1
0]
[       1        0        0        1        1        0       y4     y4*y9        0
1]
[       1        0        0        1        0       y4        0       y8        0
0]
[       0        1        1        0     y3*y9        0     y3*y8        0       y8
y14]
[       0        1        1        0        0  y3*y4*y9        0        0     y4*y9
0]
[       0        1        0        1     y3*y9        0        0        0        0
y15]
[       0        1        0        1        0  y3*y4*y9        0       y9       y9
y16]
[       0        0        0        0        0        0 y3*y4*y13        1      y13
y17]
[       1   y19*y20      y19      y20      y21        1        0        0        0
0]
(Ideal (y13 - y19, y19*y21 - y19 - y21 + 1, y19*y20 - y19 - y20 + 1, y8*y20 - y9*y20 -
y4*y21 + y4 - y8 + y9 + y21 - 1, y8*y19 - y8 - y19 + 1, y4*y19 - y4 - y19 + 1, y3*y19 - y3
- y19 + 1, y4*y9 + y9*y19 - y8 - y9, y8^2 - y8*y9 - y8 + y9, y4*y8 - y4 - y8 + 1, y3*y8 -
y3*y9 - y8 + y9, y3*y4 - y3 - y4 + 1, y4^2*y21 - y4^2 - y4*y20 - y4*y21 + 2*y4 + y20 - 1,
y3*y9*y20 + y8 - y9 - y21, y9*y19^2 - y9*y19 - y19 + 1) of Multivariate Polynomial Ring in
y3, y4, y8, y9, y13, y14, y15, y16, y17, y19, y20, y21 over Rational Field, 1)
```

```
# y13 - y19
# y3*y9*y20 + y8 - y9 - y21 => y21 = y3*y8*y20

R.<y3,y4,y8,y9,y13,y14,y15,y16,y17,y20>=QQ[];

M = matrix(R,[
 [1,       0,  1,  0,        1,        0,         1,    0,      0,    0],
 [1,       0,  1,  0,        0,       y4,        y3,    1,      1,    0],
 [1,       0,  0,  1,        1,        0,        y4,y4*y9,      0,    1],
 [1,       0,  0,  1,        0,       y4,         0,   y8,      0,    0],
 [0,       1,  1,  0,    y3*y9,        0,     y3*y8,    0,     y8, y14],
 [0,       1,  1,  0,        0,y3*y9*y4,         0,    0, y4*y9,    0],
 [0,       1,  0,  1,    y3*y9,        0,         0,    0,      0, y15],
 [0,       1,  0,  1,        0,y3*y9*y4,         0,   y9,     y9, y16],
 [0,       0,  0,  0,        0,        0,y3*y4*y13,    1,    y13, y17],
 [1,y13*y20,y13,y20,y3*y8*y20,        1,         0,    0,      0,    0]]); M
J=ideal(M.minors(6));
```

```
JJ=ideal(y3*y4*y8*y9*y13*y14*y15*y16*y17*y20)
KK=J.saturation(JJ)
KK
```

```
[       1        0        1        0        1        0        1        0        0
0]
[       1        0        1        0        0       y4       y3        1        1
0]
[       1        0        0        1        1        0       y4     y4*y9        0
1]
[       1        0        0        1        0       y4        0       y8        0
0]
[       0        1        1        0     y3*y9        0     y3*y8        0       y8
y14]
[       0        1        1        0        0  y3*y4*y9        0        0     y4*y9
0]
[       0        1        0        1     y3*y9        0        0        0        0
y15]
[       0        1        0        1        0  y3*y4*y9        0       y9       y9
y16]
[       0        0        0        0        0        0 y3*y4*y13        1      y13
y17]
[       1   y13*y20      y13      y20 y3*y8*y20        1        0        0        0
0]
(Ideal (y13 - y17, y8 - y14 + y15 - 1, y17*y20 - y17 - y20 + 1, y16*y20 - y16 - y20 + 1,
y15*y20 - y16, y9*y20 - y14*y20 - y9 + y14 - y15 + y16 - y20 + 1, y4*y20 - y4 - y20 + 1,
y16*y17 - y16 - y17 + 1, y15*y17 - y15 - y17 + 1, y14*y17 - y14 - y17 + 1, y9*y17 - y14 +
y15 - y16, y4*y17 - y4 - y17 + 1, y3*y17 - y3 - y17 + 1, y15*y16 - y16^2 - y15 + y16,
y14*y16 - y16^2 - y14 + y16, y9*y16 - y16^2 - y9 + y16, y4*y16 - 1, y3*y16 - y3 - y16 + 1,
y9*y15 - y14*y15 + y15^2 - y16^2 - y9 + y14 - 2*y15 + 2*y16, y4*y15 - y15 + y16 - 1,
y9*y14 - y14^2 + y14*y15 - y16^2 - y9 - y15 + 2*y16, y4*y14 - y14 + y16 - 1, y3*y14 -
y3*y15 + y3 - y14 + y16 - 1, y4*y9 - y9 + y16 - 1, y3*y9 - y9 - y15 + y16, y3*y4 - y3 - y4
+ 1) of Multivariate Polynomial Ring in y3, y4, y8, y9, y13, y14, y15, y16, y17, y20 over
Rational Field, 0)
```

```
# y13 - y17, y15*y20 - y16
# y8 - y14 + y15 - 1 => y14 = y8*y15

R.<y3,y4,y8,y9,y13,y15,y20>=QQ[];

M = matrix(R,[
 [1,       0,  1,  0,         1,        0,         1,     0,      0,       0],
 [1,       0,  1,  0,         0,       y4,        y3,     1,      1,       0],
 [1,       0,  0,  1,         1,        0,        y4,y4*y9,      0,       1],
 [1,       0,  0,  1,         0,       y4,         0,    y8,      0,       0],
 [0,       1,  1,  0,     y3*y9,        0,     y3*y8,     0,     y8, y8*y15],
 [0,       1,  1,  0,         0,y3*y9*y4,         0,     0, y4*y9,       0],
 [0,       1,  0,  1,     y3*y9,        0,         0,     0,      0,     y15],
 [0,       1,  0,  1,         0,y3*y9*y4,         0,    y9,     y9,y15*y20],
 [0,       0,  0,  0,         0,        0,y3*y4*y13,     1,    y13,     y13],
 [1,y13*y20,y13,y20,y3*y8*y20,        1,         0,     0,      0,       0]]);\
     M
```

```
J=ideal(M.minors(6));
JJ=ideal(y3*y4*y8*y9*y13*y15*y20)
KK=J.saturation(JJ)
KK
```

```
[       1        0        1        0        1        0        1        0        0
0]
[       1        0        1        0        0       y4       y3        1        1
0]
[       1        0        0        1        1        0       y4     y4*y9        0
1]
[       1        0        0        1        0       y4        0       y8        0
0]
[       0        1        1        0     y3*y9        0     y3*y8        0       y8
y8*y15]
[       0        1        1        0        0  y3*y4*y9        0        0     y4*y9
0]
[       0        1        0        1     y3*y9        0        0        0        0
y15]
[       0        1        0        1        0  y3*y4*y9        0       y9       y9
y15*y20]
[       0        0        0        0        0        0 y3*y4*y13        1      y13
y13]
[       1   y13*y20      y13      y20 y3*y8*y20        1        0        0        0
0]
(Ideal (y15*y20 + y3 - y15 - 1, y13*y20 - y13 - y20 + 1, y9*y20 - y9 - y20 + 1, y8*y20 -
y8 - y20 + 1, y4*y20 - y4 - y20 + 1, y3*y20 - 1, y13*y15 - y13 - y15 + 1, y8*y15 - y8 -
y15 + 1, y4*y15 - y3, y3*y15 + y9*y15 - y15^2 - y3 - y9 + 1, y9*y13 + y3 - y8 - y15,
y8*y13 - y8 - y13 + 1, y4*y13 - y4 - y13 + 1, y3*y13 - y3 - y13 + 1, y4*y9 - y3 - y9 +
y15, y3*y9 - y3 - y9 + 1, y8^2 - y8*y9 - y8 + y9, y4*y8 - y4 - y8 + 1, y3*y8 - y3 - y8 +
1, y3*y4 - y3 - y4 + 1, y3^2 + y9*y15 - y15^2 - 2*y3 - y9 + y15 + 1, y9^2*y15 - y9*y15^2 -
y9^2 + y15^2 + y9 - y15) of Multivariate Polynomial Ring in y3, y4, y8, y9, y13, y15, y20
over Rational Field, 1)
```

```
KK[0].primary_decomposition();
```

```
[Ideal (y20 - 1, y13 - 1, y9 - y15, y8 - 1, y3 - 1, y4*y15 - 1) of Multivariate Polynomial
Ring in y3, y4, y8, y9, y13, y15, y20 over Rational Field, Ideal (y13 - 1, y9 - 1, y8 - 1,
y4 - 1, y3 - y15, y15*y20 - 1) of Multivariate Polynomial Ring in y3, y4, y8, y9, y13,
y15, y20 over Rational Field, Ideal (y20 - 1, y15 - 1, y8 - 1, y4 - 1, y3 - 1, y9*y13 - 1)
of Multivariate Polynomial Ring in y3, y4, y8, y9, y13, y15, y20 over Rational Field,
Ideal (y20 - 1, y15 - 1, y13 - 1, y8 - y9, y4 - 1, y3 - 1) of Multivariate Polynomial Ring
in y3, y4, y8, y9, y13, y15, y20 over Rational Field]
```

```
# results

# for y20 - 1, y13 - 1, y9 - y15, y8 - 1, y3 - 1, y4*y15 - 1 the slack \
   matrix has the form

M = matrix(R,[
 [1,  0,  1,  0,  1,  0, y9,  0,  0,  0],
 [1,  0,  1,  0,  0,  1, y9,  1,  1,  0],
 [1,  0,  0,  1,  1,  0,  1,  1,  0,  1],
```

```
 [1,  0,  0,  1,  0,  1,  0,  1,  0,  0],
 [0,  1,  1,  0, y9,  0, y9,  0,  1, y9],
 [0,  1,  1,  0,  0, y9,  0,  0,  1,  0],
 [0,  1,  0,  1, y9,  0,  0,  0,  0, y9],
 [0,  1,  0,  1,  0, y9,  0, y9, y9, y9],
 [0,  0,  0,  0,  0,  0,  1,  1,  1,  1],
 [1,  1,  1,  1,  1, y9,  0,  0,  0,  0]]);

M.matrix_from_rows_and_columns\
   ([3,2,1,0,7,6,5,4,8,9],[0,1,3,2,5,4,7,6,9,8])
[ 1  0  1  0  1  0  1  0  0  0]
[ 1  0  1  0  0  1  1  1  1  0]
[ 1  0  0  1  1  0  1 y9  0  1]
[ 1  0  0  1  0  1  0 y9  0  0]
[ 0  1  1  0 y9  0 y9  0 y9 y9]
[ 0  1  1  0  0 y9  0  0 y9  0]
[ 0  1  0  1 y9  0  0  0  0  1]
[ 0  1  0  1  0 y9  0 y9 y9  1]
[ 0  0  0  0  0  0  1  1  1  1]
[ 1  1  1  1 y9  1  0  0  0  0]
```

```
# for y13 - 1, y9 - 1, y8 - 1, y4 - 1, y3 - y15, y15*y20 - 1 the slack \
   matrix has the form

M = matrix(R,[
 [1,  0,  1,  0,   1,  0,  1,  0,  0,  0],
 [1,  0,  1,  0,   0,  1, y3,  1,  1,  0],
 [1,  0,  0,  1,   1,  0,  1,  1,  0,  1],
 [1,  0,  0,  1,   0,  1,  0,  1,  0,  0],
 [0,  1,  1,  0,  y3,  0, y3,  0,  1, y3],
 [0,  1,  1,  0,   0, y3,  0,  0,  1,  0],
 [0,  1,  0,  1,  y3,  0,  0,  0,  0, y3],
 [0,  1,  0,  1,   0, y3,  0,  1,  1,  1],
 [0,  0,  0,  0,   0,  0, y3,  1,  1,  1],
 [y3, 1, y3,  1,  y3, y3,  0,  0,  0,  0]]);

M.matrix_from_rows_and_columns\
   ([3,2,1,8,9,6,5,4,0,7],[7,1,3,8,5,9,0,6,4,2])
[ 1  0  1  0  1  0  1  0  0  0]
[ 1  0  1  0  0  1  1  1  1  0]
[ 1  0  0  1  1  0  1 y3  0  1]
[ 1  0  0  1  0  1  0 y3  0  0]
[ 0  1  1  0 y3  0 y3  0 y3 y3]
[ 0  1  1  0  0 y3  0  0 y3  0]
[ 0  1  0  1 y3  0  0  0  0  1]
[ 0  1  0  1  0 y3  0 y3 y3  1]
[ 0  0  0  0  0  0  1  1  1  1]
[ 1  1  1  1 y3  1  0  0  0  0]
```

```
# for y20 - 1, y15 - 1, y8 - 1, y4 - 1, y3 - 1, y9*y13 - 1 the slack \
   matrix has the form

M = matrix(R,[
 [1,      0,  1,  0,     1,    0,    1,    0,     0,      0],
 [1,      0,  1,  0,     0,    1,    1,    1,     1,      0],
 [1,      0,  0,  1,     1,    0,    1,   y9,     0,      1],
 [1,      0,  0,  1,     0,    1,    0,    1,     0,      0],
 [0,      1,  1,  0,    y9,    0,    1,    0,     1,      1],
 [0,      1,  1,  0,     0,   y9,    0,    0,    y9,      0],
 [0,      1,  0,  1,    y9,    0,    0,    0,     0,      1],
 [0,      1,  0,  1,     0,   y9,    0,   y9,    y9,      1],
 [0,      0,  0,  0,     0,    0,    1,   y9,     1,      1],
 [y9,     1,  1, y9,    y9,   y9,    0,    0,     0,      0]]);

M.matrix_from_rows_and_columns\
   ([0,1,2,8,9,5,6,7,3,4],[6,1,2,9,4,8,0,7,5,3])
[ 1  0  1  0  1  0  1  0  0  0]
[ 1  0  1  0  0  1  1  1  1  0]
[ 1  0  0  1  1  0  1 y9  0  1]
[ 1  0  0  1  0  1  0 y9  0  0]
[ 0  1  1  0 y9  0 y9  0 y9 y9]
[ 0  1  1  0  0 y9  0  0 y9  0]
[ 0  1  0  1 y9  0  0  0  0  1]
[ 0  1  0  1  0 y9  0 y9 y9  1]
[ 0  0  0  0  0  0  1  1  1  1]
[ 1  1  1  1 y9  1  0  0  0  0]
```

```
# for y20 - 1, y15 - 1, y13 - 1, y8 - y9, y4 - 1, y3 - 1 the slack matrix\
    has the form

M = matrix(R,[
 [1,   0,   1,   0,   1,    0,   1,    0,   0,   0],
 [1,   0,   1,   0,   0,    1,   1,    1,   1,   0],
 [1,   0,   0,   1,   1,    0,   1,   y8,   0,   1],
 [1,   0,   0,   1,   0,    1,   0,   y8,   0,   0],
 [0,   1,   1,   0, y8,    0, y8,    0, y8, y8],
 [0,   1,   1,   0,   0,   y8,   0,    0, y8,   0],
 [0,   1,   0,   1, y8,    0,   0,    0,   0,   1],
 [0,   1,   0,   1,   0,   y8,   0,   y8, y8,   1],
 [0,   0,   0,   0,   0,    0,   1,    1,   1,   1],
 [1,   1,   1,   1, y8,    1,   0,    0,   0,   0]]);
```