Proposition 7.5 (Class 30)

June 19, 2015

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

# 1)

#[
# [    1,    0,    1,    0,       1,       0,    1,    0],
# [    1,    0,    1,    0,       0,       1,   y9,    0],
# [    1,    0,    0,    1,       1,       0,  y10,    0],
# [    1,    0,    0,    1,       0,       1,  y11,    0],
# [    0,    1,    1,    0,      y1,       0,  y12,    0],
# [    0,    1,    1,    0,       0,      y1,  y13,    0],
# [    0,    1,    0,    1,      y1,       0,  y14,    0],
# [    0,    1,    0,    1,       0,      y1,  y15,    0],
# [   y2,    0,   y4,    0,      y6,       0,    0,    1],
# [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
# [   y2,    0,    0,   y5,      y6,       0,    0,  y17],
# [   y2,    0,    0,   y5,       0,      y8,    0,  y18],
# [    0,   y3,   y4,    0,  y6*y7,       0,    0,  y19],
# [    0,   y3,   y4,    0,       0,  y8*y7,    0,  y20],
# [    0,   y3,    0,   y5,  y6*y7,       0,    0,  y21],
# [    0,   y3,    0,   y5,       0,  y8*y7,    0,  y22]]

# 2)
#[
# [    1,    0,    1,    0,       1,       0,    1,    0],
# [    1,    0,    1,    0,       0,       1,   y9,    0],
# [    1,    0,    0,    1,       1,       0,  y10,    0],
# [    1,    0,    0,    1,       0,       1,  y11,    0],
# [    0,    1,    1,    0,      y1,       0,  y12,    0],
# [    0,    1,    1,    0,       0,      y1,  y13,    0],
# [    0,    1,    0,    1,      y1,       0,  y14,    0],
# [    0,    1,    0,    1,       0,      y1,  y15,    0],
# [   y2,    0,   y4,    0,      y6,       0,    0,    1],
# [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
# [   y2,    0,    0,   y5,  y6*y7,       0,    0,  y17],
# [   y2,    0,    0,   y5,       0,  y8*y7,    0,  y18],
# [    0,   y3,   y4,    0,      y6,       0,    0,  y19],
# [    0,   y3,   y4,    0,       0,      y8,    0,  y20],
# [    0,   y3,    0,   y5,  y6*y7,       0,    0,  y21],
# [    0,   y3,    0,   y5,       0,  y8*y7,    0,  y22]]
```

```
# case 1)
R.<y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20\
   ,y21,y22>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
 [   y2,    0,    0,   y5,      y6,       0,    0,  y17],
 [   y2,    0,    0,   y5,       0,      y8,    0,  y18],
 [    0,   y3,   y4,    0,  y6*y7,       0,    0,  y19],
 [    0,   y3,   y4,    0,       0,  y8*y7,    0,  y20],
 [    0,   y3,    0,   y5,  y6*y7,       0,    0,  y21],
 [    0,   y3,    0,   y5,       0,  y8*y7,    0,  y22]]); M
K=M.matrix_from_columns([0,1,2,3,4,7])
J=ideal(K.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*y22)
KK=J.saturation(JJ)
KK
```

```
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1   y9    0]
[   1    0    0    1    1    0  y10    0]
[   1    0    0    1    0    1  y11    0]
[   0    1    1    0   y1    0  y12    0]
[   0    1    1    0    0   y1  y13    0]
[   0    1    0    1   y1    0  y14    0]
[   0    1    0    1    0   y1  y15    0]
[  y2    0   y4    0   y6    0    0    1]
[  y2    0   y4    0    0   y8    0  y16]
[  y2    0    0   y5   y6    0    0  y17]
[  y2    0    0   y5    0   y8    0  y18]
[   0   y3   y4    0 y6*y7    0    0  y19]
[   0   y3   y4    0    0 y7*y8    0  y20]
[   0   y3    0   y5 y6*y7    0    0  y21]
[   0   y3    0   y5    0 y7*y8    0  y22]
(Ideal (y2*y22 - y4*y22 - y3 + y5, y4*y21 - y5*y21 - y4*y22 + y5*y22, y3*y21 - y5*y21 -
y3*y22 + y5*y22, y2*y21 - y5*y21 - y4*y22 + y5*y22 - y3 + y5, y3*y20 - y5*y20 - y3*y22 +
y4*y22, y2*y20 - y4*y20 - y3 + y4, y4*y19 - y5*y19 - y4*y20 + y5*y20, y3*y19 - y5*y19 -
```

```
y3*y22 + y4*y22, y2*y19 - y5*y19 - y4*y20 + y5*y20 - y3 + y4, y4*y18 - y5*y18 + y4*y20 -
y5*y20 - y4*y22 + y5*y22 - y4 + y5, y3*y18 - y5*y18 - y4*y22 + y5*y22 - y3 + y5, y2*y18 -
y5*y18 + y4*y20 - y5*y20 - y4*y22 + y5*y22 - y2 - y4 + 2*y5, y4*y17 - y5*y17 + y4*y20 -
y5*y20 - y4*y22 + y5*y22 - y4 + y5, y3*y17 - y5*y17 - y4*y22 + y5*y22 - y3 + y5, y2*y17 -
y5*y17 + y4*y20 - y5*y20 - y4*y22 + y5*y22 - y2 - y4 + 2*y5, y4*y16 - y5*y16 - y4 + y5,
y3*y16 - y5*y16 - y3 + y5, y2*y16 - y5*y16 - y2 + y5) 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, y22 over Rational Field, 0)
```

```
#  y3*y20 - y5*y20 - y3*y22 + y4*y22 => y4 = y5
#  y2*y20 - y4*y20 - y3 + y4 => y3 = y2

R.<y1,y2,y4,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,\
   y22>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
 [   y2,    0,    0,   y4,      y6,       0,    0,  y17],
 [   y2,    0,    0,   y4,       0,      y8,    0,  y18],
 [    0,   y2,   y4,    0,  y6*y7,       0,    0,  y19],
 [    0,   y2,   y4,    0,       0,  y8*y7,    0,  y20],
 [    0,   y2,    0,   y4,  y6*y7,       0,    0,  y21],
 [    0,   y2,    0,   y4,       0,  y8*y7,    0,  y22]]); M
K=M.matrix_from_columns([1,2,3,4,5,6])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*y19*y20\
   *y21*y22)
KK=J.saturation(JJ)
KK
```

```
[    1     0     1     0     1     0     1     0]
[    1     0     1     0     0     1    y9     0]
[    1     0     0     1     1     0   y10     0]
[    1     0     0     1     0     1   y11     0]
[    0     1     1     0    y1     0   y12     0]
[    0     1     1     0     0    y1   y13     0]
[    0     1     0     1    y1     0   y14     0]
[    0     1     0     1     0    y1   y15     0]
[   y2     0    y4     0    y6     0     0     1]
[   y2     0    y4     0     0    y8     0   y16]
[   y2     0     0    y4    y6     0     0   y17]
[   y2     0     0    y4     0    y8     0   y18]
[    0    y2    y4     0 y6*y7     0     0   y19]
```

```
[    0    y2    y4     0     0 y7*y8     0   y20]
[    0    y2     0    y4 y6*y7     0     0   y21]
[    0    y2     0    y4     0 y7*y8     0   y22]
(Ideal (y6*y13 - y8*y13 - y6*y15 + y8*y15, y4*y13 - y8*y13 - y4*y15 + y8*y15, y6*y12 -
y8*y12 - y6*y14 + y8*y14, y4*y12 - y8*y12 - y4*y14 + y8*y14, y6*y10 - y8*y10 - y6 + y8,
y4*y10 - y8*y10 - y4 + y8, y6*y9 - y8*y9 - y6*y11 + y8*y11, y4*y9 - y8*y9 - y4*y11 +
y8*y11, y1*y2*y13 - y7*y8*y13 - y1*y2*y15 + y7*y8*y15 - y2*y13 + y8*y13 + y2*y15 - y8*y15,
y1*y2*y12 - y7*y8*y12 - y1*y2*y14 + y7*y8*y14 - y2*y12 + y8*y12 + y2*y14 - y8*y14,
y4*y8*y11 - y6*y8*y11 - y4*y6 + y6*y8, y1*y6*y11 - y1*y8*y11 - y1*y6 + y1*y8 + y6*y14 -
y8*y14 - y6*y15 + y8*y15, y1*y4*y11 - y1*y8*y11 - y1*y4 + y1*y8 + y4*y14 - y8*y14 - y4*y15
+ y8*y15, y1*y2*y10 - y7*y8*y10 - y1*y2 + y7*y8 - y2*y10 + y8*y10 + y2 - y8, y1*y2*y9 -
y7*y8*y9 - y1*y2*y11 + y7*y8*y11 - y2*y9 + y8*y9 + y2*y11 - y8*y11, y1*y4*y6 - y1*y4*y8 +
y4*y8*y14 - y6*y8*y14 - y4*y8*y15 + y6*y8*y15, y1*y2*y4 - y4*y6*y7 - y2*y4*y14 + y2*y6*y14
- y2*y6 + y4*y6, y1*y2*y6*y14 - y6*y7*y8*y14 - y1*y2*y8*y15 + y6*y7*y8*y15 - y1*y2*y6 +
y1*y2*y8 - y2*y8*y14 + y6*y8*y14 + y2*y8*y15 - y6*y8*y15, y1*y2*y8*y11 - y6*y7*y8*y11 -
y1*y2*y8 + y6*y7*y8 - y2*y8*y11 + y6*y8*y11 - y2*y6*y14 + y2*y8*y14 + y2*y6 - y6*y8,
y4*y6*y7*y11 - y6*y7*y8*y11 + y2*y4*y11*y14 - y2*y6*y11*y14 - y4*y6*y7 + y6*y7*y8 +
y2*y6*y11 - y4*y6*y11 - y2*y8*y11 + y6*y8*y11 - y2*y4*y15 + y2*y8*y15 + y4*y6 - y6*y8,
y1^2*y2*y11 - y1*y7*y8*y11 - y1^2*y2 + y1*y7*y8 - y1*y2*y11 + y1*y8*y11 + y1*y2*y14 -
y7*y8*y14 - y1*y2*y15 + y7*y8*y15 + y1*y2 - y1*y8 - y2*y14 + y8*y14 + y2*y15 - y8*y15,
y4*y6^2*y7 - y4*y6*y7*y8 + y2*y4*y6*y14 - y2*y6^2*y14 - y2*y4*y8*y15 + y2*y6*y8*y15 +
y2*y6^2 - y4*y6^2 - y2*y6*y8 + y4*y6*y8, y6^2*y7*y8*y11 - y6*y7*y8^2*y11 - y6^2*y7*y8 +
y6*y7*y8^2 + y2*y6*y8*y11 - y6^2*y8*y11 - y2*y8^2*y11 + y6*y8^2*y11 + y2*y6^2*y14 -
y2*y6*y8*y14 - y2*y6*y8*y15 + y2*y8^2*y15 - y2*y6^2 + y2*y6*y8 + y6^2*y8 - y6*y8^2) of
Multivariate Polynomial Ring in y1, y2, y4, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15,
y16, y17, y18, y19, y20, y21, y22 over Rational Field, 1)
```

```
#  y4*y8*y11 - y6*y8*y11 - y4*y6 + y6*y8 =>  y8 = y6

R.<y1,y2,y4,y6,y7,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,y22\
   >=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y6,    0,  y16],
 [   y2,    0,    0,   y4,      y6,       0,    0,  y17],
 [   y2,    0,    0,   y4,       0,      y6,    0,  y18],
 [    0,   y2,   y4,    0,  y6*y7,       0,    0,  y19],
 [    0,   y2,   y4,    0,       0,  y6*y7,    0,  y20],
 [    0,   y2,    0,   y4,  y6*y7,       0,    0,  y21],
 [    0,   y2,    0,   y4,       0,  y6*y7,    0,  y22]]); M
K=M.matrix_from_columns([1,2,3,4,5,6,7])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*y19*y20*\
```

```
   y21*y22)
KK=J.saturation(JJ);
KK
(y9-y16) in KK[0]
(y10-y17) in KK[0]
(y11-y18) in KK[0]
[    1     0     1     0     1     0     1     0]
[    1     0     1     0     0     1    y9     0]
[    1     0     0     1     1     0   y10     0]
[    1     0     0     1     0     1   y11     0]
[    0     1     1     0    y1     0   y12     0]
[    0     1     1     0     0    y1   y13     0]
[    0     1     0     1    y1     0   y14     0]
[    0     1     0     1     0    y1   y15     0]
[   y2     0    y4     0    y6     0     0     1]
[   y2     0    y4     0     0    y6     0   y16]
[   y2     0     0    y4    y6     0     0   y17]
[   y2     0     0    y4     0    y6     0   y18]
[    0    y2    y4     0 y6*y7     0     0   y19]
[    0    y2    y4     0     0 y6*y7     0   y20]
[    0    y2     0    y4 y6*y7     0     0   y21]
[    0    y2     0    y4     0 y6*y7     0   y22]
(Ideal (y19 - y20 - y21 + y22, y17 + y20 - y22 - 1, y16 - y18 - y20 + y22, y13 - y15 - y20
+ y22, y12 - y14 - y20 + y22, y11 - y18, y10 + y20 - y22 - 1, y9 - y18 - y20 + y22, y4*y21
- y6*y21 - y4*y22 + y6*y22, y4*y20 - y6*y20 - y4*y22 + y6*y22, y14*y18 + y15*y20 + y15*y21
- y18*y21 - y14*y22 - y15*y22 - y20*y22 + y22^2 - y15 + y22, y7*y18 + y7*y20 - y7*y22 - y7
+ y21 - y22, y4*y18 - y6*y18 - y4 + y6, y1*y18 + y1*y20 - y1*y22 - y1 + y14 - y15, y2*y15
- y2*y18 + y6*y18 - y4*y22 + y4 - y6, y7*y14 - y7*y15 - y1*y21 + y1*y22, y4*y14 - y6*y14 -
y4*y15 + y6*y15, y2*y14 + y2*y20 - y6*y20 - y6*y21 - y2*y22 - y4*y22 + 2*y6*y22 - y2 + y4,
y1*y2 - y6*y7 - y4*y22 + y6*y22 - y2 + y4, y7*y15*y21 - y1*y20*y21 + y7*y20*y21 -
y7*y15*y22 + y1*y20*y22 - y7*y20*y22 - y7*y21*y22 + y7*y22^2 + y1*y21 - y7*y21 - y14*y21 +
y21^2 - y1*y22 + y7*y22 + y14*y22 - y21*y22, y7*y15*y20 - y1*y20^2 + y7*y20^2 - y7*y15*y22
+ y1*y20*y22 - 2*y7*y20*y22 + y7*y22^2 + y1*y20 - y7*y20 - y14*y20 + y20*y21 - y1*y22 +
y7*y22 + y14*y22 - y21*y22, y1*y14*y20 - y1*y15*y20 - y1*y15*y21 - y1*y20*y21 + y1*y15*y22
+ y1*y20*y22 + y1*y21*y22 - y1*y22^2 - y1*y14 + y14^2 + y1*y15 - y14*y15 + y1*y21 -
y14*y21 + y15*y21 - y1*y22, y6*y7*y15 - y1*y6*y20 + y6*y7*y20 - y1*y4*y22 + y1*y6*y22 -
y6*y7*y22 + y4*y15*y22 - y6*y15*y22 + y1*y4 - y6*y7 - y6*y14 - y4*y15 + y6*y15 + y6*y21)
of Multivariate Polynomial Ring in y1, y2, y4, y6, y7, y9, y10, y11, y12, y13, y14, y15,
y16, y17, y18, y19, y20, y21, y22 over Rational Field, 1)
True
True
True
```

```
# y9-y16, y10-y17, y11-y18
# y10 + y20 - y22 - 1 => y22 = y10*y20

R.<y1,y2,y4,y6,y7,y9,y10,y11,y12,y13,y14,y15,y19,y20,y21>=QQ[];

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

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

```
[     1      0      1      0      1      0      1      0]
[     1      0      1      0      0      1     y9      0]
[     1      0      0      1      1      0    y10      0]
[     1      0      0      1      0      1    y11      0]
[     0      1      1      0     y1      0    y12      0]
[     0      1      1      0      0     y1    y13      0]
[     0      1      0      1     y1      0    y14      0]
[     0      1      0      1      0     y1    y15      0]
[    y2      0     y4      0     y6      0      0      1]
[    y2      0     y4      0      0     y6      0     y9]
[    y2      0      0     y4     y6      0      0    y10]
[    y2      0      0     y4      0     y6      0    y11]
[     0     y2     y4      0  y6*y7      0      0    y19]
[     0     y2     y4      0      0  y6*y7      0    y20]
[     0     y2      0     y4  y6*y7      0      0    y21]
[     0     y2      0     y4      0  y6*y7      0 y10*y20]
```

```
Ideal (y13 - y15 - y19 + y21, y12 - y14 - y19 + y21, y10 + y19 - y21 - 1, y9 - y11 - y19 +
y21, y19*y20 - y20*y21 - y19 + y21, y4*y20 - y6*y20 - y4*y21 + y6*y21, y4*y19 - y6*y19 -
y4*y21 + y6*y21, y11*y14 + y14*y19 + y15*y19 + y19^2 - y14*y20 - y11*y21 - y14*y21 -
2*y19*y21 + y21^2 - y15 - 2*y19 + y20 + 2*y21, y7*y14 - y7*y15 - y1*y19 + y1*y20, y4*y14 -
y6*y14 - y4*y15 + y6*y15, y2*y14 + y2*y19 - y6*y19 - y2*y21 - y4*y21 + y6*y21 - y2 + y4,
y7*y11 + y7*y19 - y7*y21 - y7 + y19 - y20, y4*y11 - y6*y11 - y4 + y6, y2*y11 - y6*y11 -
y2*y15 - y6*y19 + y6*y20 + y4*y21 - y4 + y6, y1*y11 + y1*y19 - y1*y21 - y1 + y14 - y15,
y1*y2 - y6*y7 - y4*y21 + y6*y21 - y2 + y4, y7*y15*y20 - y1*y20^2 - y7*y15*y21 - y7*y19*y21
+ y1*y20*y21 + y7*y21^2 + y7*y19 + y1*y20 - y7*y20 - y14*y20 - y1*y21 + y14*y21 + y20*y21
- y21^2, y7*y15*y19 + y7*y19^2 - y7*y15*y21 - 2*y7*y19*y21 + y7*y21^2 - y7*y19 - y14*y19 +
y7*y21 + y14*y21 + y19*y21 - y21^2, y1*y15*y19 + y1*y19^2 - y1*y14*y20 - y1*y19*y21 +
y1*y14 - y14^2 - y1*y15 + y14*y15 - 2*y1*y19 + y1*y20 + y1*y21 + y14*y21 - y15*y21,
y6*y7*y15 + y6*y7*y19 - y1*y6*y20 - y1*y4*y21 + y1*y6*y21 - y6*y7*y21 + y4*y15*y21 -
y6*y15*y21 + y1*y4 - y6*y7 - y6*y14 - y4*y15 + y6*y15 + y6*y21, y2*y7*y15 + y2*y7*y19 -
y6*y7*y20 - y2*y7*y21 - y4*y7*y21 + y6*y7*y21 - y2*y7 + y4*y7 + y2*y19 - y6*y19 - y2*y20 +
y6*y20, y2*y4*y15 - y2*y6*y15 - y4^2*y21 + y4*y6*y21 - y2*y4 + y4^2 + y2*y6 - y4*y6,
```

```
y1*y14*y20^2 - y1*y15*y20*y21 - 2*y1*y14*y20 + y14^2*y20 + y1*y15*y20 - y14*y15*y20 -
y1*y20^2 + y1*y15*y21 + y1*y20*y21 - y14*y20*y21 + y15*y20*y21 + y1*y14 - y14^2 - y1*y15 +
y14*y15 + y1*y20 - y1*y21 + y14*y21 - y15*y21) of Multivariate Polynomial Ring in y1, y2,
y4, y6, y7, y9, y10, y11, y12, y13, y14, y15, y19, y20, y21 over Rational Field
```

```
# y10 + y19 - y21 - 1 => y21 = y10*y19
# y9 - y11 - y19 + y21 => y9*y21 = y11*y19 => y9*y10 = y11

R.<y1,y2,y4,y6,y7,y9,y10,y12,y13,y14,y15,y19,y20>=QQ[];

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

```
[     1      0      1      0      1      0      1      0]
[     1      0      1      0      0      1     y9      0]
[     1      0      0      1      1      0    y10      0]
[     1      0      0      1      0      1  y9*y10      0]
[     0      1      1      0     y1      0    y12      0]
[     0      1      1      0      0     y1    y13      0]
[     0      1      0      1     y1      0    y14      0]
[     0      1      0      1      0     y1    y15      0]
[    y2      0     y4      0     y6      0      0      1]
[    y2      0     y4      0      0     y6      0     y9]
[    y2      0      0     y4     y6      0      0    y10]
[    y2      0      0     y4      0     y6      0  y9*y10]
[     0     y2     y4      0  y6*y7      0      0    y19]
[     0     y2     y4      0      0  y6*y7      0    y20]
[     0     y2      0     y4  y6*y7      0      0 y10*y19]
[     0     y2      0     y4      0  y6*y7      0 y10*y20]
Ideal (y12 - y13 - y14 + y15, y10 + y13 - y15 - 1, y13*y20 - y15*y20 - y13 + y15, y13*y19
- y15*y19 - y13 + y15, y4*y19 - y6*y19 - y4*y20 + y6*y20, y13*y14 - y13*y15 - y14*y15 +
y15^2, y9*y14 - y9*y19 + y15*y19 - y14*y20 - y15 + y20, y7*y14 - y7*y15 - y1*y19 + y1*y20,
```

```
y4*y14 - y6*y14 - y4*y15 + y6*y15, y13^2 - y13*y15 - y13 + y15, y9*y13 - y9*y15 - y13 +
y15, y4*y13 - y6*y13 - y4*y15 + y6*y15, y2*y13 + y2*y14 - y2*y15 - y6*y19 - y4*y20 +
y6*y20 - y2 + y4, y7*y9 - y7 + y19 - y20, y4*y9 - y6*y9 - y4 + y6, y2*y9 - y6*y9 + y2*y14
- y2*y15 - y6*y19 + y6*y20 - y2 + y6, y1*y9 - y1 + y14 - y15, y1*y2 - y6*y7 - y4*y20 +
y6*y20 - y2 + y4, y2*y14*y20 - y6*y19*y20 - y4*y20^2 + y6*y20^2 - y2*y14 + y6*y19 - y2*y20
+ 2*y4*y20 - y6*y20 + y2 - y4, y7*y15*y19 - y7*y15*y20 - y1*y19*y20 + y1*y20^2 + y1*y19 -
y7*y19 - y14*y19 + y19^2 - y1*y20 + y7*y20 + y14*y20 - y19*y20, y1*y15*y19 - y1*y14*y20 +
y1*y14 - y14^2 - y1*y15 + y14*y15 - y1*y19 + y14*y19 - y15*y19 + y1*y20, y2*y14*y19 -
y6*y19^2 - y4*y20^2 + y6*y20^2 - y2*y14 - y2*y19 + 2*y6*y19 + 2*y4*y20 - 2*y6*y20 + y2 -
y4, y2*y4*y15 - y2*y6*y15 - y4^2*y20 + y4*y6*y20 - y2*y4 + y4^2 + y2*y6 - y4*y6, y2*y14^2
- y2*y14*y15 - y6*y14*y19 + y6*y15*y19 - y2*y14 + y6*y14 + y2*y15 - y6*y15, y6*y7*y13 -
y1*y4*y20 + y4*y15*y20 - y6*y15*y20 + y1*y4 - y6*y7 - y6*y13 - y6*y14 - y4*y15 + 2*y6*y15
+ y6*y19, y6*y7*y15*y20 - y1*y4*y20^2 + y4*y15*y20^2 - y6*y15*y20^2 - y6*y7*y15 +
2*y1*y4*y20 - y6*y7*y20 - y6*y14*y20 - 2*y4*y15*y20 + 2*y6*y15*y20 + y6*y19*y20 - y1*y4 +
y6*y7 + y6*y14 + y4*y15 - y6*y15 - y6*y19, y2*y7*y15*y20 - y4*y7*y20^2 - y2*y7*y15 -
y2*y7*y20 + 2*y4*y7*y20 + y2*y19*y20 - y6*y19*y20 - y2*y20^2 + y6*y20^2 + y2*y7 - y4*y7 -
y2*y19 + y6*y19 + y2*y20 - y6*y20, y4*y6*y7*y15 - y6^2*y7*y15 - y1*y4^2*y20 + y1*y4*y6*y20
+ y4^2*y15*y20 - 2*y4*y6*y15*y20 + y6^2*y15*y20 + y1*y4^2 - y1*y4*y6 - y4*y6*y7 + y6^2*y7
- y4^2*y15 + y4*y6*y15 + y4*y6*y20 - y6^2*y20) of Multivariate Polynomial Ring in y1, y2,
y4, y6, y7, y9, y10, y12, y13, y14, y15, y19, y20 over Rational Field
```

```
# y10 + y13 - y15 - 1 => y15 = y10*y13
# y12 - y13 - y14 + y15 => y12*y15 = y13*y14 => y14 = y10*y12

R.<y1,y2,y4,y6,y7,y9,y10,y12,y13,y19,y20>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,       1,       0],
 [    1,    0,    1,    0,       0,       1,      y9,       0],
 [    1,    0,    0,    1,       1,       0,     y10,       0],
 [    1,    0,    0,    1,       0,       1, y9*y10,       0],
 [    0,    1,    1,    0,      y1,       0,     y12,       0],
 [    0,    1,    1,    0,       0,      y1,     y13,       0],
 [    0,    1,    0,    1,      y1,       0,y10*y12,       0],
 [    0,    1,    0,    1,       0,      y1,y10*y13,       0],
 [   y2,    0,   y4,    0,      y6,       0,       0,       1],
 [   y2,    0,   y4,    0,       0,      y6,       0,      y9],
 [   y2,    0,    0,   y4,      y6,       0,       0,     y10],
 [   y2,    0,    0,   y4,       0,      y6,       0, y9*y10],
 [    0,   y2,   y4,    0,  y6*y7,       0,       0,     y19],
 [    0,   y2,   y4,    0,       0,  y6*y7,       0,     y20],
 [    0,   y2,    0,   y4,  y6*y7,       0,       0,y10*y19],
 [    0,   y2,    0,   y4,       0,  y6*y7,       0,y10*y20]]); M
K=M.matrix_from_columns([0,3,4,5,6,7])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y9*y10*y12*y13*y19*y20)
KK=J.saturation(JJ);
KK[0]
```

```
[     1      0      1      0      1      0      1      0]
[     1      0      1      0      0      1     y9      0]
[     1      0      0      1      1      0    y10      0]
```

```
[      1       0       0       1       0       1  y9*y10       0]
[      0       1       1       0      y1       0     y12       0]
[      0       1       1       0       0      y1     y13       0]
[      0       1       0       1      y1       0 y10*y12       0]
[      0       1       0       1       0      y1 y10*y13       0]
[     y2       0      y4       0      y6       0       0       1]
[     y2       0      y4       0       0      y6       0      y9]
[     y2       0       0      y4      y6       0       0     y10]
[     y2       0       0      y4       0      y6       0  y9*y10]
[      0      y2      y4       0   y6*y7       0       0     y19]
[      0      y2      y4       0       0   y6*y7       0     y20]
[      0      y2       0      y4   y6*y7       0       0 y10*y19]
[      0      y2       0      y4       0   y6*y7       0 y10*y20]
Ideal (y10*y20 - y10 - y20 + 1, y13*y19 - y12*y20, y10*y19 - y10 - y19 + 1, y10*y13 - y10
- y13 + 1, y10*y12 - y10 - y12 + 1, y7*y12 - y7*y13 - y1*y19 + y1*y20, y9*y10 - y9 - y10 +
1, y7*y9 - y7 + y19 - y20, y1*y9 - y1 + y12 - y13, y2*y9*y19 - y6*y9*y19 - y2*y20 +
y6*y20, y2*y7*y19 - y6*y7*y19 - y2*y19^2 + y6*y19^2 - y2*y7*y20 + y6*y7*y20 + y2*y19*y20 -
y6*y19*y20, y1*y2*y19 - y1*y6*y19 - y2*y12*y19 + y6*y12*y19 - y1*y2*y20 + y1*y6*y20 +
y2*y12*y20 - y6*y12*y20, y6*y7*y13 - y1*y2*y20 + y2*y12*y20 - y6*y12*y20, y2*y9*y12 -
y6*y9*y12 - y2*y13 + y6*y13, y1*y2*y12 - y1*y6*y12 - y2*y12^2 + y6*y12^2 - y1*y2*y13 +
y1*y6*y13 + y2*y12*y13 - y6*y12*y13, y1*y4*y10 - y6*y7*y10 - y1*y4 + y6*y7, y1*y2*y10 -
y6*y7*y10 - y1*y2 + y6*y7 - y2*y10 + y6*y10 + y2 - y6, y2*y6*y7*y10 - y4*y6*y7*y10 -
y2*y6*y7 + y4*y6*y7 - y2*y4*y10 + y4*y6*y10 + y2*y4 - y4*y6) of Multivariate Polynomial
Ring in y1, y2, y4, y6, y7, y9, y10, y12, y13, y19, y20 over Rational Field
```

```
# y1*y9 - y1 + y12 - y13 => y13 = y9*y12
# y7*y9 - y7 + y19 - y20 => y20 = y9*y19

R.<y1,y2,y4,y6,y7,y9,y10,y12,y19>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,          1,          0],
 [    1,    0,    1,    0,       0,       1,         y9,          0],
 [    1,    0,    0,    1,       1,       0,        y10,          0],
 [    1,    0,    0,    1,       0,       1,     y9*y10,          0],
 [    0,    1,    1,    0,      y1,       0,        y12,          0],
 [    0,    1,    1,    0,       0,      y1,     y9*y12,          0],
 [    0,    1,    0,    1,      y1,       0,    y10*y12,          0],
 [    0,    1,    0,    1,       0,      y1,y10*y9*y12,          0],
 [   y2,    0,   y4,    0,      y6,       0,          0,          1],
 [   y2,    0,   y4,    0,       0,      y6,          0,         y9],
 [   y2,    0,    0,   y4,      y6,       0,          0,        y10],
 [   y2,    0,    0,   y4,       0,      y6,          0,     y9*y10],
 [    0,   y2,   y4,    0,  y6*y7,       0,          0,        y19],
 [    0,   y2,   y4,    0,       0,  y6*y7,          0,     y9*y19],
 [    0,   y2,    0,   y4,  y6*y7,       0,          0,    y10*y19],
 [    0,   y2,    0,   y4,       0,  y6*y7,          0,y10*y9*y19]]); M
K=M.matrix_from_columns([1,2,3,4,5,6,7])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y9*y10*y12*y19)
KK=J.saturation(JJ);
KK[0]
```

```
[        1         0         1         0         1         0         1         0]
[        1         0         1         0         0         1        y9         0]
[        1         0         0         1         1         0       y10         0]
[        1         0         0         1         0         1     y9*y10         0]
[        0         1         1         0        y1         0       y12         0]
[        0         1         1         0         0        y1     y9*y12         0]
[        0         1         0         1        y1         0    y10*y12         0]
[        0         1         0         1         0        y1 y9*y10*y12         0]
[       y2         0        y4         0        y6         0         0         1]
[       y2         0        y4         0         0        y6         0        y9]
[       y2         0         0        y4        y6         0         0       y10]
[       y2         0         0        y4         0        y6         0     y9*y10]
[        0        y2        y4         0     y6*y7         0         0       y19]
[        0        y2        y4         0         0     y6*y7         0     y9*y19]
[        0        y2         0        y4     y6*y7         0         0    y10*y19]
[        0        y2         0        y4         0     y6*y7         0 y9*y10*y19]
Ideal (y10*y19 - y10 - y19 + 1, y10*y12 - y10 - y12 + 1, y2*y12 - y4*y19 - y2 + y4, y9*y10
- y9 - y10 + 1, y4*y10 - y6*y10 - y4 + y6, y7*y9 - y9*y19 - y7 + y19, y4*y9 - y6*y9 - y4 +
y6, y1*y9 - y9*y12 - y1 + y12, y1*y2 - y6*y7 - y4*y19 + y6*y19 - y2 + y4, y6*y7*y12 -
y1*y4*y19 + y4*y12*y19 - y6*y12*y19 + y1*y4 - y6*y7 - y4*y12 + y6*y19) of Multivariate
Polynomial Ring in y1, y2, y4, y6, y7, y9, y10, y12, y19 over Rational Field
```

```
# y2*y12 - y4*y19 - y2 + y4 => y19 = y12

R.<y1,y2,y4,y6,y7,y9,y10,y12>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,          1,          0],
 [    1,    0,    1,    0,       0,       1,         y9,          0],
 [    1,    0,    0,    1,       1,       0,        y10,          0],
 [    1,    0,    0,    1,       0,       1,    y9*y10,          0],
 [    0,    1,    1,    0,      y1,       0,        y12,          0],
 [    0,    1,    1,    0,       0,      y1,    y9*y12,          0],
 [    0,    1,    0,    1,      y1,       0,   y10*y12,          0],
 [    0,    1,    0,    1,       0,      y1,y10*y9*y12,          0],
 [   y2,    0,   y4,    0,      y6,       0,          0,          1],
 [   y2,    0,   y4,    0,       0,      y6,          0,         y9],
 [   y2,    0,    0,   y4,      y6,       0,          0,        y10],
 [   y2,    0,    0,   y4,       0,      y6,          0,    y9*y10],
 [    0,   y2,   y4,    0,  y6*y7,       0,          0,        y12],
 [    0,   y2,   y4,    0,       0,  y6*y7,          0,    y9*y12],
 [    0,   y2,    0,   y4,  y6*y7,       0,          0,   y10*y12],
 [    0,   y2,    0,   y4,       0,  y6*y7,          0,y10*y9*y12]]); M
K=M.matrix_from_columns([0,1,2,3,4,5,6])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y9*y10*y12)
KK=J.saturation(JJ);
KK[0]
```

```
[        1         0         1         0         1         0         1         0]
[        1         0         1         0         0         1        y9         0]
[        1         0         0         1         1         0       y10         0]
```

```
[         1          0          0          1          0          1     y9*y10          0]
[         0          1          1          0         y1          0        y12          0]
[         0          1          1          0          0         y1     y9*y12          0]
[         0          1          0          1         y1          0    y10*y12          0]
[         0          1          0          1          0         y1 y9*y10*y12          0]
[        y2          0         y4          0         y6          0          0          1]
[        y2          0         y4          0          0         y6          0         y9]
[        y2          0          0         y4         y6          0          0        y10]
[        y2          0          0         y4          0         y6          0     y9*y10]
[         0         y2         y4          0      y6*y7          0          0        y12]
[         0         y2         y4          0          0      y6*y7          0     y9*y12]
[         0         y2          0         y4      y6*y7          0          0    y10*y12]
[         0         y2          0         y4          0      y6*y7          0 y9*y10*y12]
```

```
Ideal (y2*y12 - y4*y12 - y2 + y4, y4*y10 - y6*y10 - y4 + y6, y2*y10 - y6*y10 - y2 + y6,
y1*y10 - y7*y10 - y1 + y7, y4*y9 - y6*y9 - y4 + y6, y2*y9 - y6*y9 - y2 + y6, y1*y9 - y7*y9
- y1 + y7, y1*y2 - y6*y7 - y4*y12 + y6*y12 - y2 + y4, y1*y4*y12 - y6*y7*y12 - y4*y12^2 +
y6*y12^2 - y1*y4 + y6*y7 + y4*y12 - y6*y12) of Multivariate Polynomial Ring in y1, y2, y4,
y6, y7, y9, y10, y12 over Rational Field
```

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

```
[Ideal (y4 - y6, y2 - y6, y1 - y7) of Multivariate Polynomial Ring in y1, y2, y4, y6, y7,
y9, y10, y12 over Rational Field, Ideal (y12 - 1, y10 - 1, y9 - 1, y1*y2 - y6*y7 - y2 +
y6) of Multivariate Polynomial Ring in y1, y2, y4, y6, y7, y9, y10, y12 over Rational
Field, Ideal (y10 - 1, y9 - 1, y2 - y4, y1*y4 - y6*y7 - y4*y12 + y6*y12) of Multivariate
Polynomial Ring in y1, y2, y4, y6, y7, y9, y10, y12 over Rational Field]
```

```
#thus we three subcases
# 1.1) y4 - y6, y2 - y6, y1 - y7
# 1.2) y12 - 1, y10 - 1, y9 - 1, y1*y2 - y6*y7 - y2 + y6
# 1.3) y10 - 1, y9 - 1, y2 - y4, y1*y4 - y6*y7 - y4*y12 + y6*y12


# 1.1)
# y4 - y6, y2 - y6, y1 - y7

R.<y1,y2,y9,y10,y12>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,          1,          0],
 [    1,    0,    1,    0,       0,       1,         y9,          0],
 [    1,    0,    0,    1,       1,       0,        y10,          0],
 [    1,    0,    0,    1,       0,       1,     y9*y10,          0],
 [    0,    1,    1,    0,      y1,       0,        y12,          0],
 [    0,    1,    1,    0,       0,      y1,     y9*y12,          0],
 [    0,    1,    0,    1,      y1,       0,    y10*y12,          0],
 [    0,    1,    0,    1,       0,      y1,y10*y9*y12,          0],
 [   y2,    0,   y2,    0,      y2,       0,          0,          1],
 [   y2,    0,   y2,    0,       0,      y2,          0,         y9],
 [   y2,    0,    0,   y2,      y2,       0,          0,        y10],
 [   y2,    0,    0,   y2,       0,      y2,          0,     y9*y10],
 [    0,   y2,   y2,    0,  y2*y1,       0,          0,        y12],
 [    0,   y2,   y2,    0,       0,  y2*y1,          0,     y9*y12],
```

```
 [    0,    y2,    0,    y2,  y2*y1,       0,          0,    y10*y12],
 [    0,    y2,    0,    y2,       0,  y2*y1,          0,y10*y9*y12]]); M
J=ideal(M.minors(6));
JJ=ideal(y1*y2*y9*y10*y12)
KK=J.saturation(JJ);
KK[0].primary_decomposition();
```

```
[        1         0         1         0         1         0         1         0]
[        1         0         1         0         0         1        y9         0]
[        1         0         0         1         1         0       y10         0]
[        1         0         0         1         0         1     y9*y10         0]
[        0         1         1         0        y1         0       y12         0]
[        0         1         1         0         0        y1     y9*y12         0]
[        0         1         0         1        y1         0    y10*y12         0]
[        0         1         0         1         0        y1 y9*y10*y12         0]
[       y2         0        y2         0        y2         0         0         1]
[       y2         0        y2         0         0        y2         0        y9]
[       y2         0         0        y2        y2         0         0       y10]
[       y2         0         0        y2         0        y2         0     y9*y10]
[        0        y2        y2         0      y1*y2         0         0       y12]
[        0        y2        y2         0         0      y1*y2         0     y9*y12]
[        0        y2         0        y2      y1*y2         0         0    y10*y12]
[        0        y2         0        y2         0      y1*y2         0 y9*y10*y12]
[Ideal (y10 - 1, y9 - 1) of Multivariate Polynomial Ring in y1, y2, y9, y10, y12 over
Rational Field, Ideal (y10 - 1, y1 - y12) of Multivariate Polynomial Ring in y1, y2, y9,
y10, y12 over Rational Field, Ideal (y12 - 1, y9 - 1) of Multivariate Polynomial Ring in
y1, y2, y9, y10, y12 over Rational Field]
```

```
#  1.1) results

# for y10 - 1, y9 - 1 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,    1,    0],
 [    1,    0,    0,    1,       1,       0,    1,    0],
 [    1,    0,    0,    1,       0,       1,    1,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y12,    0],
 [    0,    1,    0,    1,      y1,       0,  y12,    0],
 [    0,    1,    0,    1,       0,      y1,  y12,    0],
 [   y2,    0,   y2,    0,      y2,       0,    0,    1],
 [   y2,    0,   y2,    0,       0,      y2,    0,    1],
 [   y2,    0,    0,   y2,      y2,       0,    0,    1],
 [   y2,    0,    0,   y2,       0,      y2,    0,    1],
 [    0,   y2,   y2,    0,  y2*y1,       0,    0,  y12],
 [    0,   y2,   y2,    0,       0,  y2*y1,    0,  y12],
 [    0,   y2,    0,   y2,  y2*y1,       0,    0,  y12],
 [    0,   y2,    0,   y2,       0,  y2*y1,    0,  y12]]);

diagonal_matrix([1,1,1,1,1,1,1,1,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2\
   ])*M*diagonal_matrix([1,1,1,1,1,1,1,y2])
```

```
[  1   0   1   0   1   0   1   0]
[  1   0   1   0   0   1   1   0]
[  1   0   0   1   1   0   1   0]
[  1   0   0   1   0   1   1   0]
[  0   1   1   0  y1   0 y12   0]
[  0   1   1   0   0  y1 y12   0]
[  0   1   0   1  y1   0 y12   0]
[  0   1   0   1   0  y1 y12   0]
[  1   0   1   0   1   0   0   1]
[  1   0   1   0   0   1   0   1]
[  1   0   0   1   1   0   0   1]
[  1   0   0   1   0   1   0   1]
[  0   1   1   0  y1   0   0 y12]
[  0   1   1   0   0  y1   0 y12]
[  0   1   0   1  y1   0   0 y12]
[  0   1   0   1   0  y1   0 y12]
```

```
# for y10 - 1, y1 - y12 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,    1,    0],
 [    1,    0,    0,    1,       0,       1,   y9,    0],
 [    0,    1,    1,    0,      y1,       0,   y1,    0],
 [    0,    1,    1,    0,       0,     y1,y9*y1,    0],
 [    0,    1,    0,    1,      y1,       0,   y1,    0],
 [    0,    1,    0,    1,       0,     y1,y9*y1,    0],
 [   y2,    0,   y2,    0,      y2,       0,    0,    1],
 [   y2,    0,   y2,    0,       0,      y2,    0,   y9],
 [   y2,    0,    0,   y2,      y2,       0,    0,    1],
 [   y2,    0,    0,   y2,       0,      y2,    0,   y9],
 [    0,   y2,   y2,    0,  y2*y1,       0,    0,   y1],
 [    0,   y2,   y2,    0,       0,  y2*y1,    0,y9*y1],
 [    0,   y2,    0,   y2,  y2*y1,       0,    0,   y1],
 [    0,   y2,    0,   y2,       0,  y2*y1,    0,y9*y1]]);

(diagonal_matrix([1,1,1,1,1/y1,1/y1,1/y1,1/y1,1/y2,1/y2,1/y2,1/y2,1/(y2*\
   y1),1/(y2*y1),1/(y2*y1),1/(y2*y1)])*M*diagonal_matrix([1,y1,1,1,1,1,1,\
   y2])).matrix_from_rows_and_columns\
   ([0,2,1,3,4,6,5,7,8,10,9,11,12,14,13,15],[0,1,4,5,2,3,6,7])
```

```
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1    1    0]
[   1    0    0    1    1    0   y9    0]
[   1    0    0    1    0    1   y9    0]
[   0    1    1    0 1/y1    0    1    0]
[   0    1    1    0    0 1/y1    1    0]
[   0    1    0    1 1/y1    0   y9    0]
[   0    1    0    1    0 1/y1   y9    0]
[   1    0    1    0    1    0    0    1]
```

```
[   1    0    1    0    0    1    0    1]
[   1    0    0    1    1    0    0   y9]
[   1    0    0    1    0    1    0   y9]
[   0    1    1    0 1/y1    0    0    1]
[   0    1    1    0    0 1/y1    0    1]
[   0    1    0    1 1/y1    0    0   y9]
[   0    1    0    1    0 1/y1    0   y9]
```

```
# for y12 - 1, y9 - 1 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,      1,    0],
 [    1,    0,    1,    0,       0,       1,      1,    0],
 [    1,    0,    0,    1,       1,       0,    y10,    0],
 [    1,    0,    0,    1,       0,       1,    y10,    0],
 [    0,    1,    1,    0,      y1,       0,      1,    0],
 [    0,    1,    1,    0,       0,      y1,      1,    0],
 [    0,    1,    0,    1,      y1,       0,    y10,    0],
 [    0,    1,    0,    1,       0,      y1,    y10,    0],
 [   y2,    0,   y2,    0,      y2,       0,      0,    1],
 [   y2,    0,   y2,    0,       0,      y2,      0,    1],
 [   y2,    0,    0,   y2,      y2,       0,      0,  y10],
 [   y2,    0,    0,   y2,       0,      y2,      0,  y10],
 [    0,   y2,   y2,    0,  y2*y1,       0,      0,    1],
 [    0,   y2,   y2,    0,       0,  y2*y1,      0,    1],
 [    0,   y2,    0,   y2,  y2*y1,       0,      0,  y10],
 [    0,   y2,    0,   y2,       0,  y2*y1,      0,  y10]]);

diagonal_matrix([1,1,1,1,1,1,1,1,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2\
   ])*M*diagonal_matrix([1,1,1,1,1,1,1,y2])
```

```
[  1   0   1   0   1   0   1   0]
[  1   0   1   0   0   1   1   0]
[  1   0   0   1   1   0 y10   0]
[  1   0   0   1   0   1 y10   0]
[  0   1   1   0  y1   0   1   0]
[  0   1   1   0   0  y1   1   0]
[  0   1   0   1  y1   0 y10   0]
[  0   1   0   1   0  y1 y10   0]
[  1   0   1   0   1   0   0   1]
[  1   0   1   0   0   1   0   1]
[  1   0   0   1   1   0   0 y10]
[  1   0   0   1   0   1   0 y10]
[  0   1   1   0  y1   0   0   1]
[  0   1   1   0   0  y1   0   1]
[  0   1   0   1  y1   0   0 y10]
[  0   1   0   1   0  y1   0 y10]
```

```
# 1.2)
# y12 - 1, y10 - 1, y9 - 1
# y1*y2 - y6*y7 - y2 + y6 => y7 = y1
```

```
R.<y1,y2,y4,y6>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,       1,      0],
 [    1,    0,    1,    0,       0,       1,       1,      0],
 [    1,    0,    0,    1,       1,       0,       1,      0],
 [    1,    0,    0,    1,       0,       1,       1,      0],
 [    0,    1,    1,    0,      y1,       0,       1,      0],
 [    0,    1,    1,    0,       0,      y1,       1,      0],
 [    0,    1,    0,    1,      y1,       0,       1,      0],
 [    0,    1,    0,    1,       0,      y1,       1,      0],
 [   y2,    0,   y4,    0,      y6,       0,       0,      1],
 [   y2,    0,   y4,    0,       0,      y6,       0,      1],
 [   y2,    0,    0,   y4,      y6,       0,       0,      1],
 [   y2,    0,    0,   y4,       0,      y6,       0,      1],
 [    0,   y2,   y4,    0,  y6*y1,       0,       0,      1],
 [    0,   y2,   y4,    0,       0,  y6*y1,       0,      1],
 [    0,   y2,    0,   y4,  y6*y1,       0,       0,      1],
 [    0,   y2,    0,   y4,       0,  y6*y1,       0,      1]]); M
J=ideal(M.minors(6));
JJ=ideal(y1*y2*y4*y6)
KK=J.saturation(JJ);
KK[0].primary_decomposition();
```

```
[    1     0     1     0     1     0     1     0]
[    1     0     1     0     0     1     1     0]
[    1     0     0     1     1     0     1     0]
[    1     0     0     1     0     1     1     0]
[    0     1     1     0    y1     0     1     0]
[    0     1     1     0     0    y1     1     0]
[    0     1     0     1    y1     0     1     0]
[    0     1     0     1     0    y1     1     0]
[   y2     0    y4     0    y6     0     0     1]
[   y2     0    y4     0     0    y6     0     1]
[   y2     0     0    y4    y6     0     0     1]
[   y2     0     0    y4     0    y6     0     1]
[    0    y2    y4     0 y1*y6     0     0     1]
[    0    y2    y4     0     0 y1*y6     0     1]
[    0    y2     0    y4 y1*y6     0     0     1]
[    0    y2     0    y4     0 y1*y6     0     1]
```

```
#  1.2) results

# for y2 - y6 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,       1,      0],
 [    1,    0,    1,    0,       0,       1,       1,      0],
 [    1,    0,    0,    1,       1,       0,       1,      0],
 [    1,    0,    0,    1,       0,       1,       1,      0],
 [    0,    1,    1,    0,      y1,       0,       1,      0],
 [    0,    1,    1,    0,       0,      y1,       1,      0],
```

```
 [    0,    1,    0,    1,      y1,       0,       1,      0],
 [    0,    1,    0,    1,       0,      y1,       1,      0],
 [   y2,    0,   y4,    0,      y2,       0,       0,      1],
 [   y2,    0,   y4,    0,       0,      y2,       0,      1],
 [   y2,    0,    0,   y4,      y2,       0,       0,      1],
 [   y2,    0,    0,   y4,       0,      y2,       0,      1],
 [    0,   y2,   y4,    0,  y2*y1,       0,       0,      1],
 [    0,   y2,   y4,    0,       0,  y2*y1,       0,      1],
 [    0,   y2,    0,   y4,  y2*y1,       0,       0,      1],
 [    0,   y2,    0,   y4,       0,  y2*y1,       0,      1]]);


(diagonal_matrix([1,1,1,1,1,1,1,1,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2\
   ])*M*diagonal_matrix([1,1,1,1,1,1,1,y2])).matrix_from_rows_and_columns\
   ([0,1,8,9,4,5,12,13,2,3,10,11,6,7,14,15],[0,1,6,7,4,5,2,3])
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1    1    0]
[   1    0    0    1    1    0 y4/y2    0]
[   1    0    0    1    0    1 y4/y2    0]
[   0    1    1    0   y1    0    1    0]
[   0    1    1    0    0   y1    1    0]
[   0    1    0    1   y1    0 y4/y2    0]
[   0    1    0    1    0   y1 y4/y2    0]
[   1    0    1    0    1    0    0    1]
[   1    0    1    0    0    1    0    1]
[   1    0    0    1    1    0    0 y4/y2]
[   1    0    0    1    0    1    0 y4/y2]
[   0    1    1    0   y1    0    0    1]
[   0    1    1    0    0   y1    0    1]
[   0    1    0    1   y1    0    0 y4/y2]
[   0    1    0    1    0   y1    0 y4/y2]
```

```
# for y1 - 1 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,       1,      0],
 [    1,    0,    1,    0,       0,       1,       1,      0],
 [    1,    0,    0,    1,       1,       0,       1,      0],
 [    1,    0,    0,    1,       0,       1,       1,      0],
 [    0,    1,    1,    0,       1,       0,       1,      0],
 [    0,    1,    1,    0,       0,       1,       1,      0],
 [    0,    1,    0,    1,       1,       0,       1,      0],
 [    0,    1,    0,    1,       0,       1,       1,      0],
 [   y2,    0,   y4,    0,      y6,       0,       0,      1],
 [   y2,    0,   y4,    0,       0,      y6,       0,      1],
 [   y2,    0,    0,   y4,      y6,       0,       0,      1],
 [   y2,    0,    0,   y4,       0,      y6,       0,      1],
 [    0,   y2,   y4,    0,      y6,       0,       0,      1],
 [    0,   y2,   y4,    0,       0,      y6,       0,      1],
 [    0,   y2,    0,   y4,      y6,       0,       0,      1],
```

```
 [    0,    y2,    0,    y4,       0,      y6,       0,      1]]);

(diagonal_matrix([1,1,1,1,1,1,1,1,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2\
   ])*M*diagonal_matrix([1,1,1,1,1,1,1,y2])).matrix_from_rows_and_columns\
   ([0,1,4,5,8,9,12,13,2,3,6,7,10,11,14,15],[6,7,0,1,4,5,2,3])
[    1     0     1     0     1     0     1     0]
[    1     0     1     0     0     1     1     0]
[    1     0     0     1     1     0     1     0]
[    1     0     0     1     0     1     1     0]
[    0     1     1     0 y6/y2     0 y4/y2     0]
[    0     1     1     0     0 y6/y2 y4/y2     0]
[    0     1     0     1 y6/y2     0 y4/y2     0]
[    0     1     0     1     0 y6/y2 y4/y2     0]
[    1     0     1     0     1     0     0     1]
[    1     0     1     0     0     1     0     1]
[    1     0     0     1     1     0     0     1]
[    1     0     0     1     0     1     0     1]
[    0     1     1     0 y6/y2     0     0 y4/y2]
[    0     1     1     0     0 y6/y2     0 y4/y2]
[    0     1     0     1 y6/y2     0     0 y4/y2]
[    0     1     0     1     0 y6/y2     0 y4/y2]
```

```
#  1.3)

#  y10 - 1, y9 - 1, y2 - y4,
#  y1*y4 - y6*y7 - y4*y12 + y6*y12 => y7 = y1

R.<y1,y2,y6,y12>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,        1,    0],
 [    1,    0,    1,    0,       0,       1,        1,    0],
 [    1,    0,    0,    1,       1,       0,        1,    0],
 [    1,    0,    0,    1,       0,       1,        1,    0],
 [    0,    1,    1,    0,      y1,       0,      y12,    0],
 [    0,    1,    1,    0,       0,      y1,      y12,    0],
 [    0,    1,    0,    1,      y1,       0,      y12,    0],
 [    0,    1,    0,    1,       0,      y1,      y12,    0],
 [   y2,    0,   y2,    0,      y6,       0,        0,    1],
 [   y2,    0,   y2,    0,       0,      y6,        0,    1],
 [   y2,    0,    0,   y2,      y6,       0,        0,    1],
 [   y2,    0,    0,   y2,       0,      y6,        0,    1],
 [    0,   y2,   y2,    0,  y6*y1,       0,        0,  y12],
 [    0,   y2,   y2,    0,       0,  y6*y1,        0,  y12],
 [    0,   y2,    0,   y2,  y6*y1,       0,        0,  y12],
 [    0,   y2,    0,   y2,       0,  y6*y1,        0,  y12]]); M
J=ideal(M.minors(6));
JJ=ideal(y1*y2*y6*y12)
KK=J.saturation(JJ);
KK[0].primary_decomposition();
[    1     0     1     0     1     0     1     0]
```

```
[    1     0     1     0     0     1     1     0]
[    1     0     0     1     1     0     1     0]
[    1     0     0     1     0     1     1     0]
[    0     1     1     0    y1     0   y12     0]
[    0     1     1     0     0    y1   y12     0]
[    0     1     0     1    y1     0   y12     0]
[    0     1     0     1     0    y1   y12     0]
[   y2     0    y2     0    y6     0     0     1]
[   y2     0    y2     0     0    y6     0     1]
[   y2     0     0    y2    y6     0     0     1]
[   y2     0     0    y2     0    y6     0     1]
[    0    y2    y2     0 y1*y6     0     0   y12]
[    0    y2    y2     0     0 y1*y6     0   y12]
[    0    y2     0    y2 y1*y6     0     0   y12]
[    0    y2     0    y2     0 y1*y6     0   y12]
[Ideal (y2 - y6) of Multivariate Polynomial Ring in y1, y2, y6, y12 over Rational Field,
Ideal (y1 - y12) of Multivariate Polynomial Ring in y1, y2, y6, y12 over Rational Field]
```

```
#  1.3) results

# for y2 - y6 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,        1,    0],
 [    1,    0,    1,    0,       0,       1,        1,    0],
 [    1,    0,    0,    1,       1,       0,        1,    0],
 [    1,    0,    0,    1,       0,       1,        1,    0],
 [    0,    1,    1,    0,      y1,       0,      y12,    0],
 [    0,    1,    1,    0,       0,      y1,      y12,    0],
 [    0,    1,    0,    1,      y1,       0,      y12,    0],
 [    0,    1,    0,    1,       0,      y1,      y12,    0],
 [   y2,    0,   y2,    0,      y2,       0,        0,    1],
 [   y2,    0,   y2,    0,       0,      y2,        0,    1],
 [   y2,    0,    0,   y2,      y2,       0,        0,    1],
 [   y2,    0,    0,   y2,       0,      y2,        0,    1],
 [    0,   y2,   y2,    0,  y2*y1,       0,        0,  y12],
 [    0,   y2,   y2,    0,       0,  y2*y1,        0,  y12],
 [    0,   y2,    0,   y2,  y2*y1,       0,        0,  y12],
 [    0,   y2,    0,   y2,       0,  y2*y1,        0,  y12]]);

(diagonal_matrix([1,1,1,1,1,1,1,1,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2,1/y2\
   ])*M*diagonal_matrix([1,1,1,1,1,1,1,y2]))
```

```
[  1   0   1   0   1   0   1   0]
[  1   0   1   0   0   1   1   0]
[  1   0   0   1   1   0   1   0]
[  1   0   0   1   0   1   1   0]
[  0   1   1   0  y1   0 y12   0]
[  0   1   1   0   0  y1 y12   0]
[  0   1   0   1  y1   0 y12   0]
[  0   1   0   1   0  y1 y12   0]
[  1   0   1   0   1   0   0   1]
```

```
[  1   0   1   0   0   1   0   1]
[  1   0   0   1   1   0   0   1]
[  1   0   0   1   0   1   0   1]
[  0   1   1   0  y1   0   0 y12]
[  0   1   1   0   0  y1   0 y12]
[  0   1   0   1  y1   0   0 y12]
[  0   1   0   1   0  y1   0 y12]
```

```
# for y1 - y12 the slack matrix can be parametrized as follows

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,       1,    0],
 [    1,    0,    1,    0,       0,       1,       1,    0],
 [    1,    0,    0,    1,       1,       0,       1,    0],
 [    1,    0,    0,    1,       0,       1,       1,    0],
 [    0,    1,    1,    0,      y1,       0,      y1,    0],
 [    0,    1,    1,    0,       0,      y1,      y1,    0],
 [    0,    1,    0,    1,      y1,       0,      y1,    0],
 [    0,    1,    0,    1,       0,      y1,      y1,    0],
 [   y2,    0,   y2,    0,      y6,       0,       0,    1],
 [   y2,    0,   y2,    0,       0,      y6,       0,    1],
 [   y2,    0,    0,   y2,      y6,       0,       0,    1],
 [   y2,    0,    0,   y2,       0,      y6,       0,    1],
 [    0,   y2,   y2,    0,  y6*y1,       0,       0,   y1],
 [    0,   y2,   y2,    0,       0,  y6*y1,       0,   y1],
 [    0,   y2,    0,   y2,  y6*y1,       0,       0,   y1],
 [    0,   y2,    0,   y2,       0,  y6*y1,       0,   y1]]);


(diagonal_matrix([1,1,1,1,1/y1,1/y1,1/y1,1/y1,1/y2,1/y2,1/y2,1/y2,1/(y2*\
   y1),1/(y2*y1),1/(y2*y1),1/(y2*y1)])*M*diagonal_matrix([1,y1,1,1,1,1,1,\
   y2])).matrix_from_rows_and_columns\
   ([0,2,8,10,4,6,12,14,1,3,9,11,5,7,13,15],[6,7,0,1,2,3,4,5])
```

```
[    1     0     1     0     1     0     1     0]
[    1     0     1     0     0     1     1     0]
[    0     1     1     0     1     0 y6/y2     0]
[    0     1     1     0     0     1 y6/y2     0]
[    1     0     0     1  1/y1     0     1     0]
[    1     0     0     1     0  1/y1     1     0]
[    0     1     0     1  1/y1     0 y6/y2     0]
[    0     1     0     1     0  1/y1 y6/y2     0]
[    1     0     1     0     1     0     0     1]
[    1     0     1     0     0     1     0     1]
[    0     1     1     0     1     0     0 y6/y2]
[    0     1     1     0     0     1     0 y6/y2]
[    1     0     0     1  1/y1     0     0     1]
[    1     0     0     1     0  1/y1     0     1]
[    0     1     0     1  1/y1     0     0 y6/y2]
[    0     1     0     1     0  1/y1     0 y6/y2]
```

```
# case 2)

R.<y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20\
   ,y21,y22>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
 [   y2,    0,    0,   y5,  y6*y7,       0,    0,  y17],
 [   y2,    0,    0,   y5,       0,  y8*y7,    0,  y18],
 [    0,   y3,   y4,    0,      y6,       0,    0,  y19],
 [    0,   y3,   y4,    0,       0,      y8,    0,  y20],
 [    0,   y3,    0,   y5,  y6*y7,       0,    0,  y21],
 [    0,   y3,    0,   y5,       0,  y8*y7,    0,  y22]]); M
K=M.matrix_from_columns([0,1,2,3,4,5,7])
J=ideal(K.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*y22)
KK=J.saturation(JJ)
KK
```

```
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1   y9    0]
[   1    0    0    1    1    0  y10    0]
[   1    0    0    1    0    1  y11    0]
[   0    1    1    0   y1    0  y12    0]
[   0    1    1    0    0   y1  y13    0]
[   0    1    0    1   y1    0  y14    0]
[   0    1    0    1    0   y1  y15    0]
[  y2    0   y4    0   y6    0    0    1]
[  y2    0   y4    0    0   y8    0  y16]
[  y2    0    0   y5 y6*y7    0    0  y17]
[  y2    0    0   y5    0 y7*y8    0  y18]
[   0   y3   y4    0   y6    0    0  y19]
[   0   y3   y4    0    0   y8    0  y20]
[   0   y3    0   y5 y6*y7    0    0  y21]
[   0   y3    0   y5    0 y7*y8    0  y22]
(Ideal (y2*y22 - y4*y22 - y3 + y5, y4*y21 - y5*y21 - y4*y22 + y5*y22, y3*y21 - y5*y21 -
y3*y22 + y5*y22, y2*y21 - y5*y21 - y4*y22 + y5*y22 - y3 + y5, y3*y20 - y5*y20 - y3*y22 +
y4*y22, y2*y20 - y4*y20 - y3 + y4, y5*y19 - y6*y19 - y5*y20 + y6*y20 + y6 - y8, y4*y19 -
y6*y19 - y4*y20 + y6*y20 + y6 - y8, y3*y19 - y6*y19 - y5*y20 + y6*y20 - y3*y22 + y4*y22 +
y6 - y8, y2*y19 - y6*y19 - y4*y20 + y6*y20 - y3 + y4 + y6 - y8, y7*y18 + y7*y19 - y7*y22 -
y7 - y18 - y19 + y22 + 1, y6*y18 - y8*y18 + y6*y19 - y8*y19 - y6*y22 + y8*y22 - y6 + y8,
y5*y18 - y8*y18 + y6*y19 - y8*y19 + y5*y20 - y6*y20 - y5*y22 + y8*y22 - y5 - y6 + 2*y8,
```

```
y4*y18 - y8*y18 + y6*y19 - y8*y19 + y4*y20 - y6*y20 - y4*y22 + y8*y22 - y4 - y6 + 2*y8,
y3*y18 - y8*y18 + y6*y19 - y8*y19 + y5*y20 - y6*y20 - y4*y22 + y8*y22 - y3 - y6 + 2*y8,
y2*y18 - y8*y18 + y6*y19 - y8*y19 + y4*y20 - y6*y20 - y4*y22 + y8*y22 - y2 - y4 + y5 - y6
+ 2*y8, y1*y18 + y1*y19 - y1*y22 - y1 - y18 - y19 + y22 + 1, y7*y17 + y7*y19 - y7*y21 - y7
- y17 - y19 + y21 + 1, y6*y17 - y8*y17 + y6*y19 - y8*y19 - y6*y21 + y8*y21 - y6 + y8,
y5*y17 - y8*y17 + y6*y19 - y8*y19 + y5*y20 - y6*y20 - y5*y21 + y8*y21 - y5 - y6 + 2*y8,
y4*y17 - y8*y17 + y6*y19 - y8*y19 + y4*y20 - y6*y20 - y5*y21 + y8*y21 - y4*y22 + y5*y22 -
y4 - y6 + 2*y8, y3*y17 - y8*y17 + y6*y19 - y8*y19 + y5*y20 - y6*y20 - y5*y21 + y8*y21 -
y4*y22 + y5*y22 - y3 - y6 + 2*y8, y2*y17 - y8*y17 + y6*y19 - y8*y19 + y4*y20 - y6*y20 -
y5*y21 + y8*y21 - y4*y22 + y5*y22 - y2 - y4 + y5 - y6 + 2*y8, y1*y17 + y1*y19 - y1*y21 -
y1 - y17 - y19 + y21 + 1, y7*y16 + y7*y19 - y7*y20 - y7 - y16 - y19 + y20 + 1, y6*y16 -
y8*y16 + y6*y19 - y8*y19 - y6*y20 + y8*y20 - y6 + y8, y5*y16 - y8*y16 + y6*y19 - y8*y19 -
y6*y20 + y8*y20 - y5 - y6 + 2*y8, y4*y16 - y8*y16 + y6*y19 - y8*y19 - y6*y20 + y8*y20 - y4
- y6 + 2*y8, y3*y16 - y8*y16 + y6*y19 - y8*y19 - y6*y20 + y8*y20 - y3 - y6 + 2*y8, y2*y16
- y8*y16 + y6*y19 - y8*y19 - y6*y20 + y8*y20 - y2 - y6 + 2*y8, y1*y16 + y1*y19 - y1*y20 -
y1 - y16 - y19 + y20 + 1, y7*y8 - y4*y20 + y6*y20 + y4*y22 - y6*y22 + y4 - y5 - y8, y6*y7
- y4*y20 + y6*y20 + y5*y21 - y6*y21 + y4*y22 - y5*y22 + y4 - y5 - y8, y4*y6 - y5*y6 -
y4*y8 + y5*y8, y3*y6 - y5*y6 - y3*y8 + y5*y8, y2*y6 - y5*y6 - y2*y8 + y5*y8, y1*y3 -
y4*y20 + y6*y20 - y3 + y4 - y8, y1*y7*y21 + y1*y19*y21 - y7*y19*y21 - y1*y21^2 - y1*y7*y22
- y1*y19*y22 + y7*y19*y22 + y1*y21*y22 - y1*y21 + y21^2 + y1*y22 - y21*y22, y1*y5*y21 -
y5*y20*y21 + y6*y20*y21 - y1*y5*y22 + y5*y20*y22 - y6*y20*y22 - y8*y21 + y8*y22, y5*y6*y20
- y6^2*y20 - y5*y8*y20 + y6*y8*y20 + y5*y8*y21 - y6*y8*y21 - y5*y6*y22 + y6^2*y22 + y6*y8
- y8^2, y1*y5*y20 - y4*y20^2 + y6*y20^2 - y1*y4*y22 + y4*y20*y22 - y6*y20*y22 + y4*y20 -
y5*y20 - y8*y20 + y8*y22, y1*y19^2 - y7*y19^2 - y1*y19*y20 + y7*y19*y20 - y1*y19*y21 +
y1*y20*y21 - y1*y19 + y7*y19 + y1*y20 - y7*y20 + y1*y21 + y19*y21 - y20*y21 - y1*y22 - y21
+ y22, y6*y8*y19 - y8^2*y19 - y6^2*y20 + y6*y8*y20 - y6*y8*y21 + y8^2*y21 + y6^2*y22 -
y6*y8*y22, y1*y8*y19 - y1*y6*y20 - y1*y8*y21 + y1*y6*y22 - y8*y19 + y6*y20 + y8*y21 -
y6*y22, y7^2*y19 - y7^2*y20 - y7*y19 + y7*y20 - y7*y21 + y7*y22 + y21 - y22, y1*y7*y19 -
y1*y7*y20 - y7*y19 + y7*y20 - y1*y21 + y1*y22 + y21 - y22, y1*y6*y19 - y1*y6*y20 - y1*y6 +
y1*y8 - y8*y19 + y6*y20, y1*y5*y6 - y1*y5*y8 + y5*y8*y21 - y6*y8*y21 - y5*y6*y22 +
y6^2*y22, y1*y6^2*y20 - y1*y6*y8*y20 + y1*y6*y8*y21 - y1*y6^2*y22 - y1*y6*y8 + y1*y8^2 -
y8^2*y21 + y6*y8*y22) 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, y22 over Rational Field,
1)
```

```
# y3*y20 - y5*y20 - y3*y22 + y4*y22 =>  y5 = y4
# y2*y20 - y4*y20 - y3 + y4 => y3 = y2

R.<y1,y2,y4,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,\
    y22>=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
```

```
 [   y2,    0,    0,   y4,  y6*y7,       0,    0,  y17],
 [   y2,    0,    0,   y4,       0,  y8*y7,    0,  y18],
 [    0,   y2,   y4,    0,      y6,       0,    0,  y19],
 [    0,   y2,   y4,    0,       0,      y8,    0,  y20],
 [    0,   y2,    0,   y4,  y6*y7,       0,    0,  y21],
 [    0,   y2,    0,   y4,       0,  y8*y7,    0,  y22]]); M
K=M.matrix_from_columns([0,1,2,3,4,5,6])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y7*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*y19*y20\
   *y21*y22)
KK=J.saturation(JJ)
(y2*y14 - y6*y14 - y2*y1 + y6*y7) in KK[0]
```

```
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1   y9    0]
[   1    0    0    1    1    0  y10    0]
[   1    0    0    1    0    1  y11    0]
[   0    1    1    0   y1    0  y12    0]
[   0    1    1    0    0   y1  y13    0]
[   0    1    0    1   y1    0  y14    0]
[   0    1    0    1    0   y1  y15    0]
[  y2    0   y4    0   y6    0    0    1]
[  y2    0   y4    0    0   y8    0  y16]
[  y2    0    0   y4 y6*y7    0    0  y17]
[  y2    0    0   y4    0 y7*y8    0  y18]
[   0   y2   y4    0   y6    0    0  y19]
[   0   y2   y4    0    0   y8    0  y20]
[   0   y2    0   y4 y6*y7    0    0  y21]
[   0   y2    0   y4    0 y7*y8    0  y22]
True
```

```
# y2*y14 - y6*y14 - y2*y1 + y6*y7 => y1 = y7

R.<y1,y2,y4,y6,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,y22\
   >=QQ[];

M = matrix(R,[
 [    1,    0,    1,    0,       1,       0,    1,    0],
 [    1,    0,    1,    0,       0,       1,   y9,    0],
 [    1,    0,    0,    1,       1,       0,  y10,    0],
 [    1,    0,    0,    1,       0,       1,  y11,    0],
 [    0,    1,    1,    0,      y1,       0,  y12,    0],
 [    0,    1,    1,    0,       0,      y1,  y13,    0],
 [    0,    1,    0,    1,      y1,       0,  y14,    0],
 [    0,    1,    0,    1,       0,      y1,  y15,    0],
 [   y2,    0,   y4,    0,      y6,       0,    0,    1],
 [   y2,    0,   y4,    0,       0,      y8,    0,  y16],
 [   y2,    0,    0,   y4,  y6*y1,       0,    0,  y17],
 [   y2,    0,    0,   y4,       0,  y8*y1,    0,  y18],
 [    0,   y2,   y4,    0,      y6,       0,    0,  y19],
```

```
 [    0,   y2,   y4,    0,       0,      y8,    0,  y20],
 [    0,   y2,    0,   y4,  y6*y1,       0,    0,  y21],
 [    0,   y2,    0,   y4,       0,  y8*y1,    0,  y22]]); M
K=M.matrix_from_columns([0,1,2,3,4,5,6])
J=ideal(K.minors(6));
JJ=ideal(y1*y2*y4*y6*y8*y9*y10*y11*y12*y13*y14*y15*y16*y17*y18*y19*y20*\
   y21*y22)
KK=J.saturation(JJ)
(y2*y10-y6*y10+y6*y1-y2) in KK[0]
```

```
[   1    0    1    0    1    0    1    0]
[   1    0    1    0    0    1   y9    0]
[   1    0    0    1    1    0  y10    0]
[   1    0    0    1    0    1  y11    0]
[   0    1    1    0   y1    0  y12    0]
[   0    1    1    0    0   y1  y13    0]
[   0    1    0    1   y1    0  y14    0]
[   0    1    0    1    0   y1  y15    0]
[  y2    0   y4    0   y6    0    0    1]
[  y2    0   y4    0    0   y8    0  y16]
[  y2    0    0   y4 y1*y6    0    0  y17]
[  y2    0    0   y4    0 y1*y8    0  y18]
[   0   y2   y4    0   y6    0    0  y19]
[   0   y2   y4    0    0   y8    0  y20]
[   0   y2    0   y4 y1*y6    0    0  y21]
[   0   y2    0   y4    0 y1*y8    0  y22]
True
```

```
#2
# y2*y10-y6*y10+y6*y1-y2 => y1=1

#due to the condition y1=1, the slack matrices in this case are going to \
   correspond to the slack matrices from the case 1)
```