```
// Computing syzygies

Use R::=QQ[t,x,y,z],PosTo;

I:=Ideal(xy^2, t^4, xz^3, x^4);

M:=Module([xy^2,1,0,0,0],
          [t^4,0,1,0,0],
          [xz^3,0,0,1,0],
  [x^4,0,0,0,1]);



ReducedGBasis(M);

//[Vector(xy^2, 1, 0, 0, 0),
// Vector(x^4, 0, 0, 0, 1),
// Vector(xz^3, 0, 0, 1, 0),
// Vector(t^4, 0, 1, 0, 0),
// Vector(0, z^3, 0, -y^2, 0),
// Vector(0, x^3, 0, 0, -y^2),
// Vector(0, 0, 0, x^3, -z^3),
// Vector(0, t^4, -xy^2, 0, 0),
// Vector(0, 0, xz^3, -t^4, 0),
// Vector(0, 0, x^4, 0, -t^4)]


F1:=Module(
 Vector(z^3, 0, -y^2, 0),
 Vector(x^3, 0, 0, -y^2),
 Vector(0, 0, x^3, -z^3),
 Vector(t^4, -xy^2, 0, 0),
 Vector(0, xz^3, -t^4, 0),
 Vector(0, x^4, 0, -t^4));



ScalarProduct(Gens(I),Tail([0, 0, x^4, 0, -t^4]));

ScalarProduct(Gens(I),Tail([xy^2, 1, 0, 0, 0]));


//////////////////////////////////////////////////////////
//
```

```
////////////////////////////////////////////////


Use R::=QQ[t,x,y,z],PosTo;

I:=Ideal(t^3-x^3,
         t^2x^3 - y^5,
 t^5-y^5,
 t^7-y^7);


M:=Module(Vector(t^3-x^3,1,0,0,0),
          Vector(t^2x^3 - y^5,0,1,0,0),
  Vector(t^5-y^5,0,0,1,0),
  Vector(t^7-y^7,0,0,0,1));

ReducedGBasis(M);

[
Vector(0, 0, 0, t^6 + t^5y + t^4y^2 + t^3y^3 + t^2y^4 + ty^5 + y^6,
       -t^4 - t^3y - t^2y^2 - ty^3 - y^4),
Vector(t^3 - x^3, 1, 0, 0, 0),
Vector(t^2x^3 - y^5, 0, 1, 0, 0),
Vector(0, t^2, 1, -1, 0),
Vector(x^6 - ty^5, -x^3, t, 0, 0),
Vector(t^2y^5 - y^7, 0, 0, -t^2, 1),
Vector(x^3y^5 - ty^7, 0, t^3, -t^3 - x^3, t),
Vector(0, y^5, t^3, -x^3, 0),
Vector(0, 0, t^5 - y^5, -t^2x^3 + y^5, 0),
Vector(ty^9 - y^10, 0, 0, t^5 + t^3y^2 + y^5, -t^3 - ty^2),
Vector(0, 0, t^3y^2 - y^5, t^5 - t^2x^3 - x^3y^2 + y^5, -t^3 + x^3),
Vector(0, 0, t^2y^5 - y^7, t^4x^3 - t^2y^5, -t^2x^3 + y^5),
Vector(0, 0, ty^7 - y^8, -t^5x^3 + t^5y^3 - t^2x^3y^3 + t^3y^5 - x^3y^5 + y^8,
       t^3x^3 - t^3y^3 + x^3y^3 - ty^5)]

//Vector(0, t^2, 1, -1, 0) is homogeneous wrt a suitable grading
//                         Shifts[1,3,5,5,7]
//Vector(ty^9 - y^10, 0, 0, t^5 + t^3y^2 + y^5, -t^3 - ty^2)

// This is homogeneous wrt the suitable grading
S:=Module(
[
Vector(0, 0, t^6 + t^5y + t^4y^2 + t^3y^3 + t^2y^4 + ty^5 + y^6,
       -t^4 - t^3y - t^2y^2 - ty^3 - y^4),
```

```
Vector(t^2, 1, -1, 0),
Vector(y^5, t^3, -x^3, 0),
Vector(0, t^5 - y^5, -t^2x^3 + y^5, 0),
Vector(0, t^3y^2 - y^5, t^5 - t^2x^3 - x^3y^2 + y^5, -t^3 + x^3),
Vector(0, t^2y^5 - y^7, t^4x^3 - t^2y^5, -t^2x^3 + y^5),
Vector(0, ty^7 - y^8, -t^5x^3 + t^5y^3 - t^2x^3y^3 + t^3y^5 - x^3y^5 + y^8,
       t^3x^3 - t^3y^3 + x^3y^3 - ty^5)]
);


ScalarProduct(Gens(I),Tail([0, 0, 0, t^6 + t^5y + t^4y^2 + t^3y^3
                                  + t^2y^4 + ty^5 + y^6,
    -t^4 - t^3y - t^2y^2 - ty^3 - y^4]));
//0

ScalarProduct(Gens(I),Tail([x^6 - ty^5, -x^3, t, 0, 0]));
//x^6 - ty^5

ScalarProduct(Gens(I),Tail([t^2y^5 - y^7, 0, 0, -t^2, 1]));
//t^2y^5 - y^7

SyzOfGens(I);

S=SyzOfGens(I);
True

F1:=SyzOfGens(I);
Len(F1);


F2:=SyzOfGens(F1);
Len(F2);

F3:=SyzOfGens(F2);
Len(F3);


SyzOfGens(I);

S=SyzOfGens(I);
True


Res(R/I);

Len(S);
```

```
Len(SyzOfGens(I));



SyzOfGens(I);


Module([[t^2, 1, -1, 0], [t^2x^3 - y^5, -t^3 + x^3, 0, 0],
[t^7 - t^4x^3 + t^2y^5 - y^7, t^5 - t^2x^3, 0, -t^3 + x^3],
[-t^5x^3 - t^4x^3y - t^6y^2 - t^5y^3 - t^4y^4 - ty^7 - y^8,
-t^3x^3 - t^2x^3y - t^4y^2 - t^3y^3 - t^2y^4 - ty^5 - y^6,
0, tx^3 + x^3y + t^2y^2 + ty^3 + y^4], [t^5x^3 + t^4x^3y +
t^6y^2 + t^5y^3 + t^4y^4 + ty^7 + y^8, t^3x^3 + t^2x^3y +
t^4y^2 + t^3y^3 + t^2y^4 + ty^5 + y^6, 0, -tx^3 - x^3y -
t^2y^2 - ty^3 - y^4], [-t^4x^6 - t^5x^3y^2 - t^4x^3y^3 -
t^6y^4 - t^4y^6 - x^3y^7 - ty^9, -t^2x^6 - t^3x^3y^2 -
t^2x^3y^3 - t^4y^4 - x^3y^5 - t^2y^6 - y^8, 0, x^6 + tx^3y^2
+ x^3y^3 + t^2y^4 + y^6]])
// Third or fourth generator is useless

Use R::=QQ[t,x,y,z],PosTo;

I:=Ideal(t^3-x^3,
         t^2x^3 - y^5,
 t^5-y^5,
 t^7-y^7);

I:=Ideal(MinGens(I));I;
Len(It);

F1:=SyzOfGens(I);
Minimalize(F1);F1;
Len(F1);

F2:=SyzOfGens(F1);
Minimalize(F2);F2;
Len(It);


F3:=SyzOfGens(F2);F3;
Module([[0]]);



Res(R/I);
```

```
0 --> R(-14)
  --> R(-8) + R(-10) + R(-11)
  --> R(-3) + R(-5) + R(-7)
  --> R
```

```
// Computing syzygies and syzygies and syzygies

Use R::=QQ[t,x,y,z],PosTo;

I:=Ideal(xy^2, t^4, xz^3, x^4);
```

```
////////////////////////////////////////////////////
```

```
//Find the minimal gens of three polynomials (F,G,H);
Use R::=QQ[x,y,z];

F:=2x^2y-x+1;
G:=3xy^2-x^2;
H:=2x^3 - 3xy + 3y;
Interreduced([F,G,H]);

3y*F-2x*G=H;
```

```
/*
Example
Use R::=QQ[x,y];
F:=2x^2y-x;
G:=3xy^2-x;
SPoly(F,G);
//2x^2 - 3xy
*/
```

```
Define SPoly(F,G)
  A := LCM(LT(F),LT(G));
  Return LC(G)*(A/LT(F))*F - LC(F)*(A/LT(G))*G
EndDefine;




/*
Use R::=QQ[t,x,y,z];
L:=[t^3-x^3,t^5-y^5,t^7-z^7];
EqSet(Buchberger(L),ReducedGBasis(Ideal(L)));
*/
Define Buchberger(L)
  Pairs:=Flatten([[[L[I],L[J]]|J In (I+1)..Len(L)]|I In 1..Len(L)],1);
  While Pairs <> [] Do
    CurrentPair:=Head(Pairs);
    Pairs:=Tail(Pairs);
    SP := SPoly(CurrentPair[1],CurrentPair[2]);
    SP := NR(SP,L);// Reduce current pair
    If SP <> 0 Then
      Pairs:=Concat(Pairs,[[F,SP]|F In L]);
      Append(L,SP);
    EndIf;
  EndWhile;
  Return Interreduced(Monic(L));// Return the reduced GB
EndDefine;


// Homogeneous Buchberger
// Proceed deg by deg - order the pairs deg by deg
// NR lives the deg  unchanged
// SPoly raise the deg

/*
Example
Use R::=QQ[x,y];
PairLCM([2x^2y-x,3xy^2-x]);
//x^2y^2
*/
Define PairLCM(L)
  Return LCM(LT(L[1]),LT(L[2]));
EndDefine;
/*
Example
Use R::=QQ[x,y,z],Lex;
```

```
F:=x-y^3;
G:=y^2-z^4;
DegPlusOrd(F,G);
//True
F:=x-y^3;
G:=y-z^3;
DegPlusOrd(F,G);
//False
*/
Define DegPlusOrd(F,G)
  If Deg(F)<Deg(G) Then
    Return True;
  EndIf;
  Return LT(F)<LT(G);
EndDefine;

Define PairOrdDeg(P1,P2)
  D1:=Deg(PairLCM(P1));
  D2:=Deg(PairLCM(P2));
  If D1<D2 Then
    Return True;
  ElIf D1>D2 Then
    Return False;
  Else
    Return PairLCM(P1)<PairLCM(P2)
  EndIf;
EndDefine;

Define PairOrdSilly(P1,P2)
  Return PairLCM(P1)>PairLCM(P2)
EndDefine;

/*
Use R::=QQ[t,x,y,z];
L:=[t^3-x^3,t^5-y^5,t^7-z^7];
EqSet(Buchberger(L),ReducedGBasis(Ideal(L)));
*/
Define Buchberger(L)
  Pairs:=Flatten([[[L[I],L[J]]|J In (I+1)..Len(L)]|I In 1..Len(L)],1);
  While Pairs <> [] Do
    Print [Len(Pairs)];
    CurrentPair:=Head(Pairs);
    Pairs:=Tail(Pairs);
    SP := SPoly(CurrentPair[1],CurrentPair[2]);
    SP := NR(SP,L);// Reduce current pair
    If SP <> 0 Then
```

```
      Pairs:=Concat(Pairs,[[F,SP]|F In L]);
      SortBy(Pairs,Function("PairOrdDeg"));// Keep Pairs in deg+ord order
      Append(L,SP);
      Print "*";
    Else
      Print "0";
    EndIf;
  EndWhile;
  Return Interreduced(Monic(L));// Return the reduced GB
EndDefine;
```