The computer algebra program, CoCoA, is available at

http://cocoa.dima.unige.it/

You are encouraged to browse the online help at

http://cocoa.dima.unige.it/download/doc/GUI_help/

including the tutorial in the online manual there. The CoCoA reference card at available at the CoCoA homepage is also recommended.

I will assume that the program is already available on your system or that you have downloaded and installed it according to the instructions at the website. I usually run CoCoA from a shell inside Emacs, but for the purposes of this introduction, I will assume you are using the CoCoA graphical user interface. If you are running a linux system, you can start the program by typing xcocoa at the prompt.

The graphical user interface has two windows: an interactive window in which you type a command and the execution windows in which the results of your commands are displayed.

- To get started, type

\[ 1+1; \]

in the interactive window and enter the result by either hitting Control-Enter or clicking on the appropriate button in the memory bar. **Note that the command, like all CoCoA commands, end with a semicolon.** You should see the output

\[
1+1; \\
2 \\
-------------------
\]

- In the same fashion, enter

\[
A:=3; \\
B:=5; \\
A+B;
\]

You can type all three commands in the interactive window before sending them to CoCoA to be executed. The output should be
A:=3;
B:=5;
A+B;
8

Note that variable names are always capitalized and that assignments are made using :=.

- All calculations are done inside a ring environment. The default environment is $\mathbb{R} := \mathbb{Q}[x,y,z]$. To work in $\mathbb{P}^3$, for instance, you will want to change to a ring with 4 variables like so

Use S::=Q[x,y,z,w];

After entering the above command, you are working in the ring $\mathbb{Q}[x,y,z,w]$. To switch back, you can enter Use R;, and to determine the current ring, use the command CurrentRing(). When declaring a new ring, remember to use two colons, as above.

- Online Help To get help with CoCoA, you can consult the html manual or use the online help system. The two main commands to know for online help are the question mark and the double question mark commands. To find help with a command named Foo, you can type ?foo (case is unimportant). If the word foo exactly matches a CoCoA command, help for that command will be displayed. Otherwise, a list of suggestions is displayed. For example of the use of a single question mark,

?product

============= [ Product ] ============
SUMMARY: the product of the elements of a list.

== SYNTAX ==
Product(L:List):OBJECT

== DESCRIPTION ==
This function returns the product of the objects in the list L.

//==================================== EXAMPLE ===============\
Use R ::= Q[x,y];
Product([3,x,y^2]);
3xy^2
Each help manual entry includes an EXAMPLE section. It is often useful to skip directly to that section. Use the double question mark to find a list of commands related to a given keyword:

??product

See:
  ? CartesianProduct, CartesianProductList
  ? Introduction to Polynomials
  ? Product
  ? ScalarProduct

If your keyword has no exact match, the output of the single and double question mark commands is the same.

• Some basic operations on ideals:

  Use S::=Q[x,y,z,w];
  I:=Ideal(xy-z^2);
  J:=Ideal(xy-zw);
  xy IsIn I;
  FALSE
  x^2y-xz^2 IsIn I;
  TRUE

  I+J; -- sum of ideals (note: a double-dash starts a comment)
  Ideal(xy - z^2, xy - zw)
I*J; -- product of ideals

Ideal(x^2y^2 - xyz^2 - xyzw + z^3w)

Intersection(I,J);
Ideal(x^2y^2 - xyz^2 - xyzw + z^3w)

I^2; -- a product of an ideal with itself
Ideal(x^2y^2 - 2xyz^2 + z^4)

GBasis(I);
[xy - z^2]

Radical(Ideal(x^2));
Ideal(x)

• From the CoCoA tutorial, here is an illustration of the use of lists in CoCoA:

L := [2,3,"a string",[5,7],3,3]; -- L is now a list
L[3]; -- here is the 3rd component of L
a string

L[4]; -- the 4th component of L is a list, itself
[5, 7]

L[4][2]; -- the 2nd component of the 4th component of L
7

L[4,2]; -- same as above
7

Append(L,"new");
L;
[2, 3, "a string", [5, 7], 3, 3, "new"]

-- insert 8 as the 4th component of L, shifting the other
-- entries to the right:
Insert(L,4,8);
L;
[2, 3, "a string", 8, [5, 7], 3, 3, "new"]
Remove(L,4); -- remove it again
L;
[2, 3, "a string", [5, 7], 3, 3, "new"]

Len(L); -- the number of components of L
7

MakeSet(L); -- same as L but with repeats removed
[2, 3, "a string", [5, 7], "new"]

1..5; -- a range of values
[1, 2, 3, 4, 5]

[ X^2 | X In 1..5]; -- a useful way to make lists
[1, 4, 9, 16, 25]

[1,2] >< [3,4] >< [5]; -- Cartesian product: use a greater-than
-- sign ">" and a less-than sign "<" to make
-- the operator "><".
[[1, 3, 5], [1, 4, 5], [2, 3, 5], [2, 4, 5]]

• We create the ideal of the twisted cubic in \( \mathbb{P}^3 \) by first eliminating \( t \) from the equations

\[
x = t, \quad y = t^2, \quad z = t^3
\]

then homogenizing the ideal with respect to the variable, \( w \).

Use S::=Q[t,x,y,z,w];
I:=Ideal(x-t,y-t^2,z-t^3);
J:=Elim(t,I);
J;
J:=Homogenized(w,J);
J;

The output is

Ideal(-x^2 + y, -xy + z, -y^2 + xz)

Ideal(-y^2 + xz, -xy + zw, -x^2 + yw)
We then calculate the Hilbert function (I will show the input and output together):

\[
\text{Hilbert}(S/J);
\]
\[
H(t) = \frac{3}{2}t^2 + \frac{5}{2}t + 1 \quad \text{for } t \geq 0
\]

and the free resolution

\[
F:=\text{Res}(S/J);
F;
\]

\[
0 \rightarrow S^2(-3) \rightarrow S^3(-2) \rightarrow S
\]

Describe(F);

\[
\begin{bmatrix}
-x^2 + yw, & -xy + zw, & -y^2 + xz
\end{bmatrix}
\]

\[
\begin{bmatrix}
z, & -y, \\
-y, & x, \\
x, & -w
\end{bmatrix}
\]

• Here we compute the ideal of two points, \((1, 0, 1)\) and \((2, 3, 1)\) in \(\mathbb{P}^2\), the first point counted with multiplicity two (by squaring its ideal):

\[
\text{Use } S:=\mathbb{Q}[x,y,z];
I:=\text{Ideal}(x-z,y);
J:=\text{Ideal}(x-2z,y-3z);
K:=\text{Intersection}(I^2,J);
K;
\]

\[
\text{Ideal}(xy - 1/3y^2 - yz, x^2 - 1/9y^2 - 2xz + z^2, y^3 - 3y^2z)
\]

And compute its Hilbert function and free resolution (note the use of the \(\text{It}\) variable):
Hilbert(S/K);
H(0) = 1
H(1) = 3
H(t) = 4 for t >= 2

Res(S/K);
0 --> S(-3)(+)S(-4) --> S^2(-2)(+)S(-3) --> S

Describe(It);
Mat([xy - 1/3y^2 - yz, x^2 - 1/9y^2 - 2xz + z^2, y^3 - 3y^2z])
Mat([x + 1/3y - z, y^2 - 3yz],
[-y, 0],
[0, -x + 1/3y + z])

To compute the ideal for a set of points in projective space (without multiplicities), use the function IdealOfProjectivePoints:

Use R ::= Q[x,y,z];
I := IdealOfProjectivePoints([[0,0,1],[1/2,1,1],[0,1,0]]);
I;
Ideal(xz - 1/2yz, xy - 1/2yz, x^2 - 1/4yz, y^2z - yz^2)

I.Gens; -- the reduced Groebner basis
[xz - 1/2yz, xy - 1/2yz, x^2 - 1/4yz, y^2z - yz^2]

Note that I.Gens gives a list of the generators of the ideal I.