Intro to Macaulay2

Macaulay2 is a software system devoted to supporting research in algebraic geometry and commutative algebra, developed by Daniel R. Grayson and Michael E. Stillman with funding from the National Science Foundation.

Installation. Instructions and documentation can be found at

www2.macaulay2.com/Macaulay2/.

Basic operations. Try the following input to see what happens.

```
2+2
1*2*3*4
2^200
40!
1;2;3*4
4*5;
4/2
4 // 2
4 % 2
4 % 3
4 // 3
oo
o5+1
```

We can also make functions in Macaulay2; try the following:

```
f = i -> i^3
f 5
g = (x,y) -> x*y
g(6,9)
```

Rings. To work in a polynomial ring, we must first define it.

```
S = ZZ/5[x,y,z]
(x+y)^5
```

What is `ZZ/5`? How do you make the coefficient ring the rational numbers? What does the following input do?

```
1_S
0_S
numgens S
gens S
vars S
coefficientRing S
random(3, S)
basis(2, S)
```

Every polynomial ring in Macaulay2 is equipped with a monomial order.

```
S = ZZ/101[a,b,c]
(a+b+c+1)^3
```

Explicit comparison of monomials with respect to the chosen ordering is possible.

```
b^2 > a*c
```

The comparison operator `?` returns a symbol indicating the result of the comparison: the convention is that the larger monomials are printed first (leftmost).

```
b^2 ? a*c
```

The monomial ordering is also used when sorting lists with sort.

```
sort {1_S, a, a^2, b, b^2, a*b, a^3, b^3}
```

Describe the default monomial ordering used in Macaulay2. The next ring uses `MonomialOrder` to specify graded lexicographic ordering.

```
S = ZZ/101[a,b,c, MonomialOrder => GLex];
(a+b+c+1)^3
```

The next ring uses lexicographic ordering.

```
S = ZZ/101[a,b,c, MonomialOrder => Lex];
(a+b+c+1)^3
```

How would you describe the following monomial orders?

```
S = ZZ/101[a,b,c, MonomialOrder => Eliminate 2];
(a+b+c+1)^3
S = ZZ/101[a,b,c, MonomialOrder => ProductOrder{1,2}];
(a+b+c+1)^3
S = ZZ/101[a,b,c, Degrees => {1,2,3}];
(a+b+c+1)^3
```

Gröbner basics. The division algorithm discussed in class can be implemented in Macaulay2 as follows:

```
division = (f,G) -> (
  S := ring f;
  p := f;
  r := 0_S;
  m := #G;
  Q := new MutableHashTable;
  for j from 0 to m-1 do Q#j = 0_S;
  while p != 0 do (
    i := 0;
    while i < m and leadTerm(p) % leadTerm(G#i) != 0 do i = i+1;
    if i < m then (
      Q#i = Q#i + (leadTerm(p) // leadTerm(G#i));
      p = p - (leadTerm(p) // leadTerm(G#i)*G#i))
    else (
      r = r + leadTerm(p);
      p = p - leadTerm(p)));
  L := apply(m, j -> Q#j);
  return (r,L));
```

What does the following input do?

```
f = x^2*y
G1 = {x*y-x, x^2-y}
G2 = {x^2-y, x*y-x}
division(f,G1)
f % matrix{G1},f // matrix{G1}
division(f,G2)
f % matrix{G2},f // matrix{G2}
gens gb ideal G1
```

The following example indicates how the monomial order can affect the length of a Gröbner basis computation and the complexity of the answer.

```
S = QQ[x,y,z];
I = ideal(x^5+y^4+z^3-1, x^3+y^2+z^2-1);
time gens gb I
S' = QQ[x,y,z, MonomialOrder => Lex];
I' = substitute(I,S')
time gens gb I'
```

Algebraic subsets. Which affine varieties do the following ideals I define?

```
S = QQ[x,y,z];
I = ideal(x*y, x*z)
decompose(I)
clearAll
n=5
S = QQ[x_1..x_n];
M = matrix table(5,5, (i,j) -> S_i^j) factor det(M)
S = QQ[a..i];
M = genericMatrix(S,a,3,3)
I = ideal det M
I = minors(2,M)
transpose mingens I
S = QQ[t,a..i];
M = genericMatrix(S,a,3,3)
Mt = t*id_(S^3)-M
I = ideal substitute(
     contract(matrix{{t^2, t, 1}}, det(Mt)),
     {t => 0_S})
transpose gens I
```

This content of this worksheet was written by Gregory G. Smith.