MACAULAY2 - INTRODUCTION

AUGUST 23RD, 2023

If you haven't used Macaulay2 before, the easiest way to do it is to open:

`http://www.unimelb-macaulay2.cloud.edu.au/`

However, let's learn how to use Macaulay2 on departmental machines.

1. Instructions for using departmental machines locally

Open up the machine in front of you, logging in with your mathematics department username and password, and open a terminal application. Run the command:

```
M2
```

You should see something like:

```
Macaulay2, version 1.21
with packages: ConwayPolynomials, Elimination, IntegralClosure,
     InverseSystems, Isomorphism, LLLBases, MinimalPrimes, OnlineLookup,
     PrimaryDecomposition, ReesAlgebra, Saturation, TangentCone
i1 :
```

In the cursor execute the command `setup()`, it may ask you questions. Say yes to all. Then execute `exit` to leave the terminal version of Macaulay2.

Now run

```
emacs-28.1 &
```

This will start emacs, which is a nicer environment for running Macaulay2 compared to the terminal. After emacs starts, you can hit `F12` and run Macaulay2 in emacs. You may want to follow that up with `ctrl-x` followed by the key `1`. Note, you can sometimes also run `emacs &` depending on your machine configuration (where you logged in, the `emacs &` may not work). Note the symbol `&` means don't lock you out of the terminal.

2. Running Macaulay2 on departmental machines remote from your laptop/desktop

Alternately, you can also log in to departmental machines and run Macaulay2 there. Roughly the idea is to open a terminal application on your local machine, and run the following command:

```
ssh -X mathid@asimov.math.utah.edu
```

Here `mathid` is your mathematics department username. Then login with your mathematics department password.

Then follow the instructions above `M2`, `setup()`, `exit`, `emacs-28.1 &`. If you have a fast enough internet connection (for instance if you are on campus), that should be ok. However, you will need X11 on your system.

For **Mac**, you probably need XQuartz.

For **Windows**, you will need to use WSL2 and will need to follow the instructions here.

`https://learn.microsoft.com/en-us/windows/wsl/tutorials/gui-apps`

Or you can run Macaulay2 in the terminal (ie, run `M2` instead of emacs).

3. Running Macaulay2 locally on your machine

You'll need to follow the instructions here.

`https://www.macaulay2.com/Downloads/`

For Windows, you'll want to use WSL2 (ie, the Ubuntu app for windows), and then basically install it like you would in linux (ie, Ubuntu). For Mac and Linux operating systems, the procedure tends to be pretty straightforward. In many cases, Macaulay2 is available in the default repositories.

I plan to devote the second half of next Monday's seminar to helping people with this if they would like. In particular, we can also work on compiling Macaulay2 from source.

4. Simple mathematical Macaulay2 exercises

In emacs or the terminal, or the web interface, start up Macaulay2. You should see something like the following.

```
Macaulay2, version 1.21
with packages: ConwayPolynomials, Elimination, IntegralClosure,
     InverseSystems, Isomorphism, LLLBases, MinimalPrimes, OnlineLookup,
     PrimaryDecomposition, ReesAlgebra, Saturation, TangentCone
i1 :
```

After the `i1 :` type

```
p = 5 <enter>
S = ZZ/p[x,y,z] <enter>
I = ideal(x^2-y*z) <enter>
R = S/I <enter>
dim R <enter>
dim S <enter>
```

We have just created a quotient ring and computed its dimension. Let's understand ring maps. The syntax to create a ring map is:

```
phi = map(target, source, {a,b,c,...})
```

Here `target` is the target ring of the map, `source` is the source ring of the map, and `a,b,c,...` is a list of elements of the target where the *variables* of the source go. For example, run the following commands (in the same window you ran the above)

```
A = ZZ/p[a,b] <enter>
phi = map(A, S, {a^2, a*b, b^2}) <enter>
J = ker phi <enter>
```

You can also get help on any topic by running `help map` or to get help on specific syntax, `help (map, Ring, Ring, List)` or in a nicer format: `viewHelp map`, respectively `viewHelp (map, Ring, Ring, List)`.

1. Create a Frobenius map (called `F`) from the ring `R` to itself. Verify that the map is injective by checking that kernel is zero.

2. Now, we want to view the target of a map as a module over the source. To do that, we'll use the package `PushForward`. Run the following commands to learn how this works. We'll study the inclusion $R = \mathbb{Z}/5[ab, a^2, b^2] \hookrightarrow \mathbb{Z}/5[a, b] = A$.

```
loadPackage "PushForward" <enter>
psi = map(A, S, {a*b, a^2, b^2}) <enter>
isWellDefined psi <enter>
M = (pushFwd(psi))#0 <enter>
rank M <enter>
isHomogeneous M <enter>
pdim M <enter>
```

Here we viewed A as an R-module and computed its generic rank (over R) and its projective dimension[1] In particular, since the projective dimension is not zero, `M` is not free.

3. Do the same thing, but create a different Veronese subring.

4. Use the Macaulay2 help system to out to find a primary decomposition of an ideal. In particular, find a primary decomposition of this ideal:

```
T = QQ[x,y,z,w]
J = ideal(y^2*w,x*y^2+y^2*z,z^2*w^2,y*z*w^2,x*z*w^2,x*y*w^2,x^2*w^2)
```

Verify that your primary decomposition is correct using the `intersect` command. Note you can check if two ideals `I, J` are equal by inputting `I == J`.

5. Use Macaulay2 to take a free resolution of the `T`-module `T^1/J` which you constructed in the previous problem (use the documentation to figure out how to take a free resolution). Compute $\mathrm{Tor}_1^T(T/J, T/J)$ in two ways, one by manually using the `**` command (which means tensor), by working with the individual maps in the (resolution tensored with `T^1/J`). Also use

[1] note projective dimension `pdim` only works well if the module is homogeneous, which it is in this case, it also can't verify if something has projective dimension infinity, but we just care if the module is projective.

Macaulay2's built in `Tor` command. Verify that this module is nonzero by using the `dim` command (try looking it up). Did they give you the same answer?