gfanInterface2

To open Macaulay 2 you need to type the following in the terminal on the linux computers in maths, /opt/Macaulay2-32bit/bin/M2

To install gfan on the maths linux computers, you need to go to the Anders’ gfan manual and follow the red instructions only. Also, when I installed gfan I was only able to get to the commands by typing the following:

cd gfan/bin

(then it’s in gfan, then for each command I had to type ./ before the name of the command) e.g.

./gfan\_buchberger

Some very basic but important Macaulay command:

restart

exit

clearAll

viewHelp

installPackage “gfanInterface2”

installPackage(“gfanInterface2”, RemakeAllDocumentation =>true)

QQ[x,y,z]

I=ideal(f,g,h) gives the I=<f,g,h>

L={f,g,h} is a list

gens gb(I) or gens gb(L) gives you the generators of the Groebner basis of I and L respectively

flatten entries gens I = L for I and L as above, this gets you from an ideal to a list

ideal L = I for I and L as above this takes you from a list to an ideal

You can use "MPLConverter(I)" to make a list or ideal into a MarkedPolynomialList and use "last (MPL)" to get a list from a MPL (markedPolynomialList)

A MarkedPolynomialList (MPL) is a paired list where the first list is the list of the initial/leading terms of each polynomial of the second list (in order) and the second list is just a list of polynomials

for example: if we have L as our list, "markedPolynomialList{{in(f), in(g), in(h)}, {f,g,h}}"

for "L={x^2+y, y^2+z, z^2+x}" using revlex, we would input the following "markedPolynomialList{{x^2, y^2, z^2}, {x^2+y, y^2+z, z^2+x}}"

You can combine Macaulay 2 (M2) commands with gfanInterface2 commands how every you want, use M2 inside a gfanInterface2 command, use a M2 command on a gfanInterface2 output

for example: if you have "L={x^2+y, x\*y -z}" and "K={z-y^3, z^3}", and you want the reduced Groebner Basis of both L union K, "gfanBuchberger(L|K)"

for example: if you apply "last" to a MarkedPolynomialList it will give you the Groebner basis as a list, "Last gfanBuchberger({x^2+y, y^2-z})"

if you use an option then you just need to put a comma after your input and then write the option e.g. installPackage("gfanInterface2", RemakeAllDocumentation => true)

“viewHelp String”, will open the command's webpage (use to find out the right form of your input, such as ideal/list/MPL/Fan/Cone etc) e.g. “viewHelp gfanGroebnerCone”

If something doesn’t make sense on the gfan package with the documentation or the name, then look at Anders' gfan manual, the maths is the same.

oo gives you the previous line, so answer.

H#" " gives you the information of what's inside the string for H=hashTable, e.g. H#“Rays” will give the Rays output.

If you get an error message like the one below

gfanInterface2.m2:2912:1:(3):[10]: error: documentation key "(gfanLeadingterms,MarkedPolynomialList)" encountered, but "gfanLeadingterms" is not a function, command, scripted functor, or keyword

the 2912 gives you the line you should go to on the package look on the left, the (3) sometimes means the 3rd line within the documentation (if it’s in a documentation), and [10] indicates the 10th character of the line.

In codes you have to define every character you use, so if you use x for the first time and you want it to be a symbol you need to specify

e.g. x:= symbol x;

Capitals matter, gfaninterface2 is nothing to M2, you have to write gfan**I**nterface2

You can copy and paste in the terminal using Ctrl+Shift+c or Ctrl+Shift+v

Use ".." for example QQ[x\_1..x\_20] is QQ[x\_1 to x\_20]

Sometimes if using round brackets ( ) doesn’t work, you can try using curly brackets { } (but make sure you get what you expected).

For help about Macaulay 2, to find a command/function try going to Macaulay 2 page, then click development (on the left bar under documentation), if you can’t find it there go to The Macaulay2 language.

What I would do had I had another week?

Firstly, I would look at how to make the HashTable into a Fan for gfanTropicalTraverse. So the outpus of gfanTropicalTraverse is a HashTable but we want the output to be a Fan. If we find any method of changing anything into a Fan, we can always change the HashTable into a list for example by just removing the command “hashTable” from the code. So any method would do.

For the indexing problem, for instance in QQ[x\_1..x\_20] gfan confuses x\_1 with x\_11..x\_19 because of the one. So we need to change that ring to QQ[x\_10..x\_19] add 9 to the index. I’ve created a code to change the ring

--------------------------------------------------------

-- gfanRingToNewRing

--------------------------------------------------------

gfanRingToNewRing = method(Options=>{})

gfanRingToNewRing Ring := opts -> (R) -> (

input := Ring ;

x := symbol x ;

newRing(R, Variables=>{x\_10..x\_(numgens(R)+9)})

)

Then I think that we need to change the input in the form of the new ring. Maybe try this if the input is a list.

Map list to new ring

R := ring first L;

x := symbol x;

S := newRing(R, Variables=>{x\_10..x\_(numgens(R)+9)});

f := map(S,R,{x\_10..x\_(numgens(R)+9)});

M := f(L)

Then apply the gfancommand on this, so look at existing code to see how they do that, then change that answer into a M2 output, now the outputs should be in the new variables, so use an inverse map to change the output into the original variables.

Once this is done, we need some if else statements to make sure we do this only ring that have indexed variables, the gfanRingToNewRing code maps all rings to this new ring.

For the rays and lineality space problem, we might need to change the rays and lineality space’s generator into a matrix by simply using

matrix (H#”Rays”|H#”LinealitySpace) where H is the HashTable so

H=gfanTropicalTraverse

then you need to find a way of row-reducing the matrix

The aim is to get a zero in one of the positions of the variables. And then extracting the row/column of the matrix as vectors and call then the new Rays.

I tried things with making the matrix into polynomial and then finding the Groebner basis, but I don’t know whether it’s the right thing to do.

Once this has been done I would see if I can specify the variable I want to zero, maybe by changing the order of the polynomial if it’s done using Groebner Basis and changing the matrix from x y z to like y x z so that row reduction keeps y and it’ll get rid of z and x.

Finally, I would just look at some of the documentation typos and mistakes, and write some documentation for some of the functions that don’t have it like gfanVersion.