Oscar: state, plans and dreams

September 22, 2020

Develop a visionary, next generation, open source computer algebra system, integrating all systems, libraries and packages developed within the TRR.

Interfaces

At the end of the first period we promised to have interfaces in place for all the cornerstones:

- `GAP.jl` a bi-directional interface to GAP
- `Singular.jl` low-level interface to Singular
- `Polymake.jl` low-level interface to polymake
- `Hecke.jl` (ANTIC) is written in Julia anyway

there is more, but those are the most important. (Many people are involved)

And now there is:

Oscar

```
julia> Pkg.add("Oscar")
... [wait some time] ...
julia> using Oscar
 -----    -----    -----      -      -----
|     |  |     |  |     |    | |    |     |
|     |  |        |         |   |   |     |
|     |   -----   |        |     |  |-----
|     |        |  |        |-----|  |   |
|     |  |     |  |     |  |     |  |    |
 -----    -----    -----   -     -  -     -

...combining (and extending) ANTIC, GAP, Polymake and Singular
Version 0.4.0 ...
 ... which comes with absolutely no warranty whatsoever
Type: ’?Oscar’ for more information
(c) 2019-2020 by The Oscar Development Team
```

Interfaces: Status

- `GAP.jl`: complete, all functionality of all packages is available, GAP can call any Julia function and vice versa
- `Polymake.jl`: complete, all functions can be used
- `Singular.jl`: the core functionality is available, some kernel and library functions lack wrappers.

For GAP and polymake the wrappers are be automated, for Singular this is more difficult - the "meta-data" (types) is missing.

Interfaces: GAP

This means, one can use GAP commands in Oscar, but they look like GAP:

```
julia> GAP.Globals.SymmetricGroup(6)
GAP: Sym( [ 1 .. 6 ] )
julia> GAP.Globals.MaximalSubgroupClassReps(ans)
GAP: [ Alt( [ 1 .. 6 ] ), Group([ (1,2,3,4), (1,2), (5,6) ]), Gr
  Group([ (1,2), (3,4), (5,6), (1,3,5)(2,4,6), (1,3)(2,4) ]), Gr
       ]), PGL(2,5) ]

julia> typeof(ans)
Main.ForeignGAP.MPtr
```

The same is true for polymake and Singular. In fact, there are a number of notebooks (Jupyter) showcasing this.

Oscar-ification

The next step is (and we're working on it) to "Oscar-ify" the GAP, Singular and polymake objects. In order for a CA system to be usable for non-specialists, the notation (commands) and behaviour need to follow as closely as possible a standard text-book and to be consistent as far as possible.

On the other hand, the "specialist" might want to have access to the implementation details and specific algorithms...

Also: different implementation languages (Singular (C++, Singular), GAP (C, GAP), polymake (C++, perl)) force different presentations and choices, so for Julia they will have to change again....

Galois Groups

With Max Horn, we started to implement Galois groups of (rational) polynomials - to drive the development of the group interface in Oscar. Using Stauduhar's approach as in Magma:

- find a suitable field and compute roots
- find subfields - and thus the largest possible group
- for maximal subgroups
  - find a separating invariant (a multivariate polynomial)
  - evaluate at the roots
  - if this is an integer, replace the group by the subgroup

Galois Group: example

```
julia> K, a = number_field(x^4-2)
julia> set_verbose_level(:GaloisGroup, 2)
julia> galois_group(K)
using prime 1048583 with degree 2
and cycle types Set([[2, 2], [4], [1, 1, 1, 1], [1, 1, 2]])
computing subfields ...
group will have (maximal) block systems: [[1, 2]]
found Frobenius element: [1, 2, 4, 3]
Have starting group with id 3
found 1 many maximal subgroups
testing descent 3 -> 1 of index 2
...
(Group([ (1,2), (3,4), (1,3)(2,4) ]),
    Galois Context for x^4 - 2 and prime 1048583)
```

Galois Group: example cntd.

```
julia> maximal_subgroups(ans[1])
3-element Array{PermGroup,1}:
 Group([ (1,3)(2,4), (1,2)(3,4) ])
 Group([ (1,2), (1,2)(3,4) ])
 Group([ (1,3,2,4), (1,2)(3,4) ])
julia> typeof(ans[1])
PermGroup
```

Multivariates

Daniel Schulz implemented new multivariate factoring algorithms in Nemo (finite fields, rationals) and AbstractAlgebra (generic infrastructure). Tommy Hofmann and Daniel extended this to number fields, leveraging the infrastructure in Hecke.

Similar, the infrastructure was used to implement new methods for

- multivariate gcd over number fields
- Groebner basis over number fields
- absolute factorisation over the rationals (WIP)
- roots and factoring over number fields (WIP)

Commutative Algebra

Based on Nemo's multivariate arithmetic and Singular's Groebner engine we have (sparse) (multi-)graded modules over all exact rings and fields. Groebner bases over number fields automatically are re-routed to the modular method.

Extending the modular methods to the rationals, to modules and to function fields is WIP. We have

- free modules
- sub-, quotient-, and SubQuotient modules
- hom-module, tensor product
- homogenous components
- homomorphisms, images, kernel
- free resolution, complexes

Other

- GITFan (Thomas Breuer, Janko Böhm)
- GroupAtlas (Thomas)
- SingerAlg (Thomas)
- (general) projective space (WIP)
- irreducibility and factoring of number field elements
- perfect power detection of integers
- StraightLinePrograms (Rafael Fourquet)
- lots of group theory (Max, Giovanni de Franceschi)
- Reynolds operator a la Souvignier
- ...

Plans

The interfaces to the systems will be widened, more functionality will be Oscar-ified. Eg:

- Oscar-ification of GITFan (Janko and Thomas)
- OscarPolytopes to provide Oscar-style access to polymake.
- Group recognition to drive the group interface
- Integrate code developed by Jean Michel and Meinolf Geck.
- Variety-driven interface to commutative algebra (Delphine Pol)
- Exterior algebra, non-commutative algebra
- GBLA (Christian Eder)
- polymake over number fields

The Book

- The Oscar book

Dreams

- Uli Thiel's package...
- Tropical geometry
- Toric varieties
- More users and developers
- GPIspace in/ for Julia
- RandomMatrix stuff

More Information

- https://oscar.computeralgebra.de/
- https://github.com/oscar-system/Oscar.jl/
- http://nemocas.org/
- https://github.com/thofma/Hecke.jl