Number Theory in Oscar: From Class Groups to Class Fields and Beyond

Claus Fieker

June 28, 2023

Preliminaries

- Mostly in Hecke `https://github.com/thofma/Hecke.jl`
- $> 200k$ loc, in Julia
- main maintainer: Tommy Hofmann
- many contributers
- much broader scope than presented

Why Julia?

- Interactive
- As fast as c
- Solves 2-language problem
- Not maintained by us
- Modern
- Interoperates well with c

Basics

Definition

A number field is a finite extension of $\mathbb{Q}$

- no link to $\mathbb{C}$
- based in vector space structure
- inefficient in general

Primitive Element

Theorem

For a number field K, there is some $\alpha \in K$ s.th. $K = \mathbb{Q}[\alpha]$. Furthermore, for the minimal polynomial $f \in \mathbb{Q}[t]$ of α we have $K \equiv \mathbb{Q}[t]/f$.

- Sonn-Zassenhaus: given a basis, some α can be found
- mostly, we start with f, typically monic, integral, irreducible
- interesting fields are often differently given

Example

```
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _‘ |  |
  | | |_| | | | (_| |  |  Version 1.8.5 (2023-01-08)
 _/ |\__’_|_|_|\__’_|  |  Official https://julialang.org/ release
|__/                   |

julia>
```

Example

```
julia> using Oscar
 -----    -----    -----      -      -----
|     |  |     |  |     |    | |    |     |
|     |  |        |         |   |   |     |
|     |   -----   |        |     |  |-----
|     |        |  |        |-----|  |   |
|     |  |     |  |     |  |     |  |    |
 -----    -----    -----   -     -  -     -

...combining (and extending) ANTIC, GAP, Polymake and Singular
Version 0.12.2-DEV ...
 ... which comes with absolutely no warranty whatsoever
Type: '?Oscar' for more information
```

Example

```
julia> Qx, x = QQ["x"];
julia> K, a = number_field(x^3-2)
(Number field of degree 3 over QQ, _a)
julia> a^3
2
julia> basis(K)
3-element Vector{nf_elem}:
 1
 _a
 _a^2
```

More Constructions

Number Fields can be constructed in many other ways:

- special constructs (qudratic fields, cyclotomics)
- splitting fields, normal closures
- subfields
- extensions of number fields (by polynomials)
- *non-simple* fields
- composita

Examples

```
julia> Ky, y = K["y"];
julia> L, b = number_field(y^3-a)
(Relative number field of degree 3 over number field, _$)
julia> b^3
_a
julia> b^9
2
julia> k, c = number_field([x^2-2, x^2-3, x^2-5])
(Non-simple number field of degree 8 over QQ, NfAbsNSElem[_$1, _$2, _$3])
julia> c[2]^2
3
```

Primitive Element - 2

The existence theorem is effective:

```
julia> absolute_simple_field(k)
julia> ka, mp = absolute_simple_field(k);

julia> ka
Number field with defining polynomial x^8 - 40*x^6 + 352*x^4 - 960*x^2 + 5
  over rational field
julia> mp(gen(ka))
_$1 + _$2 + _$3
julia> preimage(mp, c[2])
-1//96*_a^7 + 37//96*_a^5 - 61//24*_a^3 + 15//4*_a
```

Basics

For arithmetic purposes, one wants to work with rings, in particular the ring of integers, maximal order:

Definition

$\alpha \in K$ is called integral (algebraic integer) iff the minimal polynomial is monic and integral. The set of all algebraic integers is called the integral closure of $\mathbb{Z}$ in K or maximal order $\mathbb{Z}_K$.

Actually:

- $\mathbb{Z}_K$ is a ring
- $\mathbb{Z}_K$ is a free $\mathbb{Z}$-module of rank $n = K : \mathbb{Q}$
- $\mathbb{Z}_K$ can be computed...

Examples

```
julia> ZK = maximal_order(K)
Maximal order of Number field of degree 3 over QQ
with basis nf_elem[1, _a, _a^2]

julia> maximal_order(quadratic_field(5)[1])
Maximal order of Real quadratic field defined by x^2 - 5
with basis nf_elem[1, 1//2*sqrt(5) + 1//2]
```

Class Group

- The maximal order is a Dedekind domain, hence the ideals form a group. The quotient by principal ideals is finite, the *class group*.
- The class group is trivial iff the maximal order is a PID.
- The class group controls the multiplicative structure of a number field.
- The implementation in Oscar is competitive and was involved in world record computations.

Examples

```
julia> k, a = quadratic_field(-1009);
julia> c, mc = class_group(k)
(GrpAb: Z/20, ClassGroup map of
Set of ideals of O_k)

julia> mc(c[1])
<11, 117*sqrt(-1009) + 75>
Norm: 11
Minimum: 11
two normal wrt: 11
```

Examples

```
julia> ans^10
<25937424601, 317813919752875905470287228244000*sqrt(-1009) - 49351124827
Norm: 25937424601
Minimum: 25937424601
two normal wrt: 11

julia> I = mc(10*c[1])
<2, 3*sqrt(-1009) + 3>
Norm: 2
Minimum: 2
two normal wrt: 2
```

Examples

```
julia> is_principal(I)
(false, 1)

julia> is_principal(I^2)
(true, -2)
```

Examples

```
julia> factor(3*5*7*order(I))
Dict{NfOrdIdl, Int64} with 4 entries:
  <7, -1008>              => 1
  <3, -1005>              => 1
  <5, sqrt(-1009) + 9> => 1
  <5, sqrt(-1009) + 1> => 1

julia> [preimage(mc, x) for x = keys(ans)]
4-element Vector{GrpAbFinGenElem}:
 Element of c with components [0]
 Element of c with components [0]
 Element of c with components [4]
 Element of c with components [16]
```

Basics

- A *Class Field* is an extension with abelian automorphism group
- Class Fields are parametrized by generalized class groups
- Many properties can be read off the parameters, other not (yet)
- Links to analytic theory
- "Easiest" example: the Hilbert Class Field...
  - Maximal abelian unramified extension
  - Automorphism group isomorphic to class group
  - Every ideal of base becomes principal

Examples

```
julia> k, a = wildanger_field(3, 13);
julia> k
Number field with defining polynomial x^3 - 13*x^2 + 13*x - 13
  over rational field

julia> H = hilbert_class_field(k)
Class field defined mod (<1, 1>, InfPlc{AnticNumberField, Hecke.NumFieldEm
   of structure Abelian group with structure: Z/9
```

Examples

```
julia> galois_group(H)
Group([ (1,2,3,4,5,6,7,8,9) ])

julia> discriminant(H)
<1, 1>
Norm: 1
Minimum: 1
principal generator 1
two normal wrt: 1
```

Examples

```
julia> K = number_field(H)
Non-simple number field with defining polynomials
[x^9 + (-54*_$^2 + 648*_$ - 567)*x^7 + (27*_$^2 - 216*_$ - 3780)*x^6 + (-2
  over number field with defining polynomial x^3 - 13*x^2 + 13*x - 13
    over rational field

julia> Ka, mp = absolute_simple_field(K);
julia> class_group(Ka)
(GrpAb: Z/1, ClassGroup map of
Set of ideals of O_Ka)
```

Field Factory

Theorem (Shafarevich)

Every solvable group occurs as a Galois group over $\mathbb{Q}$.

- Every solvable group is a chain of abelian (cyclic) groups
- Every solvable field is a tower of class fields
- Most towers yield the wrong group
- Sircana automated this completely

Examples

```
julia> small_group(8, 3)
<pc group of size 8 with 3 generators>

julia> describe(ans)
"D8"

julia> fields(8, 3, ZZ(10)^7)
7-element Vector{Hecke.FieldsTower}:
 Field context for the number field defined by x^8 + 6*x^4 + 1
 Field context for the number field defined by x^8 - 2*x^7 + 7*x^6 - 14*x^
 ...
```

Galois Groups

Carefully distinguish:

Galois Group vs. **Automorphism Group**

Fact (Automorphism Group)

The group of actual endomorphisms of the field

Fact (Galois Group)

For roots $\alpha_1, \ldots, \alpha_n$, the group of legal permutations of those roots.

Examples

```
julia> k, a = cyclotomiic_field(5);
julia> automorphism_group(PermGroup, k)
(Group([ (1,2,3,4) ]), Composite map consisting of the following

Group([ (1,2,3,4) ]) -> Generic group of order 4 with multiplication table
then
Generic group of order 4 with multiplication table -> Set of automorphisms
)

julia> galois_group(k)
(Group([ (1,4,2,3), (1,2)(3,4) ]), Galois Context for x^4 + x^3 + x^2 + x -
```

Examples

```
julia> k, a = wildanger_field(3, 13);
julia> automorphism_group(PermGroup, k)
(Group(()), Composite map consisting of the following

Group(()) -> Generic group of order 1 with multiplication table
then
Generic group of order 1 with multiplication table -> Set of automorphisms
)

julia> galois_group(k)
(Sym( [ 1 .. 3 ] ), Galois Context for x^3 - 13*x^2 + 13*x - 13 and prime
```

Examples

```
julia> g, s = galois_group(k);
julia> roots(s, 5)
3-element Vector{qadic}:
 4*7^0 + 6*7^1 + 2*7^2 + O(7^3)
 (5*7^0 + 4*7^2 + O(7^3))*a + 2*7^0 + 3*7^1 + O(7^3)
 (2*7^0 + 6*7^1 + 2*7^2 + O(7^3))*a + 6*7^1 + 3*7^2 + O(7^3)
```

Examples

```
julia> n = normal_subgroups(g)
3-element Vector{PermGroup}:
 Sym( [ 1 .. 3 ] )
 Alt( [ 1 .. 3 ] )
 Group(())

julia> fixed_field(s, n[2])
Number field with defining polynomial x^2 + 59488
  over rational field

julia> fixed_field(s, n[3])
Number field with defining polynomial x^6 - 26*x^5 + 178659*x^4 - 3093740*
  over rational field
```

Solvability

Classically solvabilty is linked to explicit ways to write the roots of a polynomial in terms of (nested) radicals.
This is possible iff the associated Galois group is solvable - and the usual proof is constructive...

Examples

```
julia> solve(x^3+x+1)
(Relative number field of degree 3 over relative number field, Any[((1//9*

julia> ans[1]
Relative number field ... x^3 + (3*z_3 + 3//2)*a2 + 27//2
  over relative number field ... x^2 + 31
    over number field ... x^2 + x + 1
      over rational field
```

Examples

```
julia> h = Hecke.fields(8, 3, ZZ(10)^7);
julia> solve(h[1])
(Relative number field of degree 2 over relative number field, Any[a3 + 1/

julia> ans[1]
Relative number field ... x^2 + 1//2
  over relative number field ... x^2 + a1 - 2
    over number field ... x^2 + 4
      over rational field
```

Representations

Building on top of the Gap character theory and the class field theory we can do interesting computations...

Let $G = 24T201$ of order 240. The 9-th character if G has Schur index 2, character field $\mathbb{Q}$. The representation, in Gap, is constructed over $\mathbb{Q}(\zeta_5)$.

The representation should live over a quadratic field, however, not over $\mathbb{Q}[\sqrt{5}]$.

Example

```
julia> G = transitive_group(24, 201);
julia> order(G)
240
julia> T = character_table(G)[9];
julia> R = gmodule(T)
G-module for t24n201 acting on Vector space of dimension 6
over abelian closure of Q

julia> R = gmodule(CyclotomicField, R)
G-module for t24n201 acting on Vector space of dimension 6
over cyclotomic field of order 5
```

Example

```
julia> schur_index(T)
2

julia> character_field(T)
(Cyclotomic field of order 1, Map from
Cyclotomic field of order 1 to Abelian closure of Q defined by a julia-fun
```

Example

```
julia> gmodule_minimal_field(R)
G-module for t24n201 acting on Vector space of dimension 6
over number field of degree 4 over QQ

julia> B, mB = relative_brauer_group(base_ring(R), character_field(R))
(Relative Brauer group for Cyclotomic field of order 5 over Number field o
, Map from
Relative Brauer group for Cyclotomic field of order 5 over Number field of
 to AllCoChains{2, PermGroupElem, MultGrpElem{nf_elem}}() defined by a jul
```

Example

```
julia> B(R)
Dict{Union{NumFieldOrdIdl, Hecke.NumFieldEmb}, Hecke.QmodnZElem}(
<2, 2>
two normal wrt: 2 => 1//2 + Z,
Complex embedding corresponding ... => 1//2 + Z)
```

Example

```
julia> grunwald_wang(Dict(2*ZZ => 2), Dict(complex_embeddings(QQ)[1] => 2)
Class field defined mod (<8, 8>, InfPlc{AnticNumberField, Hecke.NumFieldEmb

julia> absolute_simple_field(number_field(ans))
(Number field of degree 2 over QQ, Map with following data
Domain:
=======
Number field of degree 2 over QQ
Codomain:
=========
Non-simple number field of degree 2 over number field ...
```

Example

```
julia> compositum(base_ring(R), ans[1])
(Number field of degree 8 over QQ, Map with following data
Domain:
=======
Cyclotomic field of order 5
Codomain:
=========
Number field of degree 8 over QQ, Map with following data
Domain:
=======
Number field of degree 2 over QQ
Codomain:
=========
```

Example

```
julia> gmodule_over(ans[3], gmodule(ans[1], R))
G-module for t24n201 acting on Vector space of dimension 6
over number field of degree 2 over QQ

julia> base_ring(ans)
Number field with defining polynomial x^2 + 2
  over rational field
```

Galois Cohomology

Theorem (Shafarevich)

Let K/k be abelian and $k/\mathbb{Q}$ normal, then canonically

$$H^2(\mathit{Aut}(k/\mathbb{Q}), C_k) = \langle\delta\rangle$$

is cyclic of order $n = k : \mathbb{Q}$. Furthermore

$$1 \to \mathit{Aut}(K/k) \to \mathit{Aut}(K/\mathbb{Q}) \to \mathit{Aut}(k/\mathbb{Q}) \to 1$$

is exact, and

$$H^2(\mathit{Aut}(k/\mathbb{Q}), \mathit{Aut}(K/k)) \ni \mathit{Aut}(K/\mathbb{Q}) = \rho(\delta)$$

for the canonical projection $\rho : C_k \to \mathit{Aut}(K/k)$

Chinburg

C_k, the idel class group, is neither finite, nor finitely presented nor are its elements.

Theorem (Chinburg)

There exists a finitely presented module that is cohomologically equivalent to the idel class group C_k.

- Debeerst (PhD with Bley) made it algorithmic
- Aslam improved the local part
- Implemented in Oscar
- Can compute the local fundamental class
- Can compute global class in many cases

Example

```
julia> k, a = quadratic_field(2);
julia> A = abelian_extensions(k, [2], ZZ(10)^4,
                        signatures = [(4,0)]);
julia> a = A[findall(is_normal, A)];
julia> [describe(galois_group(x, QQ)[1]) for x = a]
3-element Vector{String}:
 "C2 x C2"
 "C2 x C2"
 "C4"
julia> [describe(galois_group(x, QQ)[1]) for x = A]
15-element Vector{String}:
 "C2 x C2"
 "D8"
```