Advanced algebraic number theory

B. Allombert

IMB
CNRS/Université de Bordeaux

08/06/2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 676541

polgalois

We can compute the Galois group of the Galois closure of a number field, as an transitive group. Restricted to degree ≤ 7, or degree ≤ 11 with the `galdata` optional package.

```
P1 = x^4-5;
polgalois(P1)
%2 = [8, -1, 1, "D(4)"]
```

Interpretation: the Galois group has order 8, is not contained in the alternating group, and is isomorphic to D_4.

polgalois

```
P2 = x^4-x^3-7*x^2+2*x+9;
polgalois(P2)
%4 = [12, 1, 1, "A4"]
```

The Galois group has order 12 and signature 1, and is isomorphic to A_4.

```
P3 = x^4-x^3-3*x^2+x-1;
polgalois(P3)
%6 = [24, -1, 1, "S4"]
```

The Galois group has order 24 and signature -1, and is isomorphic to S_4.

nfsplitting

We can compute a polynomial defining the splitting field of the input polynomial, that is, the smallest field over which the input polynomial is a product of linear factors.

```
Q1 = nfsplitting(P1)
%7 = x^8 + 70*x^4 + 15625
Q2 = nfsplitting(P2)
%8 = x^12 - 59*x^10 + 1269*x^8 - 12231*x^6
  + 51997*x^4 - 79707*x^2 + 26569
```

This is the same as a defining polynomial for the Galois closure of the number field generated by one root of the input polynomial.

nfsplitting

The polynomial output by nfsplitting can be large.

```
Q3 = nfsplitting(P3)
%9 = x^24+12*x^23-66*x^22-1232*x^21+735*x^20
 +54012*x^19+51764*x^18-1348092*x^17-2201841*x^16
 +21708244*x^15+41344014*x^14-241723272*x^13
 -454688929*x^12+1972336584*x^11+3130578366*x^10
 -12348327032*x^9-13356023346*x^8+59757161004*x^7
 +32173517686*x^6-204540935496*x^5-111764768888*x^4
 +433089193668*x^3-155456858376*x^2-422808875280*x
 +320938557273
```

polredbest

We can use `polredbest` to compute a simpler polynomial defining the same number field.

```
Q3 = polredbest(Q3)
%10 = x^24-6*x^23+18*x^22-38*x^21+60*x^20-54*x^19
 -13*x^18+126*x^17-228*x^16+220*x^15+24*x^14
 -396*x^13+521*x^12-216*x^11-48*x^10-32*x^9-66*x^8
 +666*x^7-1013*x^6+348*x^5+510*x^4-654*x^3+234*x^2
 +36*x+9
```

galoisinit

We can use `galoisinit` to compute the automorphism group of a number field that is Galois over $\mathbb{Q}$, under certain condition on the group ("weakly super-solvable").

```
gal = galoisinit(Q3);
```

The `gen` component is a list of generators of the automorphism group, expressed as permutations of the roots.

```
gal.gen
%12 = [Vecsmall([19,11,17,14,13,12,10,9,8,7,2,6,5,
 4,23,22,3,21,1,24,18,16,15,20]),Vecsmall([14,10,5,
 19,3,24,11,16,22,2,7,20,17,1,21,8,13,23,4,12,15,9,
 18,6]),Vecsmall([5,15,6,13,20,19,23,7,11,18,21,4,
 12,17,16,2,24,22,3,1,9,10,8,14]),Vecsmall([2,1,9,
 10,16,21,14,17,3,4,19,18,22,7,20,5,8,12,11,15,6,
 13,24,23])]
```

galoisinit

The `orders` components contains orders of composition factors of the group, and their product is the order of the group.

```
ord = gal.orders
%13 = Vecsmall([2, 2, 3, 2])
prod(i=1,#ord,ord[i])
%14 = 24
```

With the function `galoisidentify`, we can obtain the GAP4 index of the group.

```
galoisidentify(gal)
%15 = [24, 12]
```

See `http://pari.math.u-bordeaux.fr/galpol/` for the numbering.

Effective Galois theory

`galoissubgroups` computes the list of all subgroups of a group.

```
L = galoissubgroups(gal);
#L
%17 = 30
```

Then we can compute fixed fields of various subgroups of the Galois group with `galoisfixedfield`.

```
R1 = galoisfixedfield(gal,L[25])[1];
polgalois(R1)
%19 = [24, 1, 1, "S_4(6d) = [2^2]S(3)"]
R2 = galoisfixedfield(gal,L[28])[1];
polgalois(R2)
%21 = [24, -1, 1, "S_4(6c) = 1/2[2^3]S(3)"]
```

Ramification groups

We can compute ramification groups. Let's first find the ramified primes.

```
nf = nfinit(Q3);
factor(nf.disc)
%23 =
[ 3 28]
[11 16]
```

The ramified primes are 3 and 11.

```
dec3 = idealprimedec(nf,3);
pr3 = dec3[1];
[#dec3, pr3.f, pr3.e]
%26 = [4, 1, 6]
```

There are 4 prime ideals above 3. They have residue degree 1 and ramification index 6.

Ramification groups

We compute the sequence of ramification groups with `idealramgroups`.

```
ram3 = idealramgroups(nf,gal,pr3);
#ram3
%28 = 3
```

There are three nontrivial ramification groups to consider.

```
galoisidentify(ram3[1])
%29 = [6, 1]
galoisisabelian(ram3[1])
%30 = 0
```

The decomposition group has order 6, and is isomorphic to S_3.

Ramification groups

```
galoisidentify(ram3[2])
%31 = [6, 1]
```

The inertia group equals the decomposition group (we already knew that since the residue degree is 1).

```
galoisidentify(ram3[3])
%32 = [3, 1]
```

The wild inertia group is the cyclic group C_3, and all the higher ramification groups are trivial.

Ramification groups

```
dec11 = idealprimedec(nf,11);
pr11 = dec11[1];
[#dec11, pr11.f, pr11.e]
%35 = [4, 2, 3]
```

There are 4 prime ideals above 11. They have residue degree 2 and ramification index 3.

```
ram11 = idealramgroups(nf,gal,pr11);
#ram11
%37 = 2
```

The wild ramification group is trivial (which we knew since 11 is coprime to the group order).

Ramification groups

```
galoisidentify(ram11[1])
%38 = [6, 1]
galoisidentify(ram11[2])
%39 = [3, 1]
```

The decomposition group is isomorphic to S_3 (we already knew it had index 4 in the Galois group), and the inertia group is C_3 (we already knew it had index 2 in the decomposition group).

Frobenius elements

At an unramified prime, we can compute the Frobenius element with `idealfrobenius`.

```
dec2 = idealprimedec(nf,2);
pr2 = dec2[1];
[#dec2, pr2.f, pr2.e]
%42 = [6, 4, 1]
frob2 = idealfrobenius(nf,gal,pr2);
permorder(frob2)
%44 = 4
```

We check that the Frobenius element has order equal to the residue degree.

Explicit Kronecker–Weber theorem

We can construct abelian extensions of $\mathbb{Q}$ with `polsubcyclo`.

```
N = 7*13*19;
L1 = polsubcyclo(N,3);
```

We now have the list of degree 3 subfields of $\mathbb{Q}(\zeta_N)$, where $N = 7 \cdot 13 \cdot 19$.

```
L2 = [P | P <- L1, #factor(nfinit(P).disc)[,1]==3]
%47 = [x^3+x^2-576*x+5123, x^3+x^2-576*x-64,
  x^3+x^2-576*x-5251, x^3+x^2-576*x+1665]
```

We select the ones that are ramified at the three primes 7, 13 and 19.

Explicit Kronecker–Weber theorem

We compute the structure and generators of $(\mathbb{Z}/N\mathbb{Z})^\times$ with `znstar`.

```
G = znstar(N)
%48 = [1296, [36, 6, 6], [Mod(743, 1729),
  Mod(248, 1729), Mod(407, 1729)]]
```

We construct the matrix of a specific subgroup of index 3:

```
H = mathnfmodid([1,0;-1,1;0,-1],3);
```

Explicit Kronecker–Weber theorem

We construct the corresponding abelian field.

```
pol = galoissubcyclo(G,H)
%50 = x^3 + x^2 - 576*x - 64
factor(nfinit(pol).disc)
%51 =
[ 7 2]
[13 2]
[19 2]
```

We check the ramification of the corresponding number field.

Hilbert class field

To compute a Hilbert class field, we first need to compute the class group.

```
bnf = bnfinit(a^2+23);
bnf.cyc
%53 = [3]
```

The class group is isomorphic to $\mathbb{Z}/3\mathbb{Z}$.

```
bnr = bnrinit(bnf,1);
bnr.mod
%55 = [[1, 0; 0, 1], []]
```

Technical point: we need to represent the class group as a ray class group with `bnrinit`.

Hilbert class field

Now we compute the Hilbert class field with `rnfkummer`.

```
R = rnfkummer(bnr)
%56 = x^3 - 3*x + Mod(a, a^2 + 23)
```

Conversely, from an abelian extension, we can recover its corresponding class group `rnfconductor`.

```
[cond,bnr,subg] = rnfconductor(bnf,R);
cond
%58 = [[1, 0; 0, 1], []]
subg
%59 = [3]
```

Here the conductor is trivial, and its norm group is trivial in the class group.

Ray class fields

We can also consider class fields with nontrivial conductor.

```
bnf = bnfinit(a^2+3);
bnr = bnrinit(bnf,6);
```

We can compute in advance the absolute degree, signature and discriminant of the corresponding class field with `bnrdisc`.

```
[deg,r1,D] = bnrdisc(bnr);
deg
%63 = 6
r1
%64 = 0
D
%65 = -34992
```

Ray class fields

We can also compute the corresponding relative information using the `bnrdisc(...,1)` option.

```
[degrel,r1rel,Drel] = bnrdisc(bnr,,,1);
degrel
%67 = 3
r1rel
%68 = 0
Drel
%69 =
[36  0]
[ 0 36]
```

The relative discriminant is given in HNF form.

Ray class fields

We compute a defining polynomial with `rnfkummer`. This can be time-consuming, so it is better to compute the relevant information without constructing the field, if possible.

```
R = rnfkummer(bnr)
%70 = x^3 - 2
P = rnfequation(bnf,R)
%71 = x^6 + 9*x^4 - 4*x^3 + 27*x^2 + 36*x + 31
nf = nfinit(P);
```

We check the absolute discriminant and signature.

```
nf.disc
%73 = -34992
nf.sign
%74 = [0, 3]
```

Ray class fields

We can also compute Frobenius information without the explicit field.

```
id31 = idealprimedec(bnf,31)[1];
bnrisprincipal(bnr,id31,0)
%76 = [0]~
```

The Frobenius at $\mathfrak{p}_{31}$ is trivial.

```
ispower(Mod(2,31),3)
%77 = 1
```

Indeed, 31 is split in $F(\sqrt[3]{2})$.

Analytic class number formula

We compute the value at 0 of the first derivative of the Hecke L-function attached to a nontrivial character χ of our class group, using `lfun`.

```
r = lfun([bnr,[1]],0,1)
%78 = 1.3473773483293841009181878914456 + 0.E-61*I
```

By the factorisation of the Dedekind zeta function and the analytic class number formula, we can recover a unit in the class field.

```
R2 = algdep(exp(r),3)
%79 = x^3 - 3*x^2 - 3*x - 1
P2 = rnfequation(bnf,R2);
nfisisom(P2,nf)!=0
%81 = 1
```

We reconstructed the class field using the L-function.

Modulus with infinite places

If the base field has real places, we can specify the modulus at infinity by providing a list of 0 or 1 of length the number of real embeddings.

```
bnf=bnfinit(a^2-217);
bnf.cyc
%83 = []
bnrinit(bnf,1).cyc
%84 = []
bnrinit(bnf,[1,[1,1]]).cyc
%85 = [2]
```

The field $\mathbb{Q}(\sqrt{217})$ has narrow class number 2.

Transcendental methods

For quadratic fields, ray class groups can be computed using transcendental methods using `quadhilbert` and `quadray`.

```
quadhilbert(-31)
%86 = x^3 + x^2 + 1
lift(quadray(13,7))
%87 = x^3 + (-7*y - 11)*x^2 + (56*y + 73)*x
  + (-91*y - 118)
```

With `rnfkummer`, the cost of the computation mostly depends on the degree of the extension but not much on the conductor; with transcendental methods, the cost mostly depends on the conductor.

Galois action on the class group

We can compute the Galois action on ray class groups with `bnrgaloismatrix`, i.e. the Galois action on the relative Galois group, without the explicit abelian extension.

```
bnf = bnfinit(x^2+2*3*5*7*11);
bnf.cyc
%89 = [4, 2, 2, 2]
bnr = bnrinit(bnf,1,1);
gal = galoisinit(bnf);
m = bnrgaloismatrix(bnr,gal)[1]
%92 =
[3 0 0 0]
[0 1 0 0]
[0 0 1 0]
[0 0 0 1]
```

Questions ?

Have fun with GP !