THIRD LECTURE FOR THE RESEARCH SCHOOL CIMPA 2015: "ALGEBRAIC NUMBER FIELDS AND CLASS FIELDS"

DANIEL C. MAYER

1. Algebraic number fields

1.1. **The Herbrand Quotient viewed from Cohomology.** Let $N|K$ be a cyclic relative extension of number fields with relative degree $n := [N : K]$ and automorphism group $G = \mathrm{Gal}(N|K) = \langle \sigma \rangle$ generated by σ.

We consider the Galois cohomology of the unit groups U_N and U_K of N and K, viewed as G-modules. Let two endomorphisms of U_N be defined by

$$\Delta : U_N \to U_N,\ E \mapsto E^{\sigma-1} \quad \text{and} \quad \mathcal{N} : U_N \to U_N,\ E \mapsto E^{1+\sigma+\cdots+\sigma^{n-1}}.$$

Then we obviously have $\Delta \circ \mathcal{N} = \mathcal{N} \circ \Delta = 1$.

Remark 1.1. It is usual to write the action of a group G on a G-module as powers with symbolic exponents. In particular, $E^{\sigma-1} = \frac{\sigma(E)}{E}$ and $E^{1+\sigma+\cdots+\sigma^{n-1}} = E \cdot \sigma(E) \cdots \sigma^{n-1}(E)$.

Definition 1.1. The *Herbrand quotient* of the G-module U_N is defined as

$$h(G, U_N) := \frac{\#\mathrm{H}^0(G, U_N)}{\#\mathrm{H}^{-1}(G, U_N)},$$

where

$$\mathrm{H}^0(G, U_N) := \ker(\mathcal{N})/\mathrm{im}(\Delta) \simeq U_K/\mathrm{Norm}_{N|K}(U_N)$$

and

$$\mathrm{H}^{-1}(G, U_N) := \ker(\Delta)/\mathrm{im}(\mathcal{N}) \simeq E_{N|K}/U_N^{\sigma-1}.$$

Here we denote by $E_{N|K}$ the subgroup of relative units $E \in U_N$ such that $\mathrm{Norm}_{N|K}(E) = 1$.

Theorem 1.1. *(Herbrand, 1932)*
If $N|K$ is cyclic of prime degree $n = p$ and r of the real Archimedean places of K become complex in N (note that $r > 0$ can only occur for $p = 2$), then the Herbrand quotient of U_N is given by

$$h(G, U_N) = p^{r-1} \quad \textit{resp.} \quad \frac{(U_K : \mathrm{Norm}_{N|K}(U_N))}{(E_{N|K} : U_N^{\sigma-1})} = \frac{2^r}{[N : K]}.$$

Date: May 05, 2015.
2000 *Mathematics Subject Classification.* Primary 11Y40, 11Y65, 14H52, 14K22, 11R37, 11R29, 11R11; Secondary 11Y11, 11Y05.
Key words and phrases. algebraic number theory computations, continued fractions, elliptic curves, complex multiplication, primality, factorization, quadratic fields.

We now discuss a few important applications of Theorem 1.1 on the Herbrand quotient. In particular, we draw conclusions for groups of *ambiguous principal ideals* which turned out to be crucial for the classification of *dihedral* and *pure metacyclic* absolute extensions $N|\mathbb{Q}$.

Theorem 1.2. *(The first assertion is due to Iwasawa, 1956)*

(1) *If $N|K$ is a* **Galois** *extension with group $G = \mathrm{Gal}(N|K)$, then the first cohomology group $\mathrm{H}^1(G, U_N)$ of the G-module U_N is isomorphic to the quotient of the group $\mathcal{P}_N^G$ of ambiguous principal ideals of N by the subgroup $\mathcal{P}_K$ of principal ideals of K, since the map*

$$\mathcal{P}_N^G \to \mathrm{H}^1(G, U_N),\ (A) \mapsto (A^{\tau-1})_{\tau \in G}$$

is an epimorphism with kernel $\mathcal{P}_K$.

(2) *In particular, if $N|K$ is* **cyclic** *of* **odd** *prime degree $p \geq 3$, then the Galois cohomology of U_N is periodic with length 2, and therefore*

$$\#(\mathcal{P}_N^G/\mathcal{P}_K) = \#\mathrm{H}^1(G, U_N) = \#\mathrm{H}^{-1}(G, U_N) = \#\mathrm{H}^0(G, U_N) \cdot [N : K].$$

(3) *Even more specifically, if $N|K$ is* **unramified**, *then the order of the p-principalization kernel is given by*

$$\#\ker(j_{N|K}) = \#(\mathcal{P}_N^G/\mathcal{P}_K) = (U_K : \mathrm{Norm}_{N|K}(U_N)) \cdot [N : K],$$

where $j_{N|K} : \mathrm{Cl}_p(K) \to \mathrm{Cl}_p(N)$ denotes the extension homomorphism of the p-class groups of K and N.

Theorem 1.3. *Let the base field $K = \mathbb{Q}(\sqrt{d})$ be quadratic with fundamental discriminant $d \in \mathbb{Z}$ and $N|K$ be cyclic of odd prime degree $p \geq 3$ with conductor $f \geq 1$ such that $N|\mathbb{Q}$ is dihedral of order $2p$.*

1.2. **Remarks on Algorithms.** A few general suggestions for writing a script in any programming language:

- Reduce the CPU time complexity of an algorithm by *avoiding superfluous nested loops.*
- Construct an algorithm by means of the *top-down design*,
  using the *principle of successive refinements.* This will
  (1) clarify the *structure* of the source code,
  (2) admit the separation of measuring the *CPU time of subalgorithms*,
  (3) enable easy *reusage of subalgorithms* in other algorithms.

Example 1.1. Suppose we want to *classify* a sequence of number fields K with fixed absolute degree $[K : \mathbb{Q}] = n$ and fixed signature (r, c) such that $r + 2c = n$, according to certain invariants $i(K)$ with finitely many possible values $i(K) \in \{v_1, \ldots, v_n\}$.

Then a coarse first draft of the algorithm is given as follows.

(1) We precompute a list of *characterizing parameters* (such as discriminants d_K, or conductors c_K, etc.) for the desired sequence of base fields K.

Then we loop through the list and for each individual parameter entry d of the list we perform the following steps.

(2) We construct the base field K characterized by d.
(3) We construct extensions L of the base field K which are needed for determining the desired invariants $i(K)$.
(4)

1.3. **Construction of a series of base fields.** The following Magma code snippet shows an *intrinsic function* with explicit *signature* (list of parameters and their types) for sifting members with p-class group $\mathrm{Cl}_p(K)$ of type (p,p) from a sequence of complex quadratic fields $K=\mathbb{Q}(\sqrt{d})$ given by their discriminant $d<0$.

```
intrinsic CmpQdrPxP(d::RngIntElt,p::RngIntElt){}

      ZX<X> := PolynomialRing(Integers());
      K := NumberField(X^2+d); // base field
      // K := QuadraticField(-d); // variant without X
      O := MaximalOrder(K);
      SetClassGroupBounds("GRH");
      C,mC := ClassGroup(O);

      if ([p,p] eq pPrimaryInvariants(C,p)) then
         printf "%7o,\n",d;
         ...
end intrinsic; // CmpQdrPxP
```

1.4. **Construction of extension fields.**

```
        sA := [AbelianExtension(Inverse(mQ)*mC)
               where Q,mQ := quo<C|x`subgroup>: x in sS];
        sN := [NumberField(x): x in sA];
        sR := [MaximalOrder(x): x in sA];
        sF := [AbsoluteField(x): x in sN];
        sM := [MaximalOrder(x): x in sF];
        sM := [OptimizedRepresentation(x): x in sF];
        sA := [NumberField(DefiningPolynomial(x)): x in sM];
        sO := [Simplify(LLL(MaximalOrder(x))): x in sA];
```

1.5. **Principalization with explicit Artin Map.** We want to determine the exact p-principalization type $\varkappa(K):=(\ker(j_{L_i|K}))_{1\le i\le n}$ of a number field K in a multiplet $L_1,\ldots,L_n$ of abelian extensions of relative degree p sharing a common conductor c, where each $j_{L_i|K}:\mathrm{Cl}_p(K)\to\mathrm{Cl}_p(L_i)$ denotes the class extension homomorphism from K to L_i.

$\varkappa(K)$ should include information

- on fixed points (Taussky's conditions A and B) and
- on (partial or complete) cycle patterns,

and thus can only be determined with the aid of an explicit Artin-Galois correspondence

$$N_i \mapsto \mathrm{Gal}(\mathrm{F}_c(K)|L_i) \mapsto L_i$$

between norm subgroups $N_i := \mathrm{Norm}(\mathrm{Cl}_p(L_i))$ of the class group $\mathrm{Cl}_p(K)$ and the extension fields $L_i|K$ within the ray class field modulo c, $\mathrm{F}_c(K)$, of K.

```
sS := Subgroups(C: Quot := [p]);
sI := [];
for j in [1..p+1] do
   Append(~sI,0);
end for; // j
n := Ngens(C);
ct := 0; // local counter
for x in sS do
   ct := ct+1;
   if (Order(C.(n-1)) div p)*C.(n-1) in x`subgroup then
      sI[1] := ct;
   end if; // n-1
   if (Order(C.n) div p)*C.n in x`subgroup then
      sI[2] := ct;
   end if; // n
   for e in [1..p-1] do
      if ((Order(C.(n-1)) div p)*C.(n-1))+(e*(Order(C.n) div p)*C.n) in x`subgrou
         sI[e+2] := ct;
      end if; // product
   end for; // e
end for; // x
```

1.6. **Computation of the Artin pattern.**

```
TTT := [];
for j in [1..#sO] do
   CO := ClassGroup(sO[j]);
   Append(~TTT,pPrimaryInvariants(CO,p));
end for; // j

TKT := [];
for j in [1..#sR] do
```

```
   I := sR[j]!!mC( (Order(C.(n-1)) div p)*C.(n-1) );
   if IsPrincipal(I) then
      Append(~TKT,sI[1]);
   end if; // I
   I := sR[j]!!mC( (Order(C.n) div p)*C.n );
   if IsPrincipal(I) then
     Append(~TKT,sI[2]);
   end if; // I
   for e in [1..p-1] do
      I := sR[j]!!mC( ((Order(C.(n-1)) div p)*C.(n-1)+(e*(Order(C.n) div p)*C.n))
      if IsPrincipal(I) then
         Append(~TKT,sI[e+2]);
      end if; // I
   end for; // e
end for; // j

TAB := []; // Taussky conditions A and B
for j in [1..#TKT] do
   if (j eq TKT[j]) then
      Append(~TAB,"A");
   else
      Append(~TAB,"B");
   end if; // fixed point
end for; // j

printf "%o; %o; ( ",TKT,TAB;
for j in [1..#TTT] do
   printf "%o ",TTT[j];
end for; // j
printf ")\n";
```

2. Unit norm index

By means of the following Magma program script, the unit norm index $(U_K : \mathrm{Norm}_{N|K} U_N)$ can be determined.

```
intrinsic PurQnt(d::RngIntElt){}
   printf "%3o:\n",d;
```

```
   ZX<X> := PolynomialRing(Integers());
   P4 := X^4+X^3+X^2+X+1;
   F<z> := NumberField(P4);

   FY<Y> := PolynomialRing(F);
   Pr5 := Y^5-d;
   K<w> := ext<F|Pr5>;

   Ka := AbsoluteField(K);
   ZKa := MaximalOrder(Ka);
   SetClassGroupBounds("GRH");
   CKa,mCKa := ClassGroup(ZKa);
   UKa,mUKa := IndependentUnits(ZKa);
   SetOrderUnitsAreFundamental(ZKa);
   U,m := UnitGroup(ZKa);

   r,c := Signature(Ka);
   for j in [1..r+c] do
      ur := K ! m(U.j);
      printf "%12o: %o\n",j,Norm(ur);
   end for; // j
end intrinsic; // PurQnt
```

3. Class number relation

By means of the following Magma program script, the subfield unit index $(U_N : \prod_{F<N} U_F) = 5^u$, resp. its 5-logarithm u, can be determined from the class numbers of N, L and K using Parry's formula

$$h_N = \frac{(U_N : \prod_{F<N} U_F)}{5^5} h_K h_L^4,$$

where $h_K = 1$.

```
intrinsic ParryAbs(d::RngIntElt){}

    printf "%3o: ",d;
    p := 5;

    ZX<X> := PolynomialRing(Integers());

    // Cyclotomic (cyclic quartic) field
    P4 := X^4+X^3+X^2+X+1;
```

```
    F := NumberField(P4);

    // Maximal real (quadratic) subfield
    P2 := X^2-5;
    F0 := NumberField(P2);

    SetClassGroupBounds("GRH");

    // Pure quintic field
    P5 := X^5-d;
    L := NumberField(P5);
    ZL := MaximalOrder(L);
    C,mC := ClassGroup(ZL);
    V := Valuation(Order(C),p);
    P := pPrimaryInvariants(C,p);

    // Intermediate non-Galois field
    K0a := Compositum(F0,L);
    ZK0a := MaximalOrder(K0a);
    C0a,mC0a := ClassGroup(ZK0a);
    V0a := Valuation(Order(C0a),p);
    P0a := pPrimaryInvariants(C0a,p);

    // Normal closure of pure quintic field
    Ka := Compositum(F,L);
    ZKa := MaximalOrder(Ka);
    Ca,mCa := ClassGroup(ZKa);
    Va := Valuation(Order(Ca),p);
    Pa := pPrimaryInvariants(Ca,p);

    U := Va + 5 - 4*V;
    printf "%3o, %o, %o, %o\n",U,P,P0a,Pa;

end intrinsic; // ParryAbs
```

4. Hilbert class fields

5. Ray class fields

6. Ring class fields

Theorem 6.1. *(Regulator Quotient Criterion)*
Let $p \geq 3$ be an odd prime and K be a real quadratic field with discriminant $d > 0$ and ordinary p-class rank $\varrho_p = 0$. Then K has modified p-class rank $\sigma_p = 1$ and the following characterization of p-admissible conductors q over K holds.

(1) *A prime q, or the prime power $q = p^2$, is a free regular conductor over K, $\delta_p(q) = 0$, if and only if the quotient of the regulator $\mathrm{R}(q)$ of the suborder $\mathcal{O}_q$ by the regulator $\mathrm{R}(1)$ of the maximal order $\mathcal{O}$ of K satisfies the condition*

$$v_p(\mathrm{R}(q)/\mathrm{R}(1)) < v_p(E(q)),$$

where

$$v_p(E(q)) = \begin{cases} v_p(q-1) & \text{for } q \equiv +1 \pmod{p},\ \left(\frac{d}{q}\right) = +1, \\ v_p(q+1) & \text{for } q \equiv -1 \pmod{p},\ \left(\frac{d}{q}\right) = -1, \\ 1 & \text{for } q = p,\ p \mid d, \\ 1 & \text{for } q = p^2,\ \left(\frac{d}{p}\right) = \pm 1. \end{cases}$$

(2) *If $p = 3$ and K has discriminant $d \equiv -3 \pmod 9$ and thus enables irregular 3-admissible conductors, then the following criteria hold for the irregular prime power conductor $q = 3^2 = 9$.*

$$\delta_3(9) = 0 \iff v_3(\mathrm{R}(9)/\mathrm{R}(1)) = 0,$$

$$\delta_3(9) = 1 \iff v_3(\mathrm{R}(9)/\mathrm{R}(1)) = 1.$$

References

[1] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), 235–265.

[2] W. Bosma, J. J. Cannon, C. Fieker, and A. Steels (eds.), *Handbook of Magma functions* (Edition 2.21, Sydney, 2015).

[3] H. Cohen, *A course in computational algebraic number theory*, Graduate texts in mathematics **138**, Springer, 1996.

[4] H. Cohen, *Advanced topics in computational number theory*, Graduate texts in mathematics **193**, Springer, 2000.

[5] The GAP Group, *GAP – Groups, Algorithms, and Programming — a System for Computational Discrete Algebra*, Version 4.7.7, Aachen, Braunschweig, Fort Collins, St. Andrews, 2015, (`http://www.gap-system.org`).

[6] The MAGMA Group, *MAGMA Computational Algebra System*, Version 2.21-3, Sydney, 2015, (`http://magma.maths.usyd.edu.au`).

[7] Oracle, *JDK* 7u80, *Java SE*, and *NetBeans*, Version 8.0.2, Redwood City, CA, 2015, (`http://www.oracle.com`).

[8] The PARI Group, *PARI/GP*, Version 2.7.3, Bordeaux, 2015, (`http://pari.math.u-bordeaux.fr`).

[9] M. Pohst and H. Zassenhaus, *Algorithmic algebraic number theory*, Encyclopedia of mathematics and its applications, Cambridge University Press, 1990.

[10] S.S. Wagstaff, Jr., *The Joy of Factoring*, Student Mathematical Library (STML), Vol. **68**, American Mathematical Society (AMS), 2013.

Naglergasse 53, 8010 Graz, Austria

E-mail address: `algebraic.number.theory@algebra.at`

URL: `http://www.algebra.at`