Math 129: Algebraic Number Theory
Lecture 5

William Stein

Thursday, February 19, 2004

1 Dedekind Domains

First we complete the proof begun on Tuesday that the set of nonzero fractional ideals of $\mathcal{O}_K$ is a group. Recall that a fractional ideal I is an $\mathcal{O}_K$-submodule of K that is finitely generated. We proved on Tuesday that if $\mathfrak{p}$ is a prime ideal of I, then there is a fractional ideal

$$\mathfrak{p}^{-1} = \{a \in \mathbf{K} : a\mathfrak{p} \subset \mathcal{O}_K\}$$

such that $\mathfrak{p}\mathfrak{p}^{-1} = (1) = \mathcal{O}_K$, i.e., every prime ideal has an inverse. We proved this by noting that either $\mathfrak{p}\mathfrak{p}^{-1} = \mathcal{O}_K$ or $\mathfrak{p}\mathfrak{p}^{-1} = \mathfrak{p}$. Using the fact that $\mathcal{O}_K$ is Noetherian and integrally closed, we deduced that $\mathfrak{p}\mathfrak{p}^{-1} = \mathfrak{p}$ leads to a contradiction. It remains to prove that every fractional ideal has an inverse, which we do now.

Completion of proof. So far we have proved that if $\mathfrak{p}$ is a prime ideal of $\mathcal{O}_K$, then $\mathfrak{p}^{-1} = \{a \in \mathbf{K} : a\mathfrak{p} \subset \mathcal{O}_K\}$ is the inverse of $\mathfrak{p}$ in the monoid of nonzero fractional ideals of $\mathcal{O}_K$. As mentioned after Definition **??** [on Tuesday], every nonzero fractional ideal is of the form aI for $a \in K$ and I an integral ideal, so since (a) has inverse $(1/a)$, it suffices to show that every integral ideal I has an inverse. If not, then there is a nonzero integral ideal I that is maximal among all nonzero integral ideals that do not have an inverse. Every ideal is contained in a maximal ideal, so there is a nonzero prime ideal $\mathfrak{p}$ such that $I \subset \mathfrak{p}$. Multiplying both sides of this inclusion by $\mathfrak{p}^{-1}$ and using that $\mathcal{O}_K \subset \mathfrak{p}^{-1}$, we see that $I \subset \mathfrak{p}^{-1}I \subset \mathcal{O}_K$. If $I = \mathfrak{p}^{-1}I$, then arguing as in the proof that $\mathfrak{p}^{-1}$ is the inverse of $\mathfrak{p}$, we see that each element of $\mathfrak{p}^{-1}$ preserves the finitely generated $\mathbf{Z}$-module I and is hence integral. But then $\mathfrak{p}^{-1} \subset \mathcal{O}_K$, which implies that $\mathcal{O}_K = \mathfrak{p}\mathfrak{p}^{-1} \subset \mathfrak{p}$, a contradiction. Thus $I \neq \mathfrak{p}^{-1}I$. Because I is maximal among ideals that do not have an inverse, the ideal $\mathfrak{p}^{-1}I$ does have an inverse, call it J. Then $\mathfrak{p}J$ is the inverse of I, since $\mathcal{O}_K = (\mathfrak{p}J)(\mathfrak{p}^{-1}I) = JI$. $\square$

We are now ready to deduce the crucial Theorem 1.2, which asserts that any nonzero ideal of a Dedekind domain can be expressed uniquely as a product of primes (up to order). Thus unique factorization holds for ideals in a Dedekind domain, and it is this unique factorization that initially motivated the introduction of rings of integers of number fields over a century ago.

Theorem 1.1. *Suppose I is an integral ideal of $\mathcal{O}_K$. Then I can be written as a product*

$$I = \mathfrak{p}_1 \cdots \mathfrak{p}_n$$

of prime ideals of $\mathcal{O}_K$, and this representation is unique up to order. (Exception: If $I = 0$, then the representation is not unique.)

Proof. Suppose I is an ideal that is maximal among the set of all ideals in $\mathcal{O}_K$ that can not be written as a product of primes. Every ideal is contained in a maximal ideal, so I is contained in a nonzero prime ideal $\mathfrak{p}$. If $I\mathfrak{p}^{-1} = I$, then by Theorem **??** [that the fractional ideals form an abelian group] we can cancel I from both sides of this equation to see that $\mathfrak{p}^{-1} = \mathcal{O}_K$, a contradiction. Thus I is strictly contained in $I\mathfrak{p}^{-1}$, so by our maximality assumption on I there are maximal ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ such that $I\mathfrak{p}^{-1} = \mathfrak{p}_1 \cdots \mathfrak{p}_n$. Then $I = \mathfrak{p} \cdot \mathfrak{p}_1 \cdots \mathfrak{p}_n$, a contradiction. Thus every ideal can be written as a product of primes.

Suppose $\mathfrak{p}_1 \cdots \mathfrak{p}_n = \mathfrak{q}_1 \cdots \mathfrak{q}_m$. If no $\mathfrak{q}_i$ is contained in $\mathfrak{p}_1$, then for each i there is an $a_i \in \mathfrak{q}_i$ such that $a_i \notin \mathfrak{p}_1$. But the product of the a_i is in the $\mathfrak{p}_1 \cdots \mathfrak{p}_n$, which is a subset of $\mathfrak{p}_1$, which contradicts the fact that $\mathfrak{p}_1$ is a prime ideal. Thus $\mathfrak{q}_i = \mathfrak{p}_1$ for some i. We can thus cancel $\mathfrak{q}_i$ and $\mathfrak{p}_1$ from both sides of the equation. Repeating this argument finishes the proof of uniqueness. □

Corollary 1.2. *If I is a fractional ideal of $\mathcal{O}_K$ then there exists prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ and $\mathfrak{q}_1, \ldots, \mathfrak{q}_m$, unique up to order, such that*

$$I = (\mathfrak{p}_1 \cdots \mathfrak{p}_n)(\mathfrak{q}_1 \cdots \mathfrak{q}_m)^{-1}.$$

Proof. We have $I = (a/b)J$ for some $a, b \in \mathcal{O}_K$ and integral ideal J. Applying Theorem 1.2 to (a), (b), and J gives an expression as claimed. For uniqueness, if one has two such product expressions, multiply through by the denominators and use the uniqueness part of Theorem 1.2 □

Example 1.3. The ring of integers of $K = \mathbf{Q}(\sqrt{-6})$ is $\mathcal{O}_K = \mathbf{Z}[\sqrt{-6}]$. In $\mathcal{O}_K$, we have

$$6 = -\sqrt{-6}\sqrt{-6} = 2 \cdot 3.$$

If $ab = \sqrt{-6}$, with $a, b \in \mathcal{O}_K$ and neither a unit, then $\mathrm{Norm}(a)\,\mathrm{Norm}(b) = 6$, so without loss $\mathrm{Norm}(a) = 2$ and $\mathrm{Norm}(b) = 3$. If $a = c + d\sqrt{-6}$, then $\mathrm{Norm}(a) = c^2 + 6d^2$; since the equation $c^2 + 6d^2 = 2$ has no solution with $c, d \in \mathbf{Z}$, there is no element in $\mathcal{O}_K$ with norm 2, so $\sqrt{-6}$ is irreducible. Also, $\sqrt{-6}$ is not a unit times 2 or times 3, since again the norms would not match up. Thus 6 can not be written uniquely as a product of irreducibles in $\mathcal{O}_K$. Theorem 1.1, however, implies that the principal ideal (6) can, however, be written uniquely as a product of prime ideals. Using MAGMA we find such a decomposition:

```
> R<x> := PolynomialRing(RationalField());
> K := NumberField(x^2+6);
> OK := MaximalOrder(K);
> [K!b : b in Basis(OK)];
[
    1,
    K.1    // this is sqrt(-6)
]
> Factorization(6*OK);
[
    <Prime Ideal of OK
    Two element generators:
        [2, 0]
        [2, 1], 2>,
    <Prime Ideal of OK
    Two element generators:
        [3, 0]
        [3, 1], 2>
]
```

The output means that

$$(6) = (2, 2 + \sqrt{-6})^2 \cdot (3, 3 + \sqrt{-6})^2,$$

where each of the ideals $(2, 2+\sqrt{-6})$ and $(3, 3+\sqrt{-6})$ is prime. I will discuss algorithms for computing such a decomposition in detail, probably next week. The first idea is to write $(6) = (2)(3)$, and hence reduce to the case of writing the (p), for $p \in \mathbf{Z}$ prime, as a product of primes. Next one decomposes the Artinian ring $\mathcal{O}_K \otimes \mathbf{F}_p$ as a product of local Artinian rings.

2 Algorithms for Algebraic Number Theory

The best overall reference for algorithms for doing basic algebraic number theory computations is Henri Cohen's book *A Course in Computational Algebraic Number Theory*, Springer, GTM 138.

Our main long-term algorithmic goals for this course are to understand good algorithms for solving the following problems in particular cases:

- **Ring of integers:** Given a number field K (by giving a polynomial), compute the full ring $\mathcal{O}_K$ of integers.
- **Decomposition of primes:** Given a prime number $p \in \mathbf{Z}$, find the decomposition of the ideal $p\mathcal{O}_K$ as a product of prime ideals of $\mathcal{O}_K$.
- **Class group:** Compute the group of equivalence classes of nonzero ideals of $\mathcal{O}_K$, where I and J are equivalent if there exists $\alpha \in \mathcal{O}_K$ such that $IJ^{-1} = (\alpha)$.

- **Units:** Compute generators for the group of units of $\mathcal{O}_K$.

As we will see, somewhat surprisingly it turns out that algorithmically by far the most time-consuming step in computing the ring of integers $\mathcal{O}_K$ seems to be to factor the discriminant of a polynomial whose root generates the field K. The algorithm(s) for computing $\mathcal{O}_K$ are quite complicated to describe, but the first step is to factor this discriminant, and it takes much longer in practice than all the other complicated steps.

3 Using MAGMA

This section is a first introduction to MAGMA for algebraic number theory. MAGMA is probably the best general purpose program for doing algebraic number theory computations. You can use it via the web page `http://modular.fas.harvard.edu/calc`. MAGMA is not free, but if you would like a copy for your personal computer, send me an email, and I can arrange for you to obtain a legal copy for free.

Five minute tour of the MAGMA web page and documentation.

The following examples illustrate what we've done so far in the course using MAGMA, and a little of where we are going. Feel free to ask questions as we go.

3.1 Smith Normal Form

On the first day of class we learned about Smith normal forms of matrices.

```
> A := Matrix(2,2,[1,2,3,4]);
> A;
[1 2]
[3 4]
> SmithForm(A);
[1 0]
[0 2]

[ 1  0]
[-1  1]

[-1  2]
[ 1 -1]
```

As you can see, MAGMA computed the Smith form, which is $\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$. What are the other two matrices it output? To see what any MAGMA command does, type the command by itself with no arguments followed by a semicolon.

```
> SmithForm;
Intrinsic 'SmithForm'

Signatures:

    (<Mtrx> X) -> Mtrx, AlgMatElt, AlgMatElt
    [
        k: RngIntElt,
        NormType: MonStgElt,
        Partial: BoolElt,
        RightInverse: BoolElt
    ]

        The smith form S of X, together with unimodular matrices
        P and Q such that P * X * Q = S.
```

As you can see, `SmithForm` returns three arguments, a matrix and matrices P and Q that transform the input matrix to Smith normal form. The syntax to "receive" three return arguments is natural, but uncommon in other programming languages:

```
> S, P, Q := SmithForm(A);
> S;
[1 0]
[0 2]
> P;
[ 1  0]
[-1  1]
> Q;
[-1  2]
[ 1 -1]
> P*A*Q;
[1 0]
[0 2]
```

Next, let's test the limits. We make a 10×10 integer matrix with entries between 0 and 99, and compute its Smith normal form.

```
> A := Matrix(10,10,[Random(100) : i in [1..100]]);
> time B := SmithForm(A);
Time: 0.000
```

Let's print the first row of A, the first and last row of B, and the diagonal of B:

```
> A[1];
( 4 48 84  3 58 61 53 26  9  5)
> B[1];
```

```
(1 0 0 0 0 0 0 0 0 0)
> B[10];
(0 0 0 0 0 0 0 0 0 51805501538039733)
> [B[i,i] : i in [1..10]];
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 51805501538039733 ]
```

Let's see how big we have to make A in order to slow down MAGMA. These timings below are on a 1.6Ghz Pentium 4-M laptop running Magma V2.11 under VMware Linux. I tried exactly the same computation running Magma V2.17 natively under Windows XP on the same machine, and it takes *twice* as long to do each computation, which is strange.

```
> n := 50; A := Matrix(n,n,[Random(100) : i in [1..n^2]]);
> time B := SmithForm(A);
Time: 0.050
> n := 100; A := Matrix(n,n,[Random(100) : i in [1..n^2]]);
> time B := SmithForm(A);
Time: 0.800
> n := 150; A := Matrix(n,n,[Random(100) : i in [1..n^2]]);
> time B := SmithForm(A);
Time: 4.900
> n := 200; A := Matrix(n,n,[Random(100) : i in [1..n^2]]);
> time B := SmithForm(A);
Time: 19.160
```

MAGMA can also work with finitely generated abelian groups.

```
> G := AbelianGroup([3,5,18]);
> G;
Abelian Group isomorphic to Z/3 + Z/90
Defined on 3 generators
Relations:
    3*G.1 = 0
    5*G.2 = 0
    18*G.3 = 0
> #G;
270
> H := sub<G | [G.1+G.2]>;
> #H;
15
> G/H;
Abelian Group isomorphic to Z/18
Defined on 1 generator
Relations:
    18*$.1 = 0
```

3.2 $\overline{\mathbf{Q}}$ and Number Fields

MAGMA has many commands for doing basic arithmetic with $\overline{\mathbf{Q}}$.

```
> Qbar := AlgebraicClosure(RationalField());
> Qbar;
> S<x> := PolynomialRing(Qbar);
> r := Roots(x^3-2);
> r;
[
    <r1, 1>,
    <r2, 1>,
    <r3, 1>
]
> a := r[1][1];
> MinimalPolynomial(a);
x^3 - 2
> s := Roots(x^2-7);
> b := s[1][1];
> MinimalPolynomial(b);
x^2 - 7
> a+b;
r4 + r1
> MinimalPolynomial(a+b);
x^6 - 21*x^4 - 4*x^3 + 147*x^2 - 84*x - 339
> Trace(a+b);
0
> Norm(a+b);
-339
```

There are few commands for general algebraic number fields, so usually we work in specific finitely generated subfields:

```
> MinimalPolynomial(a+b);
x^6 - 21*x^4 - 4*x^3 + 147*x^2 - 84*x - 339
> K := NumberField($1) ;  // $1 = result of previous computation.
> K;
Number Field with defining polynomial x^6 - 21*x^4 - 4*x^3 +
    147*x^2 - 84*x - 339 over the Rational Field
```

We can also define relative extensions of number fields and pass to the corresponding absolute extension.

```
> R<x> := PolynomialRing(RationalField());
> K<a> := NumberField(x^3-2);   // a is the image of x in Q[x]/(x^3-2)
> a;
```

```
a
> a^3;
2
> S<y> := PolynomialRing(K);
> L<b> := NumberField(y^2-a);
> L;
Number Field with defining polynomial y^2 - a over K
> b^2;
a
> b^6;
2
> AbsoluteField(L);
Number Field with defining polynomial x^6 - 2 over the Rational
Field
```

3.3 Rings of integers

MAGMA computes rings of integers of number fields.

```
> RingOfIntegers(K);
Maximal Equation Order with defining polynomial x^3 - 2 over ZZ
> RingOfIntegers(L);
Maximal Equation Order with defining polynomial x^2 + [0, -1, 0]
over its ground order
```

Sometimes the ring of integers of $\mathbf{Q}(a)$ isn't just $\mathbf{Z}[a]$. First a simple example, then a more complicated one:

```
> K<a> := NumberField(2*x^2-3);   // doesn't have to be monic
> 2*a^2 - 3;
0
> K;
Number Field with defining polynomial x^2 - 3/2 over the Rational
Field
> O := RingOfIntegers(K);
> O;
Maximal Order of Equation Order with defining polynomial 2*x^2 -
    3 over ZZ
> Basis(O);
[
    O.1,
    O.2
]
> [K!x : x in Basis(O)];
[
```

```
    1,
    2*a       // this is Sqrt(3)
]
```

Here's are some more examples:

```
> procedure ints(f)   // (procedures don't return anything; functions do)
      K<a> := NumberField(f);
      O := MaximalOrder(K);
      print [K!z : z in Basis(O)];
  end procedure;
> ints(x^2-5);
[
    1,
    1/2*(a + 1)
]
> ints(x^2+5);
[
    1,
    a
]
> ints(x^3-17);
[
    1,
    a,
    1/3*(a^2 + 2*a + 1)
]
> ints(CyclotomicPolynomial(7));
[
    1,
    a,
    a^2,
    a^3,
    a^4,
    a^5
]
> ints(x^5+&+[Random(10)*x^i : i in [0..4]]);  // RANDOM
[
    1,
    a,
    a^2,
    a^3,
    a^4
]
> ints(x^5+&+[Random(10)*x^i : i in [0..4]]);  // RANDOM
```

```
[
    1,
    a,
    a^2,
    1/2*(a^3 + a),
    1/16*(a^4 + 7*a^3 + 11*a^2 + 7*a + 14)
]
```

Lets find out how high of a degree MAGMA can easily deal with.

```
> d := 10; time ints(x^10+&+[Random(10)*x^i : i in [0..d-1]]);
[
    1, a, a^2, a^3, a^4, a^5, a^6, a^7, a^8, a^9
]
Time: 0.030
> d := 15; time ints(x^10+&+[Random(10)*x^i : i in [0..d-1]]);
[
    1,
    7*a,
    7*a^2 + 4*a,
    7*a^3 + 4*a^2 + 4*a,
    7*a^4 + 4*a^3 + 4*a^2 + a,
    7*a^5 + 4*a^4 + 4*a^3 + a^2 + a,
    7*a^6 + 4*a^5 + 4*a^4 + a^3 + a^2 + 4*a,
    7*a^7 + 4*a^6 + 4*a^5 + a^4 + a^3 + 4*a^2,
    7*a^8 + 4*a^7 + 4*a^6 + a^5 + a^4 + 4*a^3 + 4*a,
    7*a^9 + 4*a^8 + 4*a^7 + a^6 + a^5 + 4*a^4 + 4*a^2 + 5*a,
    7*a^10 + 4*a^9 + 4*a^8 + a^7 + a^6 + 4*a^5 + 4*a^3 + 5*a^2 +  4*a,
  ...
]
Time: 0.480
> d := 20; time ints(x^10+&+[Random(10)*x^i : i in [0..d-1]]);
[
    1,
    2*a,
    4*a^2,
    8*a^3,
    8*a^4 + 2*a^2 + a,
    8*a^5 + 2*a^3 + 3*a^2,
 ...]
Time: 3.940
> d := 25; time ints(x^10+&+[Random(10)*x^i : i in [0..d-1]]);
... I stopped it after a few minutes...
```

We can also define orders in rings of integers.

```
> R<x> := PolynomialRing(RationalField());
> K<a> := NumberField(x^3-2);
> O := Order([2*a]);
> O;
Transformation of Order over
Equation Order with defining polynomial x^3 - 2 over ZZ
Transformation Matrix:
[1 0 0]
[0 2 0]
[0 0 4]
> OK := MaximalOrder(K);
> Index(OK,O);
8
> Discriminant(O);
-6912
> Discriminant(OK);
-108
> 6912/108;
64    // perfect square...
```

3.4 Ideals

```
> R<x> := PolynomialRing(RationalField());
> K<a> := NumberField(x^3-2);
> O := Order([2*a]);
> O;
Transformation of Order over
Equation Order with defining polynomial x^3 - 2 over ZZ
Transformation Matrix:
[1 0 0]
[0 2 0]
[0 0 4]
> OK := MaximalOrder(K);
> Index(OK,O);
8
> Discriminant(O);
-6912
> Discriminant(OK);
-108
> 6912/108;
64    // perfect square...
> R<x> := PolynomialRing(RationalField());
> K<a> := NumberField(x^2-7);
> K<a> := NumberField(x^2-5);
```

```
> Discriminant(K);
20   // ????????? Yuck!
> OK := MaximalOrder(K);
> Discriminant(OK);
5    // better
> Discriminant(NumberField(x^2-20));
80
> I := 7*OK;
> I;
Principal Ideal of OK
Generator:
    [7, 0]
> J := (OK!a)*OK;    // the ! computes the natural image of a in OK
> J;
Principal Ideal of OK
Generator:
    [-1, 2]
> I*J;
Principal Ideal of OK
Generator:
    [-7, 14]
> J*I;
Principal Ideal of OK
Generator:
    [-7, 14]
> I+J;
Principal Ideal of OK
Generator:
    [1, 0]
>
> Factorization(I);
[
    <Principal Prime Ideal of OK
    Generator:
        [7, 0], 1>
]
> Factorization(3*OK);
[
    <Principal Prime Ideal of OK
    Generator:
        [3, 0], 1>
]
> Factorization(5*OK);
[
```

```
    <Prime Ideal of OK
    Two element generators:
        [5, 0]
        [4, 2], 2>
]
> Factorization(11*OK);
[
    <Prime Ideal of OK
    Two element generators:
        [11, 0]
        [14, 2], 1>,
    <Prime Ideal of OK
    Two element generators:
        [11, 0]
        [17, 2], 1>
]
```

We can even work with fractional ideals in MAGMA.

```
> K<a> := NumberField(x^2-5);
> OK := MaximalOrder(K);
> I := 7*OK;
> J := (OK!a)*OK;
> M := I/J;
> M;
Fractional Principal Ideal of OK
Generator:
    -7/5*OK.1 + 14/5*OK.2
> Factorization(M);
[
    <Prime Ideal of OK
    Two element generators:
        [5, 0]
        [4, 2], -1>,
    <Principal Prime Ideal of OK
    Generator:
        [7, 0], 1>
]
```

3.5 Next time

On Tuesday I will talk about discriminants and describe an algorithm for "factoring primes", that is writing an ideal $p\mathcal{O}_K$ as a product of prime ideals of $\mathcal{O}_K$.