BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 26, Number 2, April 1992

ALGORITHMS IN ALGEBRAIC NUMBER THEORY

H. W. LENSTRA, JR.

ABSTRACT. In this paper we discuss the basic problems of algorithmic algebraic number theory. The emphasis is on aspects that are of interest from a purely mathematical point of view, and practical issues are largely disregarded. We describe what has been done and, more importantly, what remains to be done in the area. We hope to show that the study of algorithms not only increases our understanding of algebraic number fields but also stimulates our curiosity about them. The discussion is concentrated of three topics: the determination of Galois groups, the determination of the ring of integers of an algebraic number field, and the computation of the group of units and the class group of that ring of integers.

1. INTRODUCTION

The main interest of algorithms in algebraic number theory is that they provide number theorists with a means of satisfying their professional curiosity. The praise of numerical experimentation in number theoretic research is as widely sung as purely numerological investigations are indulged in, and for both activities good algorithms are indispensable. What makes an algorithm *good* unfortunately defies definition—too many extra-mathematical factors affect its practical performance, such as the skill of the person responsible for its execution and the characteristics of the machine that may be used.

The present paper addresses itself not to the researcher who is looking for a collection of well-tested computational methods for use on his recently acquired personal computer. Rather, the intended reader is the perhaps imaginary pure mathematician who feels that he makes the most of his talents by staying away from computing equipment. It will be argued that even from this perspective the study of algorithms, when considered as objects of research rather than as tools, offers rich rewards of a theoretical nature.

The problems in pure mathematics that arise in connection with algorithms have all the virtues of good problems. They are of such a distinctly fundamental nature that one is often surprised to discover that they have not been considered earlier, which happens even in well-trodden areas of mathematics; and even in areas that are believed to be well-understood it occurs frequently that the existing theory offers no ready solutions, fundamental though the problems may be. Solutions that have been found often need tools that at first sight seem foreign to the statement of the problem.

Received by the editors October 11, 1991.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 11Y16, 11Y40.
Key words and phrases. Algebraic number theory, algorithms, complexity theory.
The author was supported by NSF under Grant No. DMS 90-02939.
This paper was given as a Progress in Mathematics Lecture at the August 8–10, 1991 meeting of the American Mathematical Society in Orono, Maine.

©1992 American Mathematical Society
0273-0979/92 $1.00 + $ 25 per page

Algebraic number theory has in recent times been applied to the solution of algorithmic problems that, in their formulations, do not refer to algebraic number theory at all. That this occurs in the context of solving diophantine equations (see, e.g., [72]) does not come as a surprise, since these lie at the very roots of algebraic number theory. A better example is furnished by the seemingly elementary problem of decomposing integers into prime factors. Among the ingredients that make modern primality tests work one may mention reciprocity laws in cyclotomic fields (see [3, 25, 24]), arithmetic in cyclic fields (see [46, 10]), the construction of Hilbert class fields of imaginary quadratic fields [5], and class number estimates of fourth degree CM-fields [1]. The best rigorously proved time bound for integer factorization is achieved by an algorithm that depends on quadratic fields (see [49]), and the currently most promising practical approach to the same problem, the *number field sieve* (see [17, 43, 44]), employs "random" number fields of which the discriminants are so huge that many traditional computational methods become totally inapplicable. The analysis of many algorithms related to algebraic number fields seriously challenges our theoretical understanding, and one is often forced to argue on the basis of heuristic assumptions that are formulated for the occasion. It is considered a relief when one runs into a standard conjecture such as the generalized Riemann hypothesis (as in [6, 15]) or Leopoldt's conjecture on the nonvanishing of the p-adic regulator [60].

In this paper we will consider algorithms in algebraic number theory for their own sake rather than with a view to any of the above applications. The discussion will be concentrated on three basic algorithmic questions that one may ask about algebraic number fields, namely, how to determine the Galois group of the normal closure of the field, or, more generally, of any polynomial over any algebraic number field; how to find the ring of integers of the field; and how to determine the unit group and the ideal class group of that ring of integers. These are precisely the subjects that are discussed in *Algorithmic algebraic number theory* by M. Pohst and H. Zassenhaus (Cambridge, 1989), but our point of view is completely different. Pohst and Zassenhaus present algorithms that "yield good to excellent results for number fields of small degree and not too large discriminant" [56, Preface], but our attitude will be decidedly and exclusively asymptotic. For the purposes of the present paper one algorithm is considered better than another if, for each positive real number N, it is at least N times as fast for all but finitely many values of the input data. It is clear that with this attitude we can make no claims concerning the practical applicability of any of the results that are achieved. In fact, following Archimedes [4] one should be able, on the basis of current physical knowledge, to find an upper estimate for all sets of numerical input data to which any algorithm will ever be applied, and an algorithm that is faster in all those finitely many instances may still be worse in our sense.

To some people the above attitude may seem absurd. To the intended reader, who is never going to apply any algorithm anyway, it comes as a liberation and a relief. Once he explicitly gives up all practical claims he will realize that he can occupy himself with algorithms without having to fear the bad dreams caused by the messy details and dirty tricks that stand between an elegant algorithmic idea and its practical implementation. He will find himself in the platonic paradise of pure mathematics, where a conceptual and concise version of an algorithm

is valued more highly than an ad hoc device that speeds it up by a factor of ten and where words have precise meanings that do not change with the changing world. He will never need to enter the dark factories that in his imagination are populated by applied mathematicians, where boxes full of numbers that they call matrices are carried around and where true electronic computers are fed with proliferating triple indices. And in his innermost self he will know that in the end his own work will turn out to have the widest application range, exactly because it was not done with any specific application in mind.

There is a small price to be paid for admission to this paradise. Algorithms and their running times can only be investigated mathematically if they are given exact definitions, and this can apparently be done only if one employs the terminology of *theoretical computer science*, which our intended reader unfortunately does not feel comfortable with either. It is only out of respect for his feelings that I have not called this paper *Complexity of algorithms in algebraic number theory*, which would have described its contents more accurately.

Although it is, from a rigorous mathematical point of view, desirable that I define what I mean by an algorithm and its running time, I will not do so. My main excuse is that I do not know these definitions myself. Even worse, I never saw a treatment of the appropriate theory that is precise, elegant, and convenient to work with. It would be a laudable enterprise to fill this apparent gap in the literature. In the meantime, I am happy to show by example that one can avoid paying the admission price, just as not all algebraists are experts on set theory or algebraic geometers on category theory. The intuitive understanding that one has of algorithms and running times, or of sets and categories, is amply sufficient. Exact definitions appear to be necessary only when one wishes to prove that algorithms with certain properties do *not* exist, and theoretical computer science is notoriously lacking in such negative results. The reader who wishes to provide his own definitions may wish to consult [74] for an account of the pitfalls to be avoided. He should bear in mind that all theorems in the present paper should become formal consequences of his definitions, which makes his task particularly academic.

My intended reader may have another allergy, namely, for *constructive mathematics*, in which purely existential proofs and the law of the excluded middle are not accepted. This has only a superficial relationship to algorithmic mathematics. Of course, it often happens that one can obtain a good algorithm by just transcribing an essentially constructive proof, but such algorithms do not tend to be the most interesting ones; many of them are mentioned in §2. In the design and analysis of algorithms one gladly invokes all the help that existing pure mathematics has to offer and often some not-yet-existing mathematics as well.

For an account of algorithms in algebraic number theory that emphasizes the practical aspects rather than complexity issues we refer to the forthcoming book by Cohen [23].

In §2 we cover the basic terminology and the basic auxiliary results to be used in later sections. In particular, we discuss several fundamental questions that, unlike integer factorization, admit a satisfactory algorithmic treatment. These include questions related to finitely generated abelian groups, to finite fields, and to the factorization of polynomials over number fields.

Section 3 is devoted to the problem of determining Galois groups. We review

the little that has been done on the complexity of this problem, including the pretty result of Landau and Miller [36] that solvability by radicals can be decided efficiently. We also point out several directions for further research.

In §4 we discuss the problem of determining the ring of integers of a given algebraic number field. The main result is a negative one—the problem is in many ways equivalent to the problem of finding the largest square factor of a given positive integer, which is intractable at present. Nevertheless, we will see that one can get quite close. There is an interesting connection with the resolution of plane curve singularities that remains to be exploited.

Section 5 considers the problem of determining the unit group $\mathscr{O}^*$ and the ideal class group $\mathrm{Cl}\,\mathscr{O}$ of the ring of integers $\mathscr{O}$ of a given number field. Showing that these are effectively computable is not entirely trivial, and since most textbooks are silent on this point, I treat it in some detail. We shall see that the existing complexity estimates for this problem still leave room for improvement, and what we have to say is far from conclusive. In §6 we prove a few explicit bounds concerning units and class groups that are needed in §5. Several results in these two sections could have been formulated in terms of the divisor class group $\mathrm{Pic}_f\,\mathscr{O}$ that appears in Arakelov theory (see [70, §I]) and that already appeared in the context of algorithms (see [65, 45]). Knowing the group $\mathrm{Pic}_c\,\mathscr{O}$ is equivalent to knowing both $\mathscr{O}^*$ and $\mathrm{Cl}\,\mathscr{O}$, which may explain why algorithms for computing $\mathscr{O}^*$ and algorithms for computing $\mathrm{Cl}\,\mathscr{O}$ are often inextricably linked. It also explains why, contrary to many authors in the field, I prefer to think of determining $\mathscr{O}^*$ and determining $\mathrm{Cl}\,\mathscr{O}$ as a single problem.

The three basic questions that are addressed in this *progress report* still offer ample opportunities for additional progress. Among the many other algorithmic questions in algebraic number theory that merit attention we mention the problem of tabulating number fields, problems from class field theory such as the calculation of Artin symbols, problems concerning quadratic forms, and the analogues of all problems discussed in this paper for function fields of curves over finite fields.

2. Preliminaries

2.1. **Algorithms and complexity.** It is assumed that the reader has an intuitive understanding of the notion of an *algorithm* as being a recipe that given one finite sequence of nonnegative integers called the *input* data, produces another, called the *output*. Formally, an algorithm may be defined as a *Turing machine*, but for several of our results it is better to choose as our "machine model" an idealized computer that is more realistic with respect to its *running time*, which is another intuitively clear notion that we do not define. We refer to [74] and the literature given there for a further discussion of these points.

The *length* of a finite sequence of nonnegative integers $n_1, n_2, \ldots, n_t$ is defined to be $\sum_{i=1}^{t} \log(n_i+2)$. It must informally be thought of as proportional to the number of bits needed to spell out the n_i in binary. By analyzing the *complexity* of an algorithm we mean in this paper finding a reasonably sharp upper bound for the running time of the algorithm expressed as a function of the length of the input data. This should, more precisely, be called *time* complexity, to distinguish it from *space* complexity. An algorithm is said to be *polynomial-time* or *good* if its running time is $(l+2)^{O(1)}$, where l is the length of the input. Studying the complexity of a *problem* means finding an algorithm

for that problem of the smallest possible complexity In the present paper we consider the complexity analysis complete when a good algorithm for a problem has been found, and we will not be interested in the value of the O-constant Informally, a problem has a good algorithm if an instance of the problem is almost as easily *solved* as it is *formulated*

Sometimes we will refer to a *probabilistic* algorithm, which is an algorithm that may use a random number generator for drawing random bits One formalization of this is a *nondeterministic* Turing machine (see [74]) Unless we use the word *probabilistic*, we do *not* allow the use of random number generators, and if we wish to emphasize this we talk of *deterministic* algorithms In the case of a probabilistic algorithm, the running time and the output are not determined by the input alone, but both have, for each fixed value of the input data, a *distribution* The *expected* running time of a probabilistic algorithm is the expectation of the running time for a given input Studying the complexity of a probabilistic algorithm means finding an upper bound for the expected running time as a function of the length of the input For a few convenient rules that can be used for this purpose we refer to [49, §12] A probabilistic algorithm is called *good* if its expected running time is $(l+2)^{O(1)}$, where l is the length of the input

Parallel algorithms have not yet played any role in algorithmic number theory, and they will not be considered here

Many results in this paper assert that "there exists" an algorithm with certain properties In all cases, such an algorithm can actually be exhibited, at least in principle

All O-constants are absolute and effectively computable unless indicated otherwise

2 2 **Encoding data.** As stated above, the input and the output of an algorithm consist of finite sequences of nonnegative integers However, in the mathematical practice of thinking and writing about algorithms one prefers to work with mathematical concepts rather than with sequences of nonnegative integers that encode them in some manner Thus, one likes to say that the input of an algorithm is given by an algebraic number field rather than by the sequence of coefficients of a polynomial that defines the field, and it is both shorter and clearer to say that one computes the kernel of a certain endomorphism of a vector space than that one determines a matrix of which the columns express a basis for that kernel in terms of a given basis of the vector space To justify such a concise mode of expression we have to agree on a way of encoding entities such as number fields, vector spaces, and maps between them by means of finite sequences of nonnegative integers That is one of the purposes of the remainder of this section Sometimes there is one obvious way to do the encoding, but often there are several, in which case the question arises whether there is a good algorithm that passes from one encoding to another When there is, we will usually not distinguish between the encodings, although for practical purposes they need not be equivalent

We shall see that the subject of encoding mathematical entities suggests several basic questions, but we will not pursue these systematically We shall not do much more than what will be needed in later sections

2 3 **Elementary arithmetic.** By $\mathbb{Z}$ we denote the ring of integers Adding a

sign bit we can clearly use nonnegative integers to represent *all* integers. The traditional algorithms for addition and subtraction take time $O(l)$, where l is the length of the input. The ordinary algorithms for multiplication and division with remainder, as well as the Euclidean algorithm for the computation of greatest common divisors, have running time $O(l^2)$. With the help of more sophisticated methods this can be improved to $l^{1+o(1)}$ for $l \to \infty$ (see [33]) An operation that is *not* known to be doable by means of a good algorithm is decomposing a positive integer into prime numbers (see [33, 50, 41]), but there is a good probabilistic algorithm for the related problem of deciding whether a given integer is prime [1]. No good algorithms are known for the problem of recognizing squarefree numbers and the problem of finding the largest square dividing a given positive integer, even when the word "good" is given a less formal meaning (see [43, §2]).

For some algorithms a prime number p is part of the input. In such a case, the prime is assumed to be encoded by itself rather than that, for example, n stands for the nth prime. Since we know no good deterministic algorithm for recognizing primes, it is natural to ask what the algorithm does if p is not prime or at least not known to be prime Some algorithms may discover that p is nonprime, either because a known property of primes is contradicted in the course of the computations, or because the algorithm spends more time than it should, such algorithms may be helpful as primality tests. Other algorithms may even give a nontrivial factor of p, which may make them applicable as integer factoring algorithms. For both types of algorithms, one can ask what can be deduced if the algorithm does appear to terminate successfully Does this assist us in proving that p is prime? What do we know about the output when we do not assume that p is prime? An algorithm for which this question has not been answered satisfactorily is Schoof's algorithm for counting the number of points on an elliptic curve over a finite field [62].

Rational numbers can be represented as pairs of integers in an obvious manner, and all field operations can be performed on them in polynomial time.

Let n be a positive integer. The elements of the ring $\mathbf{Z}/n\mathbf{Z}$ are assumed to be encoded as nonnegative integers less than n The ring operations can be performed in polynomial time. An *ideal* $I \subset \mathbf{Z}/n\mathbf{Z}$ can be encoded either by means of its index $d = [\mathbf{Z}/n\mathbf{Z} : I]$, which completely determines it and which can be any divisor of n, or by means of a finite sequence of elements that generates I, or by means of a single generator. An element of I can be represented either as an element of $\mathbf{Z}/n\mathbf{Z}$ that is divisible by d, or as an explicit $\mathbf{Z}/n\mathbf{Z}$-linear combination of the given generators of I, or as an explicit multiple of a single given generator. Using the extended Euclidean algorithm one easily sees that one can pass from any of these encodings of ideals and their elements to any other in polynomial time and that one can likewise test inclusion and equality of given ideals In particular, one can decide in polynomial time whether a given nonzero element of $\mathbf{Z}/n\mathbf{Z}$ is a unit, if so find its inverse, and if not so find a nontrivial divisor of n Taking $n = p$ to be prime we conclude that we can perform all field operations in $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ in polynomial time.

2 4. **Linear algebra.** Let F be a field, and suppose that one has agreed upon an encoding of its elements, as is the case when F is the field $\mathbf{Q}$ of rational numbers or the field $\mathbf{F}_p$ for some prime number p (see 2 3). Giving a finite-

dimensional vector space over F simply means giving a nonnegative integer n, which is the dimension of the vector space. This number n is to be given in *unary*, i.e., as a sequence $1, 1, \ldots, 1$ of n ones, so that the length of the encoding is at least n. This is because almost any algorithm related to a vector space of dimension n takes time at least n. The elements of such a vector space are encoded as sequences of n elements of F. Homomorphisms between vector spaces are encoded as matrices. A subspace of a vector space can be encoded as a sequence of elements that spans the subspace, or as a sequence of elements that forms a basis of the subspace, or as the kernel of a homomorphism from the vector space to another one. For all fields F that we shall consider the traditional algorithms from linear algebra, which are based on Gaussian elimination, are polynomial-time: algorithms that pass back and forth between different representations of subspaces, algorithms that decide inclusion and equality of subspaces, that form sums and intersections of subspaces, algorithms that construct quotient spaces, direct sums, and tensor products, algorithms for computing determinants and characteristic polynomials of endomorphisms, and algorithms that decide whether a given homomorphism is invertible and if so construct its inverse. The proofs are straightforward, the main problem being to find upper bounds for the sizes of the numbers that occur in the computations, for example when $F = \mathbf{Q}$.

If one applies any of these algorithms to $F = \mathbf{Z}/p\mathbf{Z}$ without knowing that p is prime, then one either finds a nontrivial divisor of p because some division by a nonzero element fails, or the algorithm performs successfully as if F were a field. In the latter case it is usually easy to interpret the output of the algorithm in terms of free $\mathbf{Z}/p\mathbf{Z}$-modules (see [14]), thus avoiding the assumption that p be prime.

2.5. **Finitely generated abelian groups.** Specifying a finitely generated abelian group is done by giving a sequence of nonnegative integers $d_1, d_2, \ldots, d_t$; the group is then $\bigoplus_{i=1}^{t} \mathbf{Z}/d_i\mathbf{Z}$, which enables us to represent the elements of the group by means of sequences of t integers. In our applications the group is usually either finite (all $d_i > 0$) or free abelian (all $d_i = 0$). To make the d_i unique one may require that d_i divides d_{i+1} for $1 \leq i < t$; this can be accomplished in polynomial time. One should not require the d_i to be *prime powers*, since that is, for all we know, algorithmically hard to achieve. Starting from this description of finitely generated abelian groups, one can encode maps and subgroups in various ways that are reminiscent of 2.4 and that are left to the imagination of the reader. He may also formulate the analogues of the problems mentioned in 2.4 for the current case and construct good algorithms for them using Hermite and Smith reduction of integer matrices (see [29]). The main difficulty is to keep the intermediate numbers small.

2.6. **Basis reduction.** In many cases a finitely generated free abelian group L is equipped with a bilinear symmetric map $L \times L \to \mathbf{R}$ that induces a Euclidean structure on $L_{\mathbf{R}} = L \otimes_{\mathbf{Z}} \mathbf{R}$; here $\mathbf{R}$ denotes the field of real numbers. For example, this is the case if L is a subgroup of $\mathbf{Z}^n$, with the ordinary inner product. It is also the case if L is a finitely generated subgroup of the additive group of an algebraic number field K (see 2.9), the bilinear symmetric map in this case being induced by $(x, x) = \sum_\sigma |\sigma x|^2$, where σ ranges over the field homomorphisms from K to the field $\mathbf{C}$ of complex numbers. In such cases it

is often desirable to find a *reduced basis* of L over $\mathbf{Z}$, i e, a basis of which the elements are "short" in a certain sense If the symmetric matrix that defines the bilinear map on a given basis of L is known to a certain accuracy, then a reduced basis can be found by means of a *reduction algorithm* The complexity of such an algorithm depends on the precise notion of "reduced basis" that one employs In [42] one finds a good reduction algorithm that will suffice for our purposes. See [30] for further developments

2 7. **Rings.** We use the convention that rings have unit elements, that a subring has the same unit element, and that ring homomorphisms preserve the unit element. The *characteristic* $\operatorname{char} A$ of a ring A is the nonnegative integer that generates the kernel of the unique ring homomorphism $\mathbf{Z} \to A$ The group of units of a ring A is denoted by A^*. All rings in this paper are supposed to be *commutative.*

Almost any ring that we need to encode in this paper has an additive group that is either finitely generated or a finite-dimensional vector space over $\mathbf{Q}$, for exceptions, see 2.11. Such a ring A is encoded by giving its underlying abelian group as in 2.5 or 2.4 together with the multiplication map $A \otimes A \to A$. It is straightforward to decide in polynomial time whether the multiplication map satisfies the ring axioms.

Ideals are encoded as subgroups or, equivalently, as kernels of ring homomorphisms. There are good algorithms for computing the sum, product, and intersection of ideals, as well as the ideal $I \cdot J = \{x \in A: xJ \subset I\}$ for given I and J, and the quotient ring of A modulo a given ideal.

A *polynomial* over a ring is always supposed to be given by means of a complete list of its coefficients, including the zero coefficients; thus we do not work with sparse polynomials of a very high degree

Most *finite rings* that have been encountered in algorithmic number theory "try to be fields" in the sense that one is actually happy to find a zero-divisor in the ring. This applies to the way they occur in §4 and also to the application of finite rings in primality testing [46, 10]. Nevertheless, it seems of interest to study finite rings from an algorithmic point of view for their own sake Testing whether a given finite ring is local can be done by a good probabilistic algorithm, but finding the localizations looks very difficult Testing whether it is reduced or a principal ideal ring also looks very difficult, but there may be a good algorithm for deciding whether it is quasi-Frobenius. I do not know whether isomorphism can be tested in polynomial time. Many difficulties are already encountered for finite rings that are $\mathbf{F}_p$-algebras for some prime number p. Two finite etale $\mathbf{F}_p$-algebras can be tested for isomorphism in polynomial time (cf. [14]), but there is no known good deterministic algorithm for finding the isomorphism if it exists, if they are fields, there is, but the proof depends on ring theory (see [48]).

2.8. **Finite fields.** Let p be a prime number, n a positive integer, and $q = p^n$ A finite field $\mathbf{F}_q$ of cardinality q is encoded as a ring, as in 2 7. This comes down to specifying p, n, as well as a system of n^3 elements a_{ijk} of $\mathbf{F}_p$ with the property that there is a basis $e_1, e_2, \ldots, e_n$ of $\mathbf{F}_q$ over $\mathbf{F}_p$ such that $e_i e_j = \sum_k a_{ijk} e_k$ for all i, j We refer to [48] for a description of good algorithms for various fundamental problems: performing the field operations in a given finite field, as well as exponentiation and the application of automorphisms, finding

all subfields of a given finite field $\mathbf{F}_q$, finding the irreducible polynomial of a given element of $\mathbf{F}_q$ over a given subfield, finding a primitive element of $\mathbf{F}_q$, i.e., an element $\alpha \in \mathbf{F}_q$ with $\mathbf{F}_q = \mathbf{F}_p(\alpha)$, finding a normal basis of $\mathbf{F}_q$ over a given subfield, and finding all field homomorphisms and isomorphisms from a given finite field to another. Most of these algorithms rely heavily on linear algebra.

Given a positive integer p and a system of n^3 elements a_{ijk} of $\mathbf{Z}/p\mathbf{Z}$, how does one decide whether they specify a field $\mathbf{F}_q$ as above? This is at least as hard as testing p for primality, for which no good deterministic algorithm is known. However, this is the *only* obstruction: there is a good algorithm that given p and the a_{ijk} *either* shows that they do *not* define a field, *or* shows that if p is prime they do. Namely, one runs the algorithms mentioned above for finding a primitive element α and its minimal polynomial f over $\mathbf{Z}/p\mathbf{Z}$, just as if one is working with a field, and one verifies that the map sending X to α induces an isomorphism from $(\mathbf{Z}/p\mathbf{Z})[X]/(f)$ to the structure that one is working with; if this is not true, or if anything went wrong during the course of the algorithm, one does not have a field; if it is, then as a final test one decides whether f is irreducible over $\mathbf{Z}/p\mathbf{Z}$, which for prime p can be done by means of a good algorithm (see [38, 47] and the references given there).

There are also problems for which no good algorithm is known. One is the problem of *constructing* $\mathbf{F}_{p^n}$ for a given prime p and a given positive integer n, or, equivalently, constructing an irreducible polynomial $f \in \mathbf{F}_p[X]$ of degree n; here n is supposed to be given in unary (cf. 2.4). If one accepts the generalized Riemann hypothesis then there is a good algorithm for doing this [2]. There is also a good *probabilistic* algorithm for this problem, and a deterministic algorithm that runs in $\sqrt{p}$ times polynomial time [66].

An important problem, which will come up several times in this paper, is the problem of factoring a given polynomial f in one variable over a given finite field $\mathbf{F}_{p^n}$. No good algorithm is known for this problem, even when the generalized Riemann hypothesis is assumed. There does exist a good probabilistic algorithm and a deterministic algorithm that runs in $\sqrt{p}$ times polynomial time [67]; if p is fixed, or smaller than the degree of f, then the latter algorithm is good. There also exists a good algorithm that, given $f \in \mathbf{F}_{p^n}[X]$, determines the *factorization type* of f, i.e., the number of irreducible factors and their degrees and multiplicities. We refer to [47] for a further discussion.

Algorithmic problems relating to the multiplicative group of finite fields, such as the discrete logarithm problem, are generally very difficult, see [53, 57, 41, 27, 60, 51].

2.9. **Number fields.** By a *number field* or an *algebraic number field* we mean in this paper a field extension K of finite degree of the field $\mathbf{Q}$ of rational numbers. For the basic theory of algebraic number fields, see [37, 75, 20].

An algebraic number field K is encoded as its underlying $\mathbf{Q}$-vector space together with the multiplication map $K \otimes_{\mathbf{Q}} K \to K$, as in 2.7; in other words, giving K amounts to giving a positive integer n and a system of n^3 rational numbers a_{ijk} that describe the multiplication in K on a vector space basis of K over $\mathbf{Q}$ (cf. 2.8 above). As in [48, §2], one shows that the field operations in a number field can be performed in polynomial time. Using standard arguments from field theory one shows that there are good algorithms for determining

the irreducible polynomial of a given element of K over a given subfield and for finding a primitive element of K, i.e., an element $\alpha \in K$ for which $K = \mathbf{Q}(\alpha)$. It follows that giving a number field is equivalent to giving an irreducible polynomial $f \in \mathbf{Q}[X]$ and letting the field be $\mathbf{Q}[X]/f\mathbf{Q}[X]$.

Polynomials in one variable with coefficients in an algebraic number field can be factored into irreducible factors in polynomial time. This is done with the help of basis reduction, see [42, 35, 39, 40]. We note two consequences.

First of all, from the argument given in 2.8 one sees that there is a good algorithm for deciding whether a given system of n^3 rational numbers defines a number field. Secondly, given *two* number fields $K = \mathbf{Q}(\alpha)$ and K', one can decide whether or not they are isomorphic, and if so, find all isomorphisms, in polynomial time. To do this, one factors the irreducible polynomial f of α over $\mathbf{Q}$ into irreducible factors in the ring $K'[X]$, and one observes that the *linear* factors are in bijective correspondence with the field homomorphisms $K \to K'$; such a field homomorphism is an isomorphism if and only if the two fields have the same degree over $\mathbf{Q}$.

With $K = K'$ we see from the above that one can also determine all automorphisms of K, and composing them one can make a complete multiplication table for the group $\mathrm{Aut}\, K$ of field automorphisms of K, all in polynomial time.

In the proof of 3.5 we shall see that all maximal proper subfields of a given number field of degree n can be found in polynomial time. Finding *all* subfields is asking too much, since the number of subfields is not polynomially bounded. I do not know whether all *minimal* subfields different from $\mathbf{Q}$ can be found in polynomial time, nor whether their number is $n^{O(1)}$. Intersections and composites of given subfields can be found by means of linear algebra.

We stress that for our algorithms the number field K is considered to be *variable* rather than fixed, and that we wish our running time estimates to be uniform in K.

2.10. **Orders.** An *order* in a number field K of degree n is a subring A of K of which the additive group is isomorphic to $\mathbf{Z}^n$. Among all orders in K there is a unique maximal one, which is called the *ring of integers* of K and denoted by $\mathscr{O}$. The orders in K are precisely the subrings of $\mathscr{O}$ of finite additive index. The *discriminant* Δ_A of an order A with $\mathbf{Z}$-basis $\omega_1, \omega_2, \ldots, \omega_n$ is the determinant of the matrix $(\mathrm{Tr}(\omega_i\omega_j))_{i,j}$, where $\mathrm{Tr}\colon K \to \mathbf{Q}$ is the trace map. The discriminant of every order is a nonzero integer. The discriminant of $\mathscr{O}$ is also called the discriminant of K over $\mathbf{Q}$ and is simply denoted by Δ.

There are several ways of encoding an order A in a number field K. One is by specifying A as a ring as in 2.7, which amounts to giving n and a system of n^3 integers a_{ijk}; from $A \otimes_{\mathbf{Z}} \mathbf{Q} \cong K$ it follows that the same data also encode K. Another is by specifying K as well as a sequence of elements of K that generates A as a ring, or as an abelian group. We leave it to the reader to check that there are good algorithms for transforming all these encodings into each other.

Given a number field K one can construct an order in K in polynomial time, as follows. Let n^3 rational numbers a_{ijk} be given that describe the multiplication on a $\mathbf{Q}$-basis $e_1 = 1, e_2, \ldots, e_n$ for K, and let d be the least common multiple of the denominators of the a_{ijk}. Then $A = \mathbf{Z} + \sum_{i=2}^{n} \mathbf{Z}de_i$

is an order in K. In many cases one knows the irreducible polynomial f of a primitive element α of K over $\mathbf{Q}$. If $f \in \mathbf{Z}[X]$, then one can take for A the "equation order" $\mathbf{Z}[\alpha]$, which as a ring is isomorphic to $\mathbf{Z}[X]/f\mathbf{Z}[X]$. If f does not belong to $\mathbf{Z}[X]$, then one can either replace α by $m\alpha$ for a suitable positive integer m, or use a little known generalization of the equation order, namely, the ring

$$A = \left\{ \beta \in K \quad \beta \cdot \sum_{i=0}^{n-1} \mathbf{Z}\alpha^i \subset \sum_{i=0}^{n-1} \mathbf{Z}\alpha^i \right\}$$

To find a $\mathbf{Z}$-basis for this ring, let m be the least positive integer for which the polynomial $g = mf = \sum_{i=0}^{n} a_i X^i$ has coefficients a_i in $\mathbf{Z}$ (with $a_n = m$), then

$$A = \mathbf{Z} + \sum_{i=1}^{n-1} \mathbf{Z} \cdot \left(\sum_{j=0}^{i-1} a_{n-j} \alpha^{i-j} \right)$$

These are exactly the rings A for which $\operatorname{Spec} A$ is isomorphic to a "horizontal" prime divisor of the projective line over $\mathbf{Z}$. Many results that are known for equation orders have direct analogues for rings of this type, for example, the discriminant of A equals the discriminant of g.

Applying basis reduction to a given order A as in 2.6, one can find a $\mathbf{Z}$-basis for A with the property that the integers a_{ijk} that express multiplication in this basis satisfy $a_{ijk} = |\Delta_A|^{O(n)}$. This shows that A can be encoded by means of data of length $O(n^4(2 + \log|\Delta_A|))$, and that there is a good algorithm for transforming a given encoding into one satisfying this bound. From the inequality $n \leq 2(\log|\Delta_A|)/\log 3$, which is valid for all $A \neq \mathbf{Z}$, one sees that the bound is $(2 + \log|\Delta_A|)^{O(1)}$. It is often convenient to assume that the given encoding of A satisfies this bound, and to estimate running times in terms of $|\Delta_A|$.

Let A be an order in a number field K of degree n. By a *fractional ideal* of A we mean a finitely generated nonzero A-submodule of K. The additive group of a fractional ideal is isomorphic to $\mathbf{Z}^n$. One can compute with fractional ideals as with ideals (see 2.7).

2.11 **Local fields.** A *local field* is a locally compact, nondiscrete topological field. Such a field is topologically isomorphic to the field $\mathbf{R}$ of real numbers, or to the field $\mathbf{C}$ of complex numbers, or, for some prime number p, to a finite extension of the field $\mathbf{Q}_p$ of p-adic numbers, or, for some finite field E, to the field $E((t))$ of formal Laurent series over E. A local field is uncountable, which implies that we have to be satisfied with specifying its elements only to a certain precision. The discussion below is limited to the case that the field is non-archimedean, i.e. not isomorphic to $\mathbf{R}$ or $\mathbf{C}$.

The complexity theory of local fields has not been developed as systematically as one might expect on the basis of their importance in number theory (see [19]). The first thing to do is to develop algorithms for factoring polynomials in one variable to a given precision, see [21, 14] and §4 below. Here the incomplete solution of the corresponding problem over finite fields (see 2.8) causes a difficulty, we are forced to admit probabilistic algorithms, or to allow the running time to be $\sqrt{p}$ times polynomial time, where p denotes the characteristic

of the residue class field, or to avoid the need for *completely* factoring polynomials Once one can factor polynomials, it is likely that satisfactory algorithms can be developed for the calculation of ramification indices and residue class field degrees of finite extensions of non-archimedean local fields Some further problems are mentioned at the end of §3

3 Galois groups

In this section we are concerned with the following problem

Problem 3.1. Given an algebraic number field K and a nonzero polynomial $f \in K[X]$, determine the Galois group G of f over K Can this be done in polynomial time?

In the sequel we will always assume that the polynomial f is squarefree This can be accomplished by means of a good algorithm, which replaces f by $f/\gcd(f, f')$ We denote the degree of f by n

We should specify how we want the algorithm to describe G One possibility is to require that the algorithm comes up with a complete multiplication table of a finite group that is isomorphic to G, but this has an important shortcoming Namely, the group may be very large in comparison to the length of the input, and it may not be possible to write down such a complete multiplication table in polynomial time, let alone calculate it If we insist on a complete multiplication table, then "polynomial time" in Problem 3 1 should be taken to mean polynomial time in the combined lengths of the input *plus* output Theorem 3 2 below shows that Problem 3 1 does in this sense have a polynomial time solution

If we are interested in more efficient algorithms, we should look for a more concise way of describing G For this, we view G as a permutation group of the zeroes of f rather than as an abstract group Numbering the zeroes we see that G may be regarded as a subgroup of the symmetric group S_n of order $n!$, this subgroup is determined only up to conjugacy due to the arbitrary choice of the numbering of the zeroes Instead of asking for a multiplication table of G we shall ask for a list of elements of S_n that generate G Every subgroup of S_n has a system of at most $n-1$ generators (see [52, Lemma 5 2]), and these can be specified using $O(n^2 \log n)$ bits This is bounded by a polynomial function of the length of the input, since the latter is at least n

This formulation of the problem still leaves something to be desired, namely, we do not ask how the numbering of the zeroes of f is related to other ways in which zeroes of f may be specified for example, as complex numbers to a certain precision, for a suitable embedding $K \to \mathbf{C}$, or similarly as p-adic numbers for a suitable prime number p, or as elements of an abstractly defined splitting field or of one of its subfields However, even without such a refined formulation the problem appears to be hard enough

It should be remarked that a set of generators of a subgroup G of S_n can be used to answer, in polynomial time, several natural questions about G For example, one can determine its order, one can decide whether a given element of S_n belongs to G, one can, for a given prime p, determine generators for a Sylow p-subgroup of G, one can find a composition series for G and name the isomorphism types of its composition factors, in particular, one can decide whether G is solvable For more examples, proofs, and references, see [32] It

may be that some of the ideas that underlie this theory, which depends on the classification of finite simple groups, will play a role in a possible solution of Problem 3.1.

The following result, due to Landau [35], expresses that the possibility that G is very large is the only obstruction to finding a good algorithm for Problem 3.1.

Theorem 3.2. *There is a deterministic algorithm that given K and f as in Problem 3.1 and a positive integer b decides whether the Galois group G has order at most b, and if so gives a complete list of elements of G, and that runs in time $(b+l)^{O(1)}$, where l is the length of the data specifying K and f.*

The algorithm is obtained from the standard textbook construction of a splitting field of f over K. One first factors f into irreducible factors in $K[X]$. If all factors are linear, then the splitting field is K itself. Otherwise, one passes to the field $L = K[X]/gK[X]$, where g is one of the nonlinear irreducible factors of f. Then a splitting field of f over L is also one over K, so applying the algorithm recursively one can determine a splitting field of f over K. If at any stage during the recursion it happens that one obtains a field that has degree larger than b over the initial field K, then $\#G > b$, and one stops. If this does not happen, then one eventually arrives at a splitting field M of f over K. As in 2.9 one can determine the group $\mathrm{Gal}(M/K)$ of all K-automorphisms of M, and this is G. It is then easy to make a multiplication table for G and to find an embedding of G into the symmetric group of the set of zeroes of f.

One sees from Theorem 3.2 that G can be determined in time $(\#G+l)^{O(1)}$. Since $\#G \leq n!$, it follows that for bounded n Problem 3.1 is solved in the sense that there is a polynomial time solution. This is an example of a complexity result that does not adequately reflect the practical situation: the practical problem of determining Galois groups is *not* considered to be well solved, even though the algorithms that are actually used nowadays always require n to be bounded—in fact, each value of n typically has its own algorithm (cf. [69, 26]), which does *not* follow the crude approach outlined above.

Corollary 3.3. *There is a good algorithm that given K and f decides whether G is abelian, and determines G if G is abelian and f is irreducible.*

For irreducible f this is easily deduced from Theorem 3.2 with $b = n$, since a transitive abelian permutation group of degree n has order n. For reducible f one uses that the Galois group of f is abelian if and only if the Galois group of each irreducible factor of f is abelian.

For reducible f, this algorithm does not determine the Galois group, and it is not clear whether this can be done in polynomial time. The following problem illustrates the difficulty.

Problem 3.4. Given an algebraic number field K and elements $a_1, a_2, \ldots, a_t \in K$, determine the Galois group of $\prod_{i=1}^{t}(X^2 - a_i)$ over K. Is there a good algorithm for doing this?

For $K = \mathbf{Q}$ this is indeed possible. For general algebraic number fields one can probably do it if one assumes the generalized Riemann hypothesis. Without such an assumption already the case that all a_i are units of the ring of integers

of K is difficult to handle In any case, the algorithm from Theorem 3.2 is in general too slow.

The following pretty result is due to Landau and Miller [36] It shows that one can decide in polynomial time whether f is solvable by radicals over K.

Corollary 3.5. *There is a good algorithm that given K and f decides whether G is solvable*

As in the proof of Corollary 3.3, we may assume that f is irreducible. If there were a bound of the form $n^{O(1)}$ for the order of a solvable transitive permutation group of degree n, then we could proceed in the same way as for abelian groups. However, no such bound exists, since for every integer $k \geq 0$ there is a solvable transitive permutation group of degree $n = 2^k$ and order 2^{n-1}. Instead, one uses that the order of a *primitive* solvable permutation group of degree n does have an upper bound of the form $n^{O(1)}$ (see [54]). By Galois theory, the Galois group G of f is primitive if and only if there are no nontrivial intermediate fields between K and $K(\alpha)$, where $f(\alpha) = 0$. To reduce the general case to this situation, it suffices to find a chain of fields $K = K_0 \subset K_1 \subset \quad \subset K_t = K(\alpha)$ that cannot be refined, since G is solvable if and only if for each i the Galois closure of $K_i \subset K_{i+1}$ has a solvable Galois group. Such a chain can be found inductively if one can, among all intermediate fields $K \subset L \subset K(\alpha)$ with $L \neq K(\alpha)$, find a maximal one. This is done as follows. Factor the polynomial f into monic irreducible factors over $K(\alpha)$. One of the factors is $X - \alpha$ For each other irreducible factor g we define a subfield $L_g \neq K(\alpha)$ containing K as follows. If g is linear, $g = X - \beta$, then $K(\alpha)$ has a unique K-automorphism σ with $\sigma\alpha = \beta$, and we let L_g be the field of invariants of σ. If g is nonlinear, then let β be a zero of g in an extension field of $K(\alpha)$, and $L_g = K(\alpha) \cap K(\beta)$. I claim that all maximal subfields are among the L_g, so that we can find a maximal subfield by choosing a field L_g with the largest degree over K. The correctness of the claim follows by Galois theory from the following purely group theoretic statement. Let G be a finite group, $H \subset J \subset G$ subgroups with $H \neq J$, and assume that there is no subgroup I of G with $H \subset I \subset J$, $H \neq I \neq J$, then there exists $\sigma \in G - H$ such that

$$\langle H, \sigma \rangle = J \quad \text{if } \sigma H \sigma^{-1} = H,$$

$$\langle H, \sigma H \sigma^{-1} \rangle = J \quad \text{if } \sigma H \sigma^{-1} \neq H$$

In fact, it suffices to choose $\sigma \in J - H$

This concludes the sketch of the proof of Corollary 3 5. Note that the algorithm does not determine the group G if it is solvable, even if f is irreducible One does obtain the prime divisors of $\#G$ if G is solvable

Theorem 3 2 suggests that the largest groups are the hardest to determine. However, the following result, which is taken from [34], shows that the *very* largest ones can actually be dealt with in polynomial time As above let S_n denote the full symmetric group of degree n, and let A_n be the alternating group of degree n

Theorem 3.6. *There is a good algorithm that given K and f decides whether the Galois group of f is S_n and whether or not it is A_n*

For this, one may by the above assume that $n \geq 8$ From the classification of finite simple groups it follows (see [18]) that the only sixfold transitive permu-

tation groups of degree n are A_n and S_n. Hence, if we build up the splitting field of f over K as in the proof of Theorem 3.2, then G is A_n or S_n if and only if after adjoining six zeroes of f one has obtained an extension of degree $n(n-1)(n-2)(n-3)(n-4)(n-5)$. One can distinguish between A_n and S_n by computing the discriminant Δ_f of f—this comes down to evaluating a determinant, which can be done in polynomial time—and checking whether $X^2 - \Delta_f$ has a zero in K.

In a similar way one can decide in polynomial time whether G is doubly transitive. If G is doubly transitive, one can determine the isomorphism type of the unique minimal normal subgroup of G in polynomial time, a result that is due to Kantor [31]. If one attempts to determine G itself, one runs into the following problem, which was suggested by Kantor.

Problem 3.7. Is there a polynomial time algorithm that given K and f as in Problem 3.1 and a prime number p decides whether G has a normal subgroup of index p?

Even for $p = 2$ this appears to be difficult.

Resolvent polynomials, such as $X^2 - \Delta_f$ in the proof of Theorem 3.6, play a much more important role in practical algorithms for determining Galois groups than in known complexity results (see [69, 26]).

Problem 3.8. Is there a way to exploit resolvent polynomials to obtain complexity results for varying n?

The results that we have treated so far are more algebraic than arithmetic in nature, the only exception being what we said about Problem 3.4. It should be possible to formulate and prove similar results for other sufficiently explicitly given fields over which polynomials in one variable can be factored efficiently. We now turn to techniques that do exploit the arithmetic of the field. The natural way to do this is to first consider the case of finite and local base fields.

Let E be a finite field, $f \in E[X]$ a nonzero polynomial, and n its degree. As we mentioned in 2.8, there is a good algorithm that, given E and f, determines the factorization type of f in $E[X]$. This immediately gives rise to the Galois group G, which is cyclic of order equal to the least common multiple of the degrees of the irreducible factors of f. One also obtains the cycle pattern of a permutation that generates G as a permutation group. Note that already in the case of finite fields the order of G may, for reducible f, be so large that the elements of G cannot be listed one by one in polynomial time.

We next discuss local fields.

Problem 3.9. Given a local field F and a polynomial $f \in F[X]$ with a nonzero discriminant, determine the Galois group G of f over F. What is the complexity of this problem? Is there a good algorithm for it?

I am not aware of any published work that has been done on Problem 3.9, and I will only make a few brief remarks, restricting myself to the case that F is non-archimedean. Once a satisfactory theory of factoring polynomials has been developed (see 2.11), one can prove an analogue of Theorem 3.2. This does not yet solve the problem, since even when f is irreducible the Galois group may have a very large order. Tamely ramified extensions are small, however, which suggests that the following problem should be doable.

Problem 3.10. Given F and f as in Problem 3.9, with F non-archimedean, decide whether a splitting field of f over F is tamely ramified, and if so determine its Galois group over F. Can this be done in polynomial time?

When this problem is solved, one is left with wildly ramified extensions, which occur only if p is small. In that case, one may first want to consider the following problem, which looks harder than Problem 3.10.

Problem 3.11. Given F and f as in Problem 3.9, with F non-archimedean, determine the Galois group of the maximal tamely ramified subextension M of a splitting field of f over F. Can this be done in polynomial time?

If f is irreducible of degree n, then the field M in Problem 3.11 has degree at most n^4 over F. This follows from a group-theoretic argument that was shown to me by I. M. Isaacs.

Even when all local problems are completely solved it is not clear whether they are very helpful in solving Problem 3.1. There is a well-known heuristic technique that can be used to obtain information about the Galois group, which comes down to first considering the local Galois group at primes that do not divide the discriminant of f (see [73, §1]). Not much can be proved about this method, however (cf. [34, §4]). G. Cornell has suggested to look instead at the *ramifying* primes, the rationale being that Problem 3.1 should be reducible to the case $K = \mathbf{Q}$, in which case the Galois group is generated by the inertia groups.

4. Rings of integers

In this section we consider the following problem and its complexity.

Problem 4.1. Given an algebraic number field K, determine its ring of integers $\mathcal{O}$.

Constructing an order in K as in 2.10 we see that this problem is equivalent to the following one.

Problem 4.2. Given an order A in a number field K, determine the ring of integers $\mathcal{O}$ of K.

Much of the literature on this problem assumes that the given order is an equation order $\mathbf{Z}[\alpha]$, and it is true that equation orders offer a few advantages in the initial stages of several algorithms. It may be that in many practical circumstances one never gets beyond these initial stages (cf. [8, Preface]), but in the worst case—which is what we are concerned with when we estimate the complexity of a problem—these advantages quickly disappear as the algorithm proceeds. For this reason we make no special assumptions about A except that it is an order.

Most of what we have to say about Problem 4.2 also applies to the following more general problem.

Problem 4.3. Given a commutative ring A of which the additive group is isomorphic to $\mathbf{Z}^n$ for some n, and that has a nonvanishing discriminant over $\mathbf{Z}$, determine the maximal order in $A \otimes_{\mathbf{Z}} \mathbf{Q}$.

It is not difficult to show that Problems 4.2 and 4.3 are equivalent under deterministic polynomial time reductions.

The main result on Problem 4.1, which is due to Chistov [22, 14], is a negative one.

Theorem 4.4. *Under deterministic polynomial time reductions, Problem* 4.1 *is equivalent to the problem of finding the largest square factor of a given positive integer.*

The problem of finding the largest square factor of a given positive integer m is easily reduced to Problem 4.1 by considering the number field $K = \mathbf{Q}(\sqrt{m})$. For the opposite reduction, which in computer science language is a "Turing" reduction, we refer to the discussion following Theorem 4.6 below.

Since there is no known algorithm for finding the largest square factor of a given integer m that is significantly faster than factoring m (see [43, §2]), Theorem 4.4 shows that Problem 4.1 is currently intractable. More seriously, even if someone *gives* us $\mathcal{O}$, we are not able to recognize it in polynomial time, even if probabilistic algorithms are allowed. Deciding whether the given order A in Problem 4.2 equals $\mathcal{O}$ is currently an infeasible problem, just as deciding whether a given positive integer is squarefree is infeasible. This is not just true in theory, it is also true in practice.

One possible conclusion is that $\mathcal{O}$ is not an object that one should want to work with in algorithms. It may very well be that whenever $\mathcal{O}$ is needed one can just as well work with an order A in K, and *assume* that A equals $\mathcal{O}$ until evidence to the contrary is obtained. This may happen, for example, when a certain nonzero ideal of A is found not to be invertible; in that case one can, in polynomial time, construct an order A' in K that strictly contains A and proceed with A' instead of A.

If it indeed turns out to be wise to avoid working with $\mathcal{O}$, then it is desirable that more attention be given to general orders, both algorithmically and theoretically (cf. [59]). This is precisely what has happened in the case of quadratic fields (cf. [45, 49, 28]).

The order A equals $\mathcal{O}$ if and only if all of its nonzero prime ideals $\mathfrak{p}$ are nonsingular; here we call $\mathfrak{p}$ nonsingular if the local ring $A_{\mathfrak{p}}$ is a discrete valuation ring, which is equivalent to $\dim_{A/\mathfrak{p}} \mathfrak{p}/\mathfrak{p}^2 = 1$. One may wonder, if it is intractable to find $\mathcal{O}$, can one at least find an order in K containing A of which the singularities are bounded in some manner? One result of this sort is given below in Theorem 4.7; it implies that given A, one can find an order B in K containing A such that all singularities $\mathfrak{p}$ of B are *plane* singularities, i.e., satisfy $\dim_{B/\mathfrak{p}} \mathfrak{p}/\mathfrak{p}^2 = 2$.

The geometric terminology just used should remind us of a situation in which there does exist a good method for finding the largest square factor, namely, if we are dealing with polynomials in one variable over a field. Thus, Theorem 4.4 suggests that, for a finite field E, finding the integral closure of the polynomial ring $E[t]$ in a given finite extension of $E(t)$ is a tractable problem, and results of this nature have indeed been obtained (see [22]). In geometric language, this means that it is feasible to resolve the singularities of a given irreducible algebraic curve over a given finite field. The corresponding problem over fields of characteristic zero has been considered as well (see [71]), and one may wonder whether the geometric techniques that have been proposed can also be used in the context of Problem 4.2. In any case, we can formulate Problem 4.2 geometrically by asking for the resolution of the singularities of a given irreducible *arithmetic curve.*

For many purposes, resolving singularities is a local problem, but as we see from Theorem 4 4 that is not quite the case in the context of algorithms It may be that one only needs to look locally at those prime ideals $\mathfrak{p}$ of A for which $\dim_{A/\mathfrak{p}} \mathfrak{p}/\mathfrak{p}^2 > 1$, but how does one *find* those prime ideals? And likewise, if $A \cong \mathbf{Z}[X]/f\mathbf{Z}[X]$ is an equation order, then, as all textbooks point out, one only needs to look locally at those prime numbers p for which p^2 divides the discriminant of f, but how does one *find* those prime numbers? By contrast, once one *knows* at which $\mathfrak{p}$ or p to look, the problem does admit a solution To formulate it we introduce some notation

Let A be an order in a number field K of degree n Let further C be a subring of A, for us, the most interesting cases are $C = A$ and $C = \mathbf{Z}$ For any nonzero prime ideal $\mathfrak{p}$ of C we define

$$A^{(\mathfrak{p})} = \{\beta \in \mathscr{O} \quad \mathfrak{p}^m \beta \subset A \text{ for some } m \in \mathbf{Z}_{\geq 0}\},$$

this is the "$\mathfrak{p}$-primary part" of $\mathscr{O}$ when viewed modulo A It is not difficult to show that $A^{(\mathfrak{p})}$ is an order in K and that it is the smallest order in K containing A with the property that all its prime ideals containing $\mathfrak{p}$ are nonsingular In addition, one has an isomorphism $\mathscr{O}/A \cong \bigoplus_{\mathfrak{p}} A^{(\mathfrak{p})}/A$ of C-modules, with $\mathfrak{p}$ ranging over the set of nonzero prime ideals of C, and $A^{(\mathfrak{p})} = A$ for all but finitely many $\mathfrak{p}$ Thus, to determine $\mathscr{O}$, it suffices to determine all $A^{(\mathfrak{p})}$ For a single $\mathfrak{p}$, we have the following result

Theorem 4.5. *There is a good algorithm that given K, A, C, $\mathfrak{p}$ as above, determines $A^{(\mathfrak{p})}$*

This is proved by analyzing an algorithm of Zassenhaus [77, 78] We briefly sketch the main idea Let us first consider the case $C = \mathbf{Z}$ Denote by p the prime number for which $\mathfrak{p} = p\mathbf{Z}$, and write $A^{(p)} = A^{(\mathfrak{p})}$

One needs a criterion for A to be equal to $A^{(p)}$ The *multiplier ring* $R_{\mathfrak{a}}$ of a nonzero A-ideal $\mathfrak{a}$ is defined by

$$R_{\mathfrak{a}} = \{\beta \in K \quad \beta\mathfrak{a} \subset \mathfrak{a}\},$$

this is an order in K containing A By $\mathfrak{q}$ we shall denote a typical prime ideal of A that contains p, and we let $\mathfrak{r}$ be the product of all such $\mathfrak{q}$ By standard commutative algebra, A equals $A^{(p)}$ if and only if all $\mathfrak{q}$ are invertible, and $\mathfrak{q}$ is invertible if and only if $R_{\mathfrak{q}} = A$ Also, each $R_{\mathfrak{q}}$ is contained in $R_{\mathfrak{r}}$, so that we can decide whether or not A equals $A^{(p)}$ by looking at $R_{\mathfrak{r}}$ More precisely, if $R_{\mathfrak{r}} = A$ then $A = A^{(p)}$, and if $R_{\mathfrak{r}}$ properly contains A then so does $A^{(p)}$, since clearly $R_{\mathfrak{r}} \subset A^{(p)}$

I claim that to turn the above considerations into an algorithm it suffices to have a way of determining $\mathfrak{r}$ Namely, suppose that $\mathfrak{r}$ is known Then one can determine $R_{\mathfrak{r}}$ by doing linear algebra over $\mathbf{F}_p$, using that $pR_{\mathfrak{r}}/pA$ is the kernel of the $\mathbf{F}_p$-linear map $A/pA \to \mathrm{End}(\mathfrak{r}/p\mathfrak{r})$ that sends each $x \in A/pA$ to the multiplication-by-x map If this map is found to be injective, then $R_{\mathfrak{r}} = A$, and the algorithm terminates with $A^{(p)} = A$ If it is not injective, then $R_{\mathfrak{r}}$ strictly contains A In that case one replaces A by $R_{\mathfrak{r}}$ and starts all over again Note that the number of iterations is bounded by $(\log|\Delta_A|)/(2\log p)$, where Δ_A denotes the discriminant of A

It remains to find an algorithm for determining $\mathfrak{r}$. Since the ideals $\mathfrak{q}$ are pairwise coprime, $\mathfrak{r}$ is their intersection, so $\mathfrak{r}/pA$ is the set of nilpotents of the finite ring A/pA. It can, again by linear algebra, be found as the kernel of the $\mathbf{F}_p$-linear map $A/pA \to A/pA$ that sends each $x \in A/pA$ to x^{p^t}, here t is the least positive integer for which $p^t \geq n$.

This concludes the sketch of the algorithm underlying Theorem 4.5 for $C = \mathbf{Z}$. For general C, one can either modify the above, or first determine $A^{(p)}$ for $p = \operatorname{char} C/\mathfrak{p}$ and then find $A^{(\mathfrak{p})}$ inside $A^{(p)}$.

The above algorithm gives, with a few modifications, also something if p is not supposed to be prime. This is expressed in the following theorem, which is taken from [14].

Theorem 4.6. *There is a good algorithm that given K and A as above, as well as an integer $q > 1$, determines an order B in K that contains $A^{(p)}$ for each prime number p that divides q exactly once.*

To prove this, one first observes that it suffices to exhibit a good algorithm that given K, A and q either finds B as in the statement of the theorem, or finds a nontrivial factorization $q = q_1 q_2$. Namely, in the latter case one can proceed recursively with q_1 and q_2 to find orders B_1, B_2, and one lets B be the ring generated by B_1 and B_2.

To find B or q_1, q_2, one applies the algorithm outlined above, with a few changes. The first change is that one starts by checking that q is not divisible by any prime number $p \leq n$; if it is, then either one finds a nontrivial splitting of q, or q is a small prime number and one can apply the earlier algorithm. So let it now be assumed that q has no prime factors $p \leq n$, and that $q > 1$. The second change is that one replaces, in the above algorithm, p and $\mathbf{F}_p$ everywhere by q and $\mathbf{Z}/q\mathbf{Z}$. This affects the linear algebra routines, which are only designed to work for vector spaces over fields. However, as we indicated in 2.4, they work just as well for modules over a ring $\mathbf{Z}/q\mathbf{Z}$, *until* some division in $\mathbf{Z}/q\mathbf{Z}$ fails, in which case one obtains a nontrivial factor q_1 of q. The third change is that $\mathfrak{r}/q\mathbf{Z}$ should now be calculated as the "radical of the trace form," i.e., as the kernel of the $\mathbf{Z}/q\mathbf{Z}$-linear map $A/qA \to \operatorname{Hom}(A/qA, \mathbf{Z}/q\mathbf{Z})$ that sends x to the map sending y to $\operatorname{Tr}(xy)$, where $\operatorname{Tr}: A/qA \to \mathbf{Z}/q\mathbf{Z}$ is the trace map. If q is a prime number exceeding n then this is the same $\mathfrak{r}$ as above.

One can show that the modified algorithm has the desired properties, see [14]. This concludes our sketch of the proof of Theorem 4.6.

Using Theorem 4.6 we can complete the proof of Theorem 4.4. Namely, suppose that one has an algorithm that determines the largest square divisor of any given positive integer. Calling this algorithm a few times, one can determine the largest squarefree number q for which q^2 divides the discriminant of A. Applying the algorithm of Theorem 4.6 to q one obtains an order B that contains $A^{(p)}$ for each prime p for which p^2 divides the discriminant of A, so that $B = \mathscr{O}$.

We now formulate a result that also gives information about the local structure of B at primes p for which p^2 divides q. Let A be an order in a number field K, and let q be a positive integer. We call A *nonsingular* at q if each prime ideal of A containing q is nonsingular. We call A *tame* at q if for each prime ideal $\mathfrak{p}$ of A containing q there exist an unramified extension R of the

ring $\mathbf{Z}_p$ of p-adic integers, where $p = \operatorname{char} A/\mathfrak{p}$, a positive integer e that is not divisible by p, and a unit $u \in R^*$, such that there is an isomorphism

$$\varprojlim_m A/\mathfrak{p}^m \cong R[X]/(X^e - uq)R[X]$$

of $\mathbf{Z}_p$-algebras As a partial justification of the terminology, we remark that for prime q the order A is tame at q if and only if each prime ideal $\mathfrak{p}$ of A containing q is nonsingular and tamely ramified over q, this follows from a well-known structure theorem for tamely ramified extensions of $\mathbf{Z}_q$ (see [75, §3-4]) If A is tame at q and $\mathfrak{p}$ is a prime ideal of A containing q, then $\mathfrak{p}$ is nonsingular if and only if either $p = \operatorname{char} A/\mathfrak{p}$ divides q exactly once or the number e above equals 1, and otherwise $\mathfrak{p}$ is a plane singularity

Theorem 4.7. *There is a good algorithm that, given an order A in a number field K of degree n, finds an order B in K containing A and a sequence of pairwise coprime divisors q_i, $1 \le i \le t$, of the discriminant of B, such that*

(i) *B is tame at $q = \prod_{i=1}^{t} q_i$,*
(ii) *all prime numbers dividing q exceed n,*
(iii) *B is nonsingular at all prime numbers p that do not divide q*

This follows from a closer analysis of the algorithm of Theorem 4 6 Using this theorem and the properties of tameness, one can deduce the following result, which expresses that one can approximate $\mathscr{O}$ as closely as can be expected on the basis of Theorem 4 4

Theorem 4.8. *There is a good algorithm that, given an order A in a number field K, finds an order B in K containing A and a positive integer q dividing the discriminant of B such that $B = \mathscr{O}$ if and only if q is squarefree, and such that the primes dividing $[\mathscr{O} : B]$ are exactly those that appear at least twice in q Moreover, there is a good algorithm that given this B and a nontrivial square dividing q finds an order in K that strictly contains B*

Next we discuss an algorithm that does a little more than the algorithm of Theorem 4 5 Namely, in addition to finding $A^{(\mathfrak{p})}$, it also finds all prime ideals of $A^{(\mathfrak{p})}$ containing $\mathfrak{p}$ It depends—not surprisingly, if one considers the case of an equation order $\mathbf{Z}[\alpha]$—on an algorithm for factoring polynomials in one variable over a finite field, see 2 8 Due to this ingredient it is not a deterministic polynomial time algorithm any more, and it has no extension as Theorem 4 6 that works for nonprimes

Theorem 4.9. *There is a probabilistic algorithm that runs in expected polynomial time, and there is a deterministic algorithm that runs in $\sqrt{\operatorname{char} C/\mathfrak{p}}$ times polynomial time, that given K, A, C, $\mathfrak{p}$ as in Theorem 4 5, determine*

(i) *all prime ideals of A containing $\mathfrak{p}$,*
(ii) *the order $A^{(\mathfrak{p})}$,*
(iii) *all prime ideals of $A^{(\mathfrak{p})}$ containing $\mathfrak{p}$*

One can do part (i) by analyzing the structure of the finite ring $A/\mathfrak{p}A$, as the reader may check, below we give a different argument Once one has (i), one can do (ii) by Theorem 4 5 and (iii) by applying (i) to $A^{(\mathfrak{p})}$ We sketch an alternative way to proceed, in which one constructs $A^{(\mathfrak{p})}$ and the prime ideals

simultaneously without appealing to Theorem 4.5 Let it first be assumed that $C = A$.

The algorithm works with a list of pairs B, $\mathfrak{q}$ for which B is an order in K with $A \subset B \subset A^{(\mathfrak{p})}$ and $\mathfrak{q}$ is a prime ideal of B containing $\mathfrak{p}$ Initially, there is only one pair on the list, namely, A, $\mathfrak{p}$ The purpose of the algorithm is to achieve that $\mathfrak{q}$ is nonsingular as a prime ideal of B, for each pair B, $\mathfrak{q}$ on the list If that happens, then $A^{(\mathfrak{p})}$ is the sum of all B's, and, as it turns out, the ideals $\mathfrak{q}A^{(\mathfrak{p})}$ are pairwise distinct and are precisely all prime ideals of $A^{(\mathfrak{p})}$ containing $\mathfrak{p}$.

The algorithm deals with a given pair B, $\mathfrak{q}$ in the following manner. First one determines, by means of linear algebra over the finite field $B/\mathfrak{q}$, an element $\gamma \in K$ with $\gamma \notin B$, $\gamma\mathfrak{q} \subset B$; such an element exists, see [75, Lemma 4-4-3] Next, one considers $\gamma\mathfrak{q}$. If $\gamma\mathfrak{q} \not\subset \mathfrak{q}$, then $\mathfrak{q}$ is nonsingular, and the pair B, $\mathfrak{q}$ is left alone. Suppose now that $\gamma\mathfrak{q} \subset \mathfrak{q}$. Then $B[\gamma]$ is an order in K in which $\mathfrak{q}$ is an ideal, and using linear algebra one determines the minimal polynomial g of $(\gamma \bmod \mathfrak{q})$ over the field $B/\mathfrak{q}$. This polynomial is factored into irreducible factors over $B/\mathfrak{q}$. For each irreducible factor $(h \bmod \mathfrak{q})$ of g, one now adds the pair $B[\gamma]$, $\mathfrak{q} + h(\gamma)B[\gamma]$ to the list, and one removes B, $\mathfrak{q}$.

The above is repeated until all pairs are nonsingular

If $C \neq A$, then one replaces the pair C, $\mathfrak{p}$ by $A' = C + \mathfrak{p}A$, $\mathfrak{p}A$; note that $\mathfrak{p}A$ is a prime ideal of A' with $A'/\mathfrak{p}A \cong C/\mathfrak{p}$. Applying the above with A' in the role of A one finds the order $A'^{(\mathfrak{p})}$ and all of its prime ideals containing $\mathfrak{p}$. One easily shows that $A^{(\mathfrak{p})} = A'^{(\mathfrak{p})}$, and intersecting the prime ideals just mentioned with A one finds (1). This concludes the sketch of the proof of Theorem 4.9

We note that the above algorithm also gives a convenient way of evaluating the valuations corresponding to the prime ideals containing $\mathfrak{p}$. Namely, for each nonsingular pair B, $\mathfrak{q}$ the corresponding valuation v is given by

$$v(\beta) = \max\{m \in \mathbf{Z}_{\geq 0}\colon \gamma^m\beta \in B\}$$

for $\beta \in B$, $\beta \neq 0$, where γ is as constructed in the algorithm. Since each element of K can be written as a quotient of elements of B this allows us to compute $v(\beta)$ for each $\beta \in K$.

It is well known that the p-adic valuations of a number field $K = \mathbf{Q}(\alpha)$ correspond bijectively to the irreducible factors of f over $\mathbf{Q}_p$, where f is the irreducible polynomial of α over $\mathbf{Q}$ Thus Theorem 4.9 suggests that factoring polynomials in one variable over $\mathbf{Q}_p$ to a given precision can be done by a probabilistic algorithm that runs in expected polynomial time and by a deterministic algorithm that runs in $\sqrt{p}$ times polynomial time A result of this nature is given in [14], see also [21], where a more direct approach is taken.

We close this section with a problem that is geometrically inspired.

Problem 4.10. If all singularities of A are plane singularities, can the algorithm of Theorem 4 9 be arranged in such a way that the same applies to all rings B that are encountered?

It may be of interest to see whether the methods that have been proposed for the resolution of plane curve singularities [11, 71] shed any light on this problem. One may also wish to investigate the algorithm of Theorem 4 6 from the same perspective.

An affirmative answer to Problem 4.10 may improve the performance of the algorithm. This is because the hypothesis on A is often satisfied, for example, if A is an equation order or a "generalized" equation order as in 2.10; and finding γ in the algorithm of Theorem 4.9 may become easier if q is at worst a plane singularity, so that it can be generated by two elements.

5. Class groups and units

In this section we discuss the following problem and its complexity.

Problem 5.1. Given an algebraic number field K, with ring of integers $\mathcal{O}$, determine the unit group $\mathcal{O}^*$ and the class group $\mathrm{Cl}\mathcal{O}$ of $\mathcal{O}$.

First we make a few remarks on the statement of the problem. In the previous section we saw that, given K, the ring $\mathcal{O}$ may be very hard to determine and that consequently we may have to work with subrings A of $\mathcal{O}$ that, for all we know, may be different from $\mathcal{O}$. Thus, it would have been natural to formulate the problem for any order A in K rather than just for $\mathcal{O}$. We have not done so, for several reasons. The first is that only very little work has been done for general orders in fields of degree greater than 2. The second is that most difficulties appear already in the case $A = \mathcal{O}$ and that some additional complications are avoided. Finally, it is to be noted that all algorithms for calculating unit groups and class groups that have been proposed are so time-consuming that the effort required in determining $\mathcal{O}$ appears to be negligible in comparison; and it may very well be that the best way of calculating the unit group and class group of a general order A proceeds by first determining $\mathcal{O}$, next calculating $\mathcal{O}^*$ and $\mathrm{Cl}\mathcal{O}$, and finally going back to A.

We shall denote by n and Δ the degree and the discriminant of K over $\mathbf{Q}$. It will be assumed that $\mathcal{O}$ is given by means of a multiplication table of length $(2+\log|\Delta|)^{O(1)}$, as in 2.10. We shall bound the running times of the algorithms in terms of $|\Delta|$.

The next question to be discussed is how we wish $\mathcal{O}^*$ and $\mathrm{Cl}\mathcal{O}$ to be specified. As an abstract group, we have $\mathcal{O}^* \cong (\mathbf{Z}/w\mathbf{Z}) \oplus \mathbf{Z}^{r+s-1}$, where w denotes the number of roots of unity in K and r, s denote the number of real and complex archimedean places of K, respectively. Determining $\mathcal{O}^*$ means specifying the images of the standard generators of $(\mathbf{Z}/w\mathbf{Z}) \oplus \mathbf{Z}^{r+s-1}$ under an isomorphism to $\mathcal{O}^*$; and we also like to be provided with an algorithm that calculates the inverse isomorphism. Using the logarithms at the infinite places (see [37, Chapter V, §1]) and basis reduction (see 2.6) one can prove that both these things can be achieved if we have a set of generators for $\mathcal{O}^*$. However, just *writing down* a set of generators for $\mathcal{O}^*$ may be very time-consuming. Suppose, for example, that K is real quadratic, i.e., $n = 2$ and $\Delta > 0$. Then $\mathcal{O}^*$ is generated by -1 and a single unit ε of infinite order. It is easy to see that the total number of digits of the coefficients of ε on the given basis of $\mathcal{O}$ over $\mathbf{Z}$ equals $R(\log\Delta)^{O(1)}$, where R denotes the regulator of K; see [37, Chapter V, §1] for the definition of the regulator. It is reasonable to conjecture that, for an infinite sequence of real quadratic fields, R is as large as $\Delta^{1/2+o(1)}$. Hence we cannot expect to be able to write down ε, let alone calculate it, in time significantly less than $\Delta^{1/2}$. If we are interested in more efficient algorithms, then units must be represented in a different way, for example as a product $\prod \gamma^{k(\gamma)}$ of elements $\gamma \in K^*$ with integer exponents $k(\gamma)$ that may be very large

in absolute value. This leads to the question whether there exists a system of generating units that one can express in this way using substantially fewer than $|\Delta|^{1/2}$ bits. Also, the following problem is suggested.

Problem 5.2. Given a number field K, finitely many elements $\gamma \in K^*$, and, for each γ, an integer $k(\gamma) \in \mathbf{Z}$, decide whether $\varepsilon = \prod_\gamma \gamma^{k(\gamma)}$ is a unit, i.e., belongs to $\mathscr{O}^*$, and whether it equals 1. If it is a unit, then determine its residue class modulo a given ideal and calculate, for a given embedding $\sigma\colon K \to \mathbf{C}$, the logarithm of $\sigma\varepsilon$ to a given precision.

It may be expected that the first of these—recognizing units—can be done by means of a good algorithm, even when $\mathscr{O}$ is not given, by means of *factor refinement* (cf. [7]). Good results on the other problems can probably be obtained with diophantine approximation techniques, such as basis reduction (see 2.6). The same applies to the following more general problem.

Problem 5.3. Given a number field K and a finite set Γ of elements $\gamma \in K^*$, find sets of generators for the subgroups

$$\left\{(k(\gamma))_{\gamma\in\Gamma} \in \mathbf{Z}^\Gamma\colon \prod_{\gamma\in\Gamma} \gamma^{k(\gamma)} = 1\right\}, \qquad \left\{(k(\gamma))_{\gamma\in\Gamma} \in \mathbf{Z}^\Gamma\colon \prod_{\gamma\in\Gamma} \gamma^{k(\gamma)} \in \mathscr{O}^*\right\}$$

of $\mathbf{Z}^\Gamma$ and calculate the regulator of the group of all units of the form $\prod_{\gamma\in\Gamma} \gamma^{k(\gamma)}$, $k(\gamma) \in \mathbf{Z}$, to a given precision.

Problems of this nature arise in several contexts: in an algorithm for factoring integers [44, 17], in the discrete logarithm problem [27, 60], as we shall see below; in the determination of unit groups and class groups.

Returning to Problem 5.1, we still have to describe how we wish the class group $\mathrm{Cl}\mathscr{O}$ to be specified. It is a finite abelian group, so we may first of all ask for positive integers $d_1, d_2, \ldots, d_t$ such that there is an isomorphism $\bigoplus_t \mathbf{Z}/d_t\mathbf{Z} \cong \mathrm{Cl}\mathscr{O}$ of abelian groups, and secondly for ideals $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_t$ such that one such isomorphism sends the standard generators of $\bigoplus_t \mathbf{Z}/d_t\mathbf{Z}$ to the ideal classes of the $\mathfrak{a}_t$. Once the class group has been calculated in this sense, it may remain very difficult to find the *inverse* isomorphism: given an $\mathscr{O}$-ideal, to which ideal of the form $\prod_t \mathfrak{a}_t^{m(t)}$ is it equivalent? Even testing whether a given ideal is *principal* may be very difficult.

The order $h = \#\mathrm{Cl}\mathscr{O}$ of the class group is bounded by $|\Delta|^{1/2}(n + \log|\Delta|)^{n-1}$ (see Theorem 6.5). The example of imaginary quadratic fields—i.e., $n = 2$ and $\Delta < 0$—shows that h is often as large as $|\Delta|^{1/2}(\log|\Delta|)^{O(1)}$. Hence, if we are willing to spend time at least of order $|\Delta|^{1/2}$ then we could conceivably list all ideal classes, and finding the inverse isomorphism might also become doable.

The first thing to be discussed about Problem 5.1 is whether it can be done at all, efficiently or not. This is a question that is strangely overlooked in most textbooks, two notable exceptions being [9] and [19]. For the class group, one often finds the theorem that every ideal class contains an integral ideal of norm at most the Minkowski constant $(n!/n^n)(4/\pi)^s|\Delta|^{1/2}$, where s denotes the number of complex places of K. However, this does not show that the class group is effectively computable if no effective procedure for deciding equivalence of ideals is supplied.

We shall prove a theorem from which the effective computability of $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ is clear. We begin by introducing some notation. Let K be a number field

of degree n and discriminant Δ over $\mathbf{Q}$. A *place* $\mathfrak{p}$ of K is an equivalence class of nontrivial absolute values of K. The set of archimedean places of K is denoted by S_∞. For $\mathfrak{p} \notin S_\infty$, the *norm* $\mathfrak{N}\mathfrak{p}$ of $\mathfrak{p}$ is the cardinality of the residue class field at $\mathfrak{p}$. For each place $\mathfrak{p}$, let $|\ |_\mathfrak{p}: K \to \mathbf{R}_{\geq 0}$ denote the unique absolute value belonging to $\mathfrak{p}$ with the property that $|2|_\mathfrak{p} = 2$ if $\mathfrak{p}$ is real, $|2|_\mathfrak{p} = 4$ if $\mathfrak{p}$ is complex, and $|K^*|_\mathfrak{p} = (\mathfrak{N}\mathfrak{p})^{\mathbf{Z}}$ if $\mathfrak{p}$ is non-archimedean. The *height* $H(x)$ of an element $x \in K$ is defined by $H(x) = \prod_\mathfrak{p} \max\{1, |x|_\mathfrak{p}\}$, the product extending over all places $\mathfrak{p}$ of K. For any set S of places of K with $S_\infty \subset S$ we let K_S denote the group of S-units, i.e., the subgroup of K^* consisting of those $x \in K^*$ that satisfy $|x|_\mathfrak{p} = 1$ for all places $\mathfrak{p}$ of K with $\mathfrak{p} \notin S$; in particular, we have $K_{S_\infty} = \mathscr{O}^*$ if $\mathscr{O}$ denotes the ring of integers of K.

Theorem 5.4. *Let K be an algebraic number field, Δ its discriminant over $\mathbf{Q}$, and s the number of complex places of K. Let $d = (2/\pi)^s|\Delta|^{1/2}$, and $S = S_\infty \cup \{\mathfrak{p}\ \mathfrak{p}$ is a finite place of K with $\mathfrak{N}\mathfrak{p} \leq d\}$. Then the group K_S is generated by the set of those $x \in K_S$ for which $H(x) \leq d^2$, and the ideal class group of the ring of integers of K is generated by the ideal classes of the finite primes in S.*

The proof of this theorem is given in §6.

Remark. The example of real quadratic fields shows that it is not reasonable to expect that the group $K_{S_\infty} = \mathscr{O}^*$ is generated by elements x for which $H(x)$ is substantially smaller than e^d. The group K_S in Theorem 5.4 is generally much larger than $\mathscr{O}^*$, but it is generated by elements that are much smaller.

The relevance of Theorem 5.4 for the effective determination of $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ comes from the exact sequence

$$0 \to \mathscr{O}^* \to K_S \to \mathbf{Z}^{S-S_\infty} \to \mathrm{Cl}\mathscr{O} \to 0.$$

The middle arrow sends an element $x \in K_S$ to the vector $(\mathrm{ord}_\mathfrak{p} x)_{\mathfrak{p} \in S - S_\infty}$, where $\mathrm{ord}_\mathfrak{p} x$ is the number of factors $\mathfrak{p}$ in x, so $|x|_\mathfrak{p} = \mathfrak{N}\mathfrak{p}^{-\mathrm{ord}_\mathfrak{p} x}$. The map $\mathbf{Z}^{S-S_\infty} \to \mathrm{Cl}\mathscr{O}$ sends $(m(\mathfrak{p}))_\mathfrak{p}$ to the ideal class of $\prod_\mathfrak{p} \mathfrak{p}^{m(\mathfrak{p})}$. The exactness at $\mathrm{Cl}\mathscr{O}$ follows from the last assertion of Theorem 5.4, the exactness at the other places is clear.

To calculate $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ from the sequence, one starts by calculating the set of generators of K_S given by Theorem 5.4. It is well known that there are only finitely many elements of bounded height in K (see [64, Chapter 2]), and from the proof of this result it is clear that they can be effectively determined. Determining the prime ideal factorizations of these generators one finds a matrix that describes the map $K_S \to \mathbf{Z}^{S-S_\infty}$. Applying algorithms for finitely generated abelian groups (see 2.5) one obtains $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ as the kernel and cokernel of this map.

We now turn to complexity results for Problem 5.1. Most results that have been obtained concern quadratic fields (see [45, 61, 28]). For general number fields, virtually all that is known can be found in [12] (note that, in that paper, $R^{1/2}\mathscr{D}^\varepsilon$ in Theorem 2 is a printing error for $R\mathscr{D}^\varepsilon$, and $\mathscr{D}^{1/2+\varepsilon}$ in Theorem 4 is a printing error for $R^{1/2}\mathscr{D}^\varepsilon$). The following theorem appears to be true.

Theorem 5.5. *Given K and $\mathscr{O}$, one can determine a set of generators of $\mathscr{O}^*$ and the structure of $\mathrm{Cl}\mathscr{O}$ in time at most $(2 + \log|\Delta|)^{O(n)}|\Delta|^{3/4}$ by means of a*

deterministic algorithm and in expected time at most $(2+\log|\Delta|)^{O(n)}|\Delta|^{1/2}$ *by means of a probabilistic algorithm*

In [12] one finds a weaker version of this result, in which n is kept fixed The more precise result should follow by combining [12] with results that appear in [15].

The algorithm underlying Theorem 5 5, for which we refer to [12] and the references given there, is not the same as the method for effectively determining $\mathcal{O}^*$ and $\mathrm{Cl}\mathcal{O}$ that we just indicated. However, there does exist a connection between the two methods Namely, the proof of Theorem 5 4 depends on a lemma from combinatorial group theory that constructs a set of generators of a subgroup H of a group G from a set of generators of G itself (see Lemma 6 3), whereas the algorithm of Theorem 5 5 constructs generators of the group $\mathcal{O}^*$ by letting it act on a certain graph, and it is well known that these two subjects are closely related (see [63]) It would be of interest to understand this connection better, and to see whether Theorem 5 5 can be deduced from a suitable version of Theorem 5.4

The higher exponent 3/4 in Theorem 5.5 in the case of a deterministic algorithm is due to the use of algorithms for factoring polynomials over finite fields (see 2 8). It suggests the following problem.

Problem 5.6. Can the exponent 3/4 in Theorem 5.5 be replaced by 1/2?

For quadratic fields the answer is affirmative It is likely that the method by which this is shown, which is not completely obvious, carries over to general number fields.

We close this section with an imprecise description of a probabilistic technique for the solution of Problem 5 1.

Let the notation be as introduced before Theorem 5.4, and let S consist of the archimedean primes of K and the non-archimedean primes of norm up to a certain bound b. One supposes that one has a method of drawing elements of K_S that are "random" in a certain sense For example, the method might consist of drawing elements x of K whose coordinates on the given vector space basis of K over $\mathbf{Q}$ are uniformly distributed over a certain set of rational numbers, such as the positive integers up to a certain bound, and keeping only those x that are found to belong to K_S.

To determine the class group and the units, one draws elements of K_S until one has the feeling that the subgroup H that they generate is equal to all of K_S. One may get this feeling if the number of elements that have been drawn is well over $\#S$, which is the minimal number of generators of K_S as an abelian group, and if it happened several times in succession that a newly drawn element of K_S was found to belong to the subgroup generated by the elements drawn earlier, if Problem 5 3 has a satisfactory solution then this can be tested Assuming that $H = K_S$ one can determine $\mathcal{O}^*$ and $\mathrm{Cl}\mathcal{O}$, as above, as the kernel and cokernel of the map $\phi\colon H \to \mathbf{Z}^{S-S_\infty}$ that sends x to $(\mathrm{ord}_\mathfrak{p} x)_{\mathfrak{p}\in S-S_\infty}$.

In general, one does not know that $H = K_S$, so that $\ker\phi$ and $\mathrm{coker}\,\phi$ can only be conjectured to be $\mathcal{O}^*$ and $\mathrm{Cl}\mathcal{O}$, respectively. One does know that there is an exact sequence

$$0 \to \ker\phi \to \mathcal{O}^* \to K_S/H \to \mathrm{coker}\,\phi \to \mathrm{Cl}\mathcal{O} \to (\mathrm{Cl}\mathcal{O})/C_S \to 0,$$

where C_S is the subgroup of $\mathrm{Cl}\mathcal{O}$ generated by the ideal classes of the finite

primes in S. The sequence shows that H has finite index in K_S if and only if the conjectured class group $\operatorname{coker}\phi$ is finite and the $\mathbf{Z}$-rank of the conjectured unit group $\ker\phi$ mod torsion is the same as it is for the true unit group $\mathscr{O}^*$, namely $\#S_\infty - 1$. If H has infinite index in K_S one should of course continue drawing elements of K_S.

The information that one has about the relation between the conjectured class group $\operatorname{coker}\phi$ and the true class group $\mathrm{Cl}\mathscr{O}$ is particularly meagre: one has a group homomorphism $\operatorname{coker}\phi \to \mathrm{Cl}\mathscr{O}$, but neither its injectivity nor its surjectivity is known. It is surjective if and only if the ideal classes of the finite primes in S generate the class group, and results of this nature are known only if the bound b that defines S is at least $|\Delta|^{1/2}$ times a constant depending on n. However, a significant improvement is possible if one makes an unproved assumption. Namely, Bach [6, Theorem 4] showed that if the generalized Riemann hypothesis holds, then $\mathrm{Cl}\mathscr{O}$ is generated by the ideal classes of the prime ideals of norm at most $12(\log|\Delta|)^2$. Hence if we assume the generalized Riemann hypothesis then the map $\operatorname{coker}\phi \to \mathrm{Cl}\mathscr{O}$ is surjective for values of b that are much smaller than $|\Delta|^{1/2}$. If the map is surjective, then the above exact sequence shows that

$$h'R' = hR \cdot [K_S : H], \tag{5.7}$$

where $h = \#\mathrm{Cl}\mathscr{O}$ and $R = \operatorname{reg}\mathscr{O}^*$ are the true class number and regulator, and $h' = \#\operatorname{coker}\phi$ and $R' = \operatorname{reg}\ker\phi$ the conjectured ones, here we assume that H contains all roots of unity in K, which can easily be accomplished [56, §5.4]. Now suppose that we are able to estimate hR up to a factor 2, i.e., that we can compute a number a with $a/2 < hR < a$; if one assumes the generalized Riemann hypothesis this can probably be done by means of a good algorithm, as in [16]. Then we see from (5.7) that $h'R'$ also satisfies $a/2 < h'R' < a$ if and only if $H = K_S$, and if and only if one has both $\ker\phi = \mathscr{O}^*$ and $\operatorname{coker}\phi = \mathrm{Cl}\mathscr{O}$.

The above indicates that on the assumption of the generalized Riemann hypothesis it may be possible to find a much faster probabilistic algorithm for determining $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ than the algorithm of Theorem 5.5. This leads to the following problem

Problem 5.8. Assuming the truth of the generalized Riemann hypothesis, find a probabilistic algorithm for Problem 5 1 that, for fixed n, runs in expected time

$$\exp\bigl(O((\log|\Delta|)^{1/2}(\log\log|\Delta|)^{1/2})\bigr),$$

the O-constant depending on n.

Of course, one also wants to know how the running time depends on n, and which value can be taken for the O-constant. For imaginary quadratic fields Problem 5 8 has been solved [28] For a partial solution in the general case, see [13]

6 Explicit bounds

In the present section we prove a few explicit bounds on units and class numbers of algebraic number fields, including Theorem 5.4. Several proofs in this section are most naturally formulated in terms of ideles, as in [20, Chapter

II]. To stress the elementary character of the arguments I have chosen to use more classical language.

We denote by K an algebraic number field of degree n and discriminant Δ over $\mathbf{Q}$, and by r and s the number of real and complex places of K, respectively. We embed K in $K_{\mathbf{R}} = K \otimes_{\mathbf{Q}} \mathbf{R}$, which, as an $\mathbf{R}$-algebra, is isomorphic to $\mathbf{R}^r \times \mathbf{C}^s$. We choose such an isomorphism, so that each element $a \in K_{\mathbf{R}}$ has $r+s$ coordinates a_ι, of which the first r are real and the last s complex. We put $n_\iota = 1$ for $1 \leq \iota \leq r$ and $n_\iota = 2$ for $r+1 \leq \iota \leq r+s$. The *norm* $N\colon K_{\mathbf{R}} \to \mathbf{R}$ is defined by $Na = \prod_{\iota=1}^{r+s} |a_\iota|^{n_\iota}$.

Identifying each copy of $\mathbf{C}$ with $\mathbf{R}^2$ by mapping $x + y\iota$ to $(x+y, x-y)$ we obtain an identification of $K_{\mathbf{R}}$ with the n-dimensional Euclidean space $\mathbf{R}^n$. It is well known that this identification makes the ring of integers $\mathscr{O}$ of K into a lattice of determinant $|\Delta|^{1/2}$ in $K_{\mathbf{R}}$, and more generally every fractional $\mathscr{O}$-ideal $\mathfrak{a}$ into a lattice of determinant $\mathfrak{N}\mathfrak{a} \cdot |\Delta|^{1/2}$, where $\mathfrak{N}$ denotes the ideal norm. We shall write

$$d = (2/\pi)^s |\Delta|^{1/2}.$$

Let S be a set of places of K with $S_\infty \subset S$. By I_S we denote the group of fractional $\mathscr{O}$-ideals generated by the finite primes in S, and by K_S, as in §5, the group $\{a \in K^*\colon \mathscr{O}a \in I_S\}$. Denote by $\iota_S\colon K_S \to K_{\mathbf{R}}^* \times I_S$ the embedding defined by $\iota_S a = (a, \mathscr{O}a)$. We give $K_{\mathbf{R}}^* \times I_S$ the product topology, where I_S is discrete. For any compact set $B \subset K_{\mathbf{R}}^* \times I_S$ the set $B \cap \iota_S K_S$ consists of elements of bounded height and is therefore finite Hence $\iota_S K_S$ is discrete. Also, $\iota_S K_S$ is clearly contained in the subgroup V_S of $K_{\mathbf{R}}^* \times I_S$ consisting of those pairs $(a, \mathfrak{a})$ for which $Na = \mathfrak{N}\mathfrak{a}$.

Theorem 6.1. *Let K be an algebraic number field, and let S be a set of places of K containing S_∞ and containing all finite places $\mathfrak{p}$ with $\mathfrak{N}\mathfrak{p} \leq d$, with d as above Let V_S be as above, and denote by F_S the set of all elements $(b, \mathfrak{b}) \in V_S$ for which $\mathfrak{b} \subset \mathscr{O}$, $\mathfrak{N}\mathfrak{b} \leq d$, and $|b_\iota| \leq d^{1/n}$ for $1 \leq \iota \leq r+s$ Then F_S is a compact subset of V_S and $V_S = F_S \cdot \iota_S K_S$*

Proof The compactness of F_S follows easily from the definition of F_S and the fact that V_S is closed in $K_{\mathbf{R}} \times I_S$. To prove the last assertion, let $(a, \mathfrak{a}) \in V_S$. Then $a \cdot \mathfrak{a}^{-1}$ is a lattice of determinant $Na \cdot |\Delta|^{1/2} \cdot \mathfrak{N}\mathfrak{a}^{-1} = |\Delta|^{1/2}$ in $K_{\mathbf{R}}$. By Minkowski's lattice point theorem there exists a nonzero element $b \in a\mathfrak{a}^{-1}$ with all $|b_\iota| \leq d^{1/n}$. From $\mathscr{O}b \subset a\mathfrak{a}^{-1}$ it follows that $\mathscr{O}b = a\mathfrak{a}^{-1}\mathfrak{b}$ for some integral $\mathscr{O}$-ideal $\mathfrak{b}$ Comparing determinants we see that $Nb = \mathfrak{N}\mathfrak{b}$, so $\mathfrak{N}\mathfrak{b} \leq d$. This implies that $\mathfrak{b} \in I_S$, so we have $(b, \mathfrak{b}) \in F_S$. If we write $b = ac$ then c is a nonzero element of $\mathfrak{a}^{-1}$, so $c \in K^*$. Since we also have $\mathscr{O}c = \mathfrak{a}^{-1}\mathfrak{b} \in I_S$, we even have $c \in K_S$, so $(a, \mathfrak{a}) = (b, \mathfrak{b}) \cdot \iota_S c^{-1}$. This proves Theorem 6.1

It follows from Theorem 6 1 that $V_S/\iota_S K_S$ is *compact*, if S is as in the theorem This allows one to deduce the Dirichlet unit theorem and the finiteness of the class number. Namely, take for S the set of *all* places of K. From the exact sequence $0 \to V_{S_\infty} \to V_S \to I_S \to 0$ one obtains an exact sequence

$$0 \to V_{S_\infty}/\iota_{S_\infty}\mathscr{O}^* \to V_S/\iota_S K_S \to \mathrm{Cl}\mathscr{O} \to 0,$$

where $\mathscr{O}^*$ and $\mathrm{Cl}\mathscr{O}$ are as in §5 The map to $\mathrm{Cl}\mathscr{O}$ is continuous if the latter is given the discrete topology Thus the compactness of $V_S/\iota_S K_S$ implies that $V_{S_\infty}/\iota_{S_\infty}\mathscr{O}^*$ is compact, which is essentially a restatement of the Dirichlet unit

theorem, and that $\mathrm{Cl}\mathscr{O}$ is finite In the same way one proves that $V_S/\iota_S K_S$ is compact for *every* set S of primes containing S_∞, not just for those from Theorem 6 1

From the exact sequence and Theorem 6.1 we see that every element of $\mathrm{Cl}\mathscr{O}$ is the ideal class of an integral ideal $\mathfrak{b}$ of norm at most d This implies the last assertion of Theorem 5.4. It also follows that $d \geq 1$. The other assertion of Theorem 5.4 is a special case of the following theorem, in which the height H is as defined in §5

$$H(a) = \mathfrak{N}(\mathscr{O} + \mathscr{O}a)^{-1} \cdot \prod_{i=1}^{r+s} \max\{1, |a_i|\}^{n_i}.$$

Theorem 6.2. *Let K, S be as in Theorem 6.1, with S finite Write $m_S = \max\{\mathfrak{N}\mathfrak{p} \colon \mathfrak{p} \in S - S_\infty\}$ if $S \neq S_\infty$ and $m_S = 1$ if $S = S_\infty$ Then the group K_S is generated by the set of those $a \in K_S$ satisfying $H(a) \leq d m_S$ and also by the set of those $a \in \mathscr{O} \cap K_S$ satisfying $H(a) \leq d^2 m_S$*

For the proof we need a lemma from combinatorial group theory, as well as a topological analogue

Lemma 6.3. *Let G be a group, P a set of generators for G, and H a subgroup of G Let F be a subset of G such that $G = FH$ Then H is generated by its intersection with $F^{-1}PF = \{x^{-1}yz \;\; x, z \in F, y \in P\}$*

Proof. Replacing P by $P \cup P^{-1}$ we may assume that $P = P^{-1}$, and replacing F by a subset we may assume that the multiplication map $F \times H \to G$ is bijective. Let $J \subset H$ be the subgroup generated by $H \cap F^{-1}PF$. If $y \in P$, $z \in F$, then $yz = xh$ for some $x \in F$, $h \in H$, and then $h = x^{-1}yz \in H \cap F^{-1}PF \subset J$. This proves that $PF \subset FJ$, so $PFJ \subset FJ$. Hence the nonempty set FJ is stable under left multiplication by P, which by our assumptions on P implies that $FJ = G$ From $J \subset H$ and the bijectivity of $F \times H \to G$ we now obtain $J = H$. This proves Lemma 6.3.

Lemma 6.4. *Let G be a Hausdorff topological group, and denote by G_1 the connected component of the unit element 1 of G Let $P \subset G$ be a subset containing 1 such that G is generated by $P \cup G_1$ Let $H \subset G$ be a discrete subgroup, and let F be a compact subset of G such that $G = FH$ Then H is generated by its intersection with $F^{-1}PF$*

Proof The set $H \cap F^{-1}F$ lies in the discrete subgroup H, so $(G - H) \cup (H \cap F^{-1}F)$ is open, and it contains the compact set $F^{-1}F$ Hence it contains $F^{-1}UF$ for some open neighborhood U of 1 Intersecting with H we see that $H \cap F^{-1}F = H \cap F^{-1}UF$ The subgroup of G generated by U is open, so it contains G_1. Therefore G is generated by $P \cup U$ Applying Lemma 6 3 we find that H is generated by

$$\begin{aligned} H \cap (F^{-1}(P \cup U)F) &= (H \cap F^{-1}PF) \cup (H \cap F^{-1}UF) \\ &= (H \cap F^{-1}PF) \cup (H \cap F^{-1}F) = (H \cap F^{-1}PF), \end{aligned}$$

where in the last step we use that $1 \in P$ This proves Lemma 6 4.

To prove Theorem 6 2, we apply Lemma 6.4 to

$$G = V_S, \qquad H = \iota_S K_S, \qquad F = F_S,$$
$$P = \{x \in V_S \colon x^2 = 1\} \cup \{((\mathfrak{N}\mathfrak{p})^{1/n}, \mathfrak{p}) \colon \mathfrak{p} \in S - S_\infty\},$$

where F_S is as in Theorem 6.1 and where $(\mathfrak{N}\mathfrak{p})^{1/n}$ is viewed as an element of $K_{\mathbf{R}}^*$ via the natural inclusion $\mathbf{R}^* \subset K_{\mathbf{R}}^*$. Using Theorem 6.1 one readily verifies that the conditions of Lemma 6.4 are satisfied. Hence K_S is generated by the set of those elements $a \in K_S$ for which there exist $(b, \mathfrak{b})$, $(c, \mathfrak{c}) \in F_S$, and $(y, \mathfrak{y}) \in P$ such that

$$(a, \mathscr{O}a) = (b, \mathfrak{b})^{-1} \cdot (y, \mathfrak{y}) \cdot (c, \mathfrak{c}).$$

Then $\mathscr{O}a = \mathfrak{b}^{-1}\mathfrak{y}\mathfrak{c}$, so the denominator ideal $\operatorname{den} a$ of a divides $\mathfrak{b}$. For all $\iota \in \{1, 2, \ldots, r+s\}$ we have

$$|b_\iota| \le d^{1/n}, \qquad |y_\iota| \le m_S^{1/n}, \qquad |c_\iota| \le d^{1/n},$$

so for each subset $J \subset \{1, 2, \ldots, r+s\}$ we have

$$\prod_{\iota \in J} |c_\iota|^{n_\iota} \le d^{n_J/n} \qquad \text{where } n_J = \sum_{\iota \in J} n_\iota,$$

$$\prod_{\iota \in J} |b_\iota|^{-n_\iota} = \mathfrak{N}\mathfrak{b}^{-1} \cdot \prod_{\iota \notin J} |b_\iota|^{n_\iota} \le \mathfrak{N}\mathfrak{b}^{-1} \cdot d^{1-n_J/n},$$

$$\prod_{\iota \in J} |a_\iota|^{n_\iota} = \prod_{\iota \in J} |b_\iota|^{-n_\iota} |c_\iota|^{n_\iota} |y_\iota|^{n_\iota} \le \mathfrak{N}\mathfrak{b}^{-1} \cdot d \cdot m_S.$$

Choosing $J = \{\iota : |a_\iota| > 1\}$ we obtain

$$H(a) = \mathfrak{N}(\operatorname{den} a) \cdot \prod_{\iota \in J} |a_\iota|^{n_\iota} \le \mathfrak{N}\mathfrak{b} \cdot \mathfrak{N}\mathfrak{b}^{x-1} \cdot d \cdot m_S = d \cdot m_S.$$

This proves the first assertion of Theorem 6.2. To prove the second assertion, we use Minkowski's lattice point theorem to choose a nonzero element $b' \in \mathfrak{b}$ with $|b'_\iota| \le (d \cdot \mathfrak{N}\mathfrak{b})^{1/n}$ for all ι. Then $b'\mathfrak{b}^{-1}$ is an integral ideal of norm at most d, so $b' \in \mathscr{O} \cap K_S$. Also $b'a \in \mathscr{O} \cap K_S$, and we have

$$H(b') = \prod_\iota \max\{1, |b'_\iota|\}^{n_\iota} \le d \cdot \mathfrak{N}\mathfrak{b} \le d^2,$$

$$H(b'a) \le \prod_\iota \max\{1, |b'_\iota|\}^{n_\iota} \cdot \prod_\iota \max\{1, |a_\iota|\}^{n_\iota}$$

$$\le d \cdot \mathfrak{N}\mathfrak{b} \cdot \mathfrak{N}\mathfrak{b}^{-1} \cdot d \cdot m_S = d^2 m_S.$$

Since we can write $a = (b'a)/b'$, this proves the second assertion of Theorem 6.2.

Remark Theorem 6.2 is also valid if the bound $d^2 m_S$ is replaced by $\max\{d^2 m_{S'}, d m_S\}$, where $S' = S_\infty \cup \{\mathfrak{p} : \mathfrak{N}\mathfrak{p} \le d\}$. This is proved by applying Theorem 6.2 to S' and choosing a nonzero element of height at most $d \cdot \mathfrak{N}\mathfrak{p}$ in each prime $\mathfrak{p} \in S - S'$.

As a further application of Theorem 6.1, we deduce upper bounds for the class number $h = \#\mathrm{Cl}\mathscr{O}$ and for the product hR of the class number and the regulator $R = \operatorname{reg}\mathscr{O}^*$. The upper bound for hR resembles the upper bound that Siegel [68] proved using properties of the zeta function of K. For similar upper bounds, see [58].

Theorem 6.5. *Let K be an algebraic number field of degree n and discriminant Δ over $\mathbf{Q}$, and let s denote the number of complex places of K. Let $d = (2/\pi)^s|\Delta|^{1/2}$. Then the class number h and the regulator R of K satisfy*

$$h \leq d \cdot \frac{(n-1+\log d)^{n-1}}{(n-1)!},$$

$$hR \leq d \cdot \frac{(\log d)^{n-1-s} \cdot (n-1+\log d)^s}{(n-1)!}$$

Proof We saw above that every ideal class contains an integral ideal of norm at most d, so

$$h \leq \#\{\mathfrak{b} \subset \mathscr{O} : \mathfrak{N}\mathfrak{b} \leq d\}.$$

For each positive integer m, the number of $\mathscr{O}$-ideals of norm m is at most the number of vectors $x = (x_i)_{i=1}^n \in \mathbf{Z}_{>0}^n$ satisfying $\prod_i x_i = m$. One proves this by considering how rational primes can split in K. Thus we obtain

$$\#\{\mathfrak{b} \subset \mathscr{O} : \mathfrak{N}\mathfrak{b} \leq d\} \leq \#\left\{x \in \mathbf{Z}_{>0}^n : \prod x_i \leq d\right\}.$$

Replacing each x by the box $\prod_{i=1}^n (x_i - 1, x_i]$ we can estimate the right side by a volume:

$$\#\left\{x \in \mathbf{Z}_{>0}^n : \prod x_i \leq d\right\} \leq \operatorname{vol}\left\{x \in \mathbf{R}_{>0}^n : \prod \max\{1, x_i\} \leq d\right\}.$$

Writing $y_i = \log x_i$ we see that the volume is equal to $J(n, \log d)$, where generally for $n \in \mathbf{Z}_{>0}$, $\delta \in \mathbf{R}_{\geq 0}$ we put

$$J(n, \delta) = \int_{y \in \mathbf{R}^n,\ \sum_i \max\{0,\ y_i\} \leq \delta} \exp\left(\sum_i y_i\right) dy.$$

This integral is found to be

$$J(n, \delta) = e^\delta \cdot \sum_{i=0}^{n-1} \binom{n-1}{i} \frac{\delta^i}{i!}$$

$$\leq e^\delta \cdot \sum_{i=0}^{n-1} \binom{n-1}{i} \frac{(n-1)^{n-1-i}\delta^i}{(n-1)!} = e^\delta \cdot \frac{(n-1+\delta)^{n-1}}{(n-1)!}.$$

Putting $\delta = \log d$ we obtain the inequality for h.

For hR, we apply Theorem 6.1 with S equal to the set of all places of K. Let $u = \#S_\infty - 1 = n - 1 - s$, and define the group homomorphism $\lambda : V_S \to \mathbf{R}^u \times I_S$ by $\lambda(a, \mathfrak{a}) = ((n_i \log |a_i|)_{i=1}^u, \mathfrak{a})$. This is a surjective group homomorphism with a compact kernel, so $\lambda \iota_S K_S$ is discrete in $\mathbf{R}^u \times I_S$ with a compact quotient. From the definition of the regulator one derives that hR equals the volume of a fundamental domain for $\lambda \iota_S K_S$ in $\mathbf{R}^u \times I_S$. Hence Theorem 6.1 implies that $hR \leq \operatorname{vol} \lambda F_S$. For each nonzero $\mathscr{O}$-ideal $\mathfrak{b}$ with $\mathfrak{N}\mathfrak{b} \leq d$ we have, by an easy computation,

$$\operatorname{vol} \lambda\{(b, \mathfrak{b}) \in V_S : |b_i| \leq d^{1/n} \text{ for all } i\} = \frac{\left(\log(d/\mathfrak{N}\mathfrak{b})\right)^u}{u!}.$$

Therefore

$$hR \leq \sum_{\mathfrak{N}\mathfrak{b} \leq d} \frac{\left(\log(d/\mathfrak{N}\mathfrak{b})\right)^u}{u!},$$

where the sum is over integral $\mathcal{O}$-ideals $\mathfrak{b}$. Proceeding as with h one finds that this is bounded above by

$$\int_{y\in\mathbf{R}^n,\ \sum_i \max\{0,y_i\}\le\delta} \exp\Big(\sum_i y_i\Big)\cdot\frac{\big(\delta-\sum_i \max\{0,y_i\}\big)^u}{u!}\,dy,$$

with $\delta=\log d$. Using that $s=n-1-u\ge 0$ one finds after some computation the integral to be

$$e^\delta\cdot\sum_{i=0}^{s}\binom{s}{i}\frac{\delta^{u+i}}{(u+i)!}\le e^\delta\cdot\sum_{i=0}^{s}\binom{s}{i}\frac{(n-1)^{s-i}\delta^{u+i}}{(n-1)!}=e^\delta\cdot\frac{\delta^u\cdot(n-1+\delta)^s}{(n-1)!}.$$

This proves Theorem 6.5.

Remark. The upper bound for h in Theorem 6.5 is also valid when d, at both occurrences, is replaced by the Minkowski constant $d'=(n!/n^n)(4/\pi)^s|\Delta|^{1/2}$ of K, since every ideal class contains an integral ideal of norm at most d'.

Acknowledgments

The author gratefully acknowledges the hospitality and support of the Institute for Advanced Study (Princeton). I also thank Enrico Bombieri, Johannes Buchmann, Joe Buhler, Gary Cornell, Pierre Deligne, Bas Edixhoven, Boas Erez, David Ford, Marty Isaacs, Ravi Kannan, Bill Kantor, Susan Landau, Andries Lenstra, Arjen Lenstra, Kevin McCurley, John McKay, Andrew Odlyzko, Michael Pohst, Carl Pomerance, and Jeff Shallit for their assistance and helpful advice.

References

1 L M Adleman and M A Huang, *Recognizing primes in random polynomial time*, Research report, Dept of Computer Science, Univ of Southern California, 1988, Lecture Notes in Math, Springer, Heidelberg (to appear) Extended abstract Proc 19th Ann ACM Sympos on Theory of Computing (STOC), ACM, New York 1987, pp 462–469

2 L M Adleman and H W Lenstra, Jr, *Finding irreducible polynomials over finite fields*, Proc 18th Ann ACM Sympos on Theory of Computing (STOC), ACM, New York (1986, pp 350–355

3 L M Adleman, C Pomerance, and R S Rumely, *On distinguishing prime numbers from composite numbers*, Ann of Math (2) **117** (1983), 173–206

4 Archimedes, *The sand reckoner*, in Opera quae quidem extant J Hervagius, Basel, 1544 (Greek and Latin)

5 A O L Atkin and F Morain, *Elliptic curves and primality proving* (to appear)

6 E Bach, *Explicit bounds for primality testing and related problems*, Math Comp **55** (1990), 355-380

7 E Bach and J O Shallit, *Factor refinement*, J Algorithms (to appear)

8 W E H Berwick, *Integral bases*, Cambridge Univ Press Cambridge, 1927

9 Z I Borevič and I R Šafarevic, *Teoriya chisel*, Izdat "Nauka", Moscow, 1964 English transl *Number theory*, Academic Press, New York 1966

10 W Bosma and M P M van der Hulst, *Primality proving with cyclotomy* Academisch proefschrift, Universiteit van Amsterdam, 1990

11 E Brieskorn and H Knorrer *Ebene algebraische Kurven*, Birkhauser Basel, 1981

12 J Buchmann, *Complexity of algorithms in algebraic number theory*, Proceedings of the first conference of the Canadian Number Theory Association (R A Mollin, ed), De Gruyter, Berlin, 1990, pp 37–53

13 ——— *A subexponential algorithm for the determination of class groups and regulators of algebraic number fields*, Seminaire de Theorie des Nombres, Paris 1988–1989 (C Goldstein, ed), Birkhauser, Boston, 1990 pp 27–41

14 J Buchmann and H W Lenstra Jr Manuscript in preparation

15 J Buchmann and V Shoup, *Constructing nonresidues in finite fields and the extended Riemann hypothesis*, in preparation Extended abstract Proc 23rd Ann ACM Sympos on Theory of Computing (STOC) ACM, New York 1991 pp 72–79

16 J Buchmann and H C Williams, *On the computation of the class number of an algebraic number field*, Math Comp **53** (1989), 679–688

17 J P Buhler H W Lenstra, Jr , and C Pomerance, *Factoring integers with the number field sieve*, in preparation

18 P J Cameron, *Finite permutation groups and finite simple groups*, Bull London Math Soc **13** (1981), 1–22

19 J W S Cassels, *Local fields*, Cambridge Univ Press, Cambridge, 1986

20 J W S Cassels and A Frohlich (eds), *Algebraic number theory*, Proceedings of an instructional conference, Academic Press, London, 1967

21 A L Chistov, *Efficient factorization of polynomials over local fields*, Dokl Akad Nauk SSSR **293** (1987), 1073–1077, English transl Soviet Math Dokl **35** (1987), 430–433

22 ——— *The complexity of constructing the ring of integers of a global field*, Dokl Akad Nauk SSSR **306** (1989), 1063–1067, English transl Soviet Math Dokl **39** (1989), 597–600

23 H Cohen, *A course in computational algebraic number theory*, in preparation

24 H Cohen and A K Lenstra, *Implementation of a new primality test*, Math Comp **48** (1987), 103–121

25 H Cohen and H W Lenstra, Jr , *Primality testing and Jacobi sums*, Math Comp **42** (1984), 297–330

26 D F Ford and J McKay, *Computation of Galois groups from polynomials over the rationals*, Computer Algebra (D V Chudnovsky and R D Jenks, eds), Lecture Notes in Pure and Appl Math , vol 113, Marcel Dekker, New York, 1989, pp 145–150

27 D Gordon, *Discrete logarithms using the number field sieve* (to appear)

28 J L Hafner and K S McCurley, *A rigorous subexponential algorithm for computation of class groups*, J Amer Math Soc **2** (1989), 837–850

29 ———, *Asymptotically fast triangularization of matrices over rings*, SIAM J Comput (to appear)

30 R Kannan, *Algorithmic geometry of numbers*, Annual Review of Computer Sciences, vol 2 (J F Traub, B J Grosz, B W Lampson, N J Nilsson, eds), Annual Reviews Inc , Palo Alto, 1987, pp 231–267

31 W M Kantor, unpublished

32 W M Kantor and E M Luks, *Computing in quotient groups*, Proc 22nd Ann ACM Sympos on Theory of Computing (STOC), ACM, New York 1990, pp 524–534

33 D E Knuth, *The art of computer programming*, Vol 2, Seminumerical Algorithms, Addison-Wesley, Reading, MA, second edition, 1981

34 S Landau, *Polynomial time algorithms for Galois groups*, Eurosam 84 (J Fitch, ed), Lecture Notes in Comput Sci , vol 174 Springer, Berlin, 1984, pp 225–236

35 ———, *Factoring polynomials over algebraic number fields*, SIAM J Comput **14** (1985) 184–195

36 S Landau and G L Miller, *Solvability by radicals is in polynomial time*, J Comput System Sci **30** (1985), 179–208

37 S Lang, *Algebraic number theory*, Addison-Wesley, Reading, MA, 1970

38 A K Lenstra, *Factorization of polynomials*, Computational Methods in Number Theory (H W Lenstra, Jr and R Tijdeman, eds), Math Centre Tracts , vol 154/155, Mathematisch Centrum Amsterdam, 1982, pp 169–198

39 ——, *Factoring polynomials over algebraic number fields* Computer Algebra (J A van Hulzen, ed) Lecture Notes in Comput Sci vol 162, Springer, Berlin, 1983, pp 245–254

40 ——, *Factoring multivariate polynomials over algebraic number fields*, SIAM J Comput **16** (1987), 591–598

41 A K Lenstra and H W Lenstra, Jr , *Algorithms in number theory*, Handbook of Theoretical Computer Science, Vol A, Algorithms and Complexity (J van Leeuwen, ed), Elsevier, Amsterdam, 1990, pp 673–715

42 A K Lenstra, H W Lenstra, Jr , and L Lovasz, *Factoring polynomials with rational coefficients*, Math Ann **261** (1982), 515–534

43 A K Lenstra, H W Lenstra, Jr , M S Manasse, and J M Pollard, *The factorization of the ninth Fermat number* (to appear)

44 ——, *The number field sieve*, in preparation Extended abstract Proc 22nd Ann ACM Sympos on Theory of Computing (STOC), ACM, New York 1990, pp 564–572

45 H W Lenstra, Jr , *On the calculation of regulators and class numbers of quadratic fields*, Journees Arithmetiques 1980 (J Armitage, ed), London Math Soc Lecture Note Ser , vol 56, Cambridge Univ Press, Cambridge, 1982, pp 123–150

46 ——, *Galois theory and primality testing*, Orders and Their Applications (I Reiner, K Roggenkamp, eds), Lecture Notes in Math , vol 1142, Springer, Heidelberg, 1985, pp 169–189

47 ——, *Algorithms for finite fields*, Number Theory and Cryptography (J H Loxton, ed), London Math Soc Lecture Note Ser , vol 154, Cambridge Univ Press, Cambridge, 1990, pp 76–85

48 ——, *Finding isomorphisms between finite fields*, Math Comp **56** (1991), 329–347

49 H W Lenstra, Jr and C Pomerance, *A rigorous time bound for factoring integers*, J Amer Math Soc (to appear)

50 H W Lenstra, Jr and R Tijdeman (eds), *Computational methods in number theory*, Mathematical Centre Tracts, vol 154/155, Mathematisch Centrum, Amsterdam, 1982

51 R Lovorn, *Rigorous, subexponential algorithms for discrete logarithms over finite fields*, thesis, University of Georgia, in preparation

52 A McIver and P M Neumann, *Enumerating finite groups*, Quart J Math Oxford Ser (2) **38** (1987), 473–488

53 A M Odlyzko, *Discrete logarithms in finite fields and their cryptographic significance*, Advances in Cryptology (T Beth, N Cot, and I Ingemarsson, eds), Lecture Notes in Comput Sci , vol 209, Springer, Berlin, 1985, pp 224–314

54 P P Palfy, *A polynomial bound for the orders of primitive solvable groups*, J Algebra 77 (1982), 127–137

55 M E Pohst, *Three principal tasks of computational algebraic number theory*, Number Theory and Applications (R A Mollin, ed), Kluwer, Dordrecht, 1989, pp 123–133

56 M Pohst and H Zassenhaus, *Algorithmic algebraic number theory*, Cambridge Univ Press, Cambridge, 1989

57 C Pomerance, *Fast rigorous factorization and discrete logarithm algorithms*, Discrete Algorithms and Complexity (D S Johnson, T Nishizeki, A Nozaki, and H S Wilf, eds), Academic Press, Orlando, 1987, pp 119–143

58 R Queme, *Relations d'inegalites effectives en theorie algebrique des nombres*, Sem Theor Nombres Bordeaux, 1987–1988, 19-01–19-19

59 J W Sands, *Generalization of a theorem of Siegel*, Acta Arith **58** (1991), 47–57

60 O Schirokauer, *Discrete logarithms and local units*, in preparation

61 R J Schoof, *Quadratic fields and factorization*, Computational Methods in Number Theory (H W Lenstra, Jr and R Tijdeman, eds), Math Centre Tracts , vol 154/155, Mathematisch Centrum Amsterdam, 1982, pp 235–286

62 ——, *Elliptic curves over finite fields and the computation of square roots* mod p , Math Comp **44** (1985), 483–494

63 J-P Serre, *Arbres amalgames* SL_2 , Asterisque **46** (1977)

64. ——, *Lectures on the Mordell-Weil theorem*, Vieweg, Braunschweig, 1989.

65. D. Shanks, *The infrastructure of a real quadratic field and its applications*, Proceedings of the 1972 number theory conference, Univ. of Colorado, Boulder, CO, 1972, pp. 217–224.

66. V. Shoup, *New algorithms for finding irreducible polynomials over finite fields*, Math. Comp. **54** (1990), 435–447.

67. ——, *On the deterministic complexity of factoring polynomials over finite fields*, Inform. Process. Lett. **33** (1990), 261–267.

68. C. L. Siegel, *Abschatzung von Einheiten*, Nachr. Akad. Wiss. Gottingen Math.-Phys. Kl. 1969, pp. 71–86; *Gesammelte Abhandlungen*, Band IV, Springer, Berlin, 1979, pp. 66–81.

69. R. P. Stauduhar, *The determination of Galois groups*, Math. Comp. **27** (1973), 981–996.

70. L. Szpiro, *Présentation de la théorie d'Arakélov*, Current Trends in Arithmetical Algebraic Geometry (K. A. Ribet, ed.), Contemp. Math., vol. 67, Amer. Math. Soc., Providence, RI, 1987, pp. 279–293.

71. J. Teitelbaum, *The computational complexity of the resolution of plane curve singularities*, Math. Comp. **54** (1990), 797–837.

72. N. Tzanakis and B. M. M. de Weger, *How to solve explicitly a Thue-Mahler equation* (to appear).

73. F. J. van der Linden, *The computation of Galois groups*, Computational Methods in Number Theory (H. W. Lenstra, Jr. and R. Tijdeman, eds.), Math. Centre Tracts., vol. 154/155, Mathematisch Centrum Amsterdam, 1982, pp. 199–211.

74. P. van Emde Boas, *Machine models, computational complexity and number theory*, Computational Methods in Number Theory (H. W. Lenstra, Jr. and R. Tijdeman, eds.), Math. Centre Tracts., vol. 154/155, Mathematisch Centrum Amsterdam, 1982, pp. 7–42.

75. E. Weiss, *Algebraic number theory*, McGraw-Hill, New York 1963, reprinted, Chelsea, New York, 1976.

76. H. Zantema, *Class numbers and units*, Computational Methods in Number Theory (H. W. Lenstra, Jr. and R. Tijdeman, eds.), Math. Centre Tracts., vol. 154/155, Mathematisch Centrum Amsterdam, 1982, pp. 212–234.

77. H. Zassenhaus, *Ein Algorithmus zur Berechnung einer Minimalbasis uber gegebener Ordnung*, Funktionalanalysis, Approximationstheorie, numerische Mathematik, Oberwolfach 1965 (L. Collatz, G. Meinardus, and H. Unger, eds.), Birkhauser, Basel, 1967, pp. 90–103.

78. H. G. Zimmer, *Computational problems, methods and results in algebraic number theory*, Lecture Notes in Math., vol. 262, Springer, Berlin, 1972.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA 94720