A Primer on Algebra and Number Theory for Computer Scientists (version 0.1)

Victor Shoup[1]

September 6, 2002

[1]Courant Institute, New York University, shoup@cs.nyu.edu

Preface

This is a very preliminary version of some notes that are intended to introduce computer science students to elementary notions of algebra and number theory, especially those notions that are relevant to the study of cryptography. These notes are meant to be quite self contained — the only prerequisites are some programming experience, a little knowledge of probabilities, and, most importantly, the ability to read and write mathematical proofs (or at least the willingness and ability to learn to do so).

Many proofs of theorems that are relatively straightforward applications of definitions and other theorems are left as exercises for the reader. There are also many examples that the reader can verify as well. Other than these, there are no "formal" exercises in these notes.

As stated above, these notes are in a very preliminary form. They certainly need more proof reading and polishing, and I may eventually want to add some new material as well. Also, there are currently no bibliographic references, which is a serious problem that I will have to correct very soon.

If you find any typos or more serious problems, please let me know, so that I can correct the problem in a future revision.

Remarks on Notation

- $\log(x)$ denotes the natural logarithm of x.

- For any function f from a set A into a set B, if $A' \subset A$, then $f(A') := \{f(a) \in B : a \in A'\}$. For $b \in B$, $f^{-1}(b) := \{a \in A : f(a) = b\}$, and more generally, for $B' \subset B$, $f^{-1}(B') := \{a \in A : f(a) \in B'\}$.

  f is called **one to one** or **injective** if $f(a) = f(b)$ implies $a = b$. f is called **onto** or **surjective** if $f(A) = B$. f is called **bijective** if it is both injective and surjective; in this case, f is called a **bijection**.

- Suppose f and g are functions from either the non-negative integers or non-negative reals into the non-negative reals. More generally, we may allow the domain of definition of f and g to consist all off integers or reals above some fixed bound. Then we write $f = O(g)$ if there exist constants $c > 0$ and $x_0 \geq 0$ such that $f(x) \leq cg(x)$ for all $x \geq x_0$. We write $f = \Omega(g)$ if $g = O(f)$, and we write $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$. We write $f \sim g$ if $\lim_{x\to\infty} f(x)/g(x) = 1$, and we write $f = o(g)$ if $\lim_{x\to\infty} f(x)/g(x) = 0$.

  One also may write $O(g)$ in an expression to denote an implicit function f such that $f = O(g)$. For example, one may write $(n+1)(n+2) = n^2 + O(n)$. Similarly for $\Omega(g)$, $\Theta(g)$, and $o(g)$. Note that $f \sim g$ is equivalent to $f = g(1 + o(1))$.

Contents

Chapter 1

Basic Properties of the Integers

This chapter reviews some of the basic properties of the integers, including notions of divisibility and primality, unique factorization into primes, greatest common divisors, and least common multiples.

1.1 Divisibility and Primality

Consider the integers $\mathbb{Z} = \{\ldots, -1, 0, 1, 2, \ldots\}$. For $a, b \in \mathbb{Z}$, we say that b **divides** a, and write $b \mid a$, if there exists $c \in \mathbb{Z}$ such that $a = bc$. If $b \mid a$, then b is called a **divisor** of a. If b does not divide a, then we write $b \nmid a$.

We first state some simple facts:

Theorem 1.1 *For all $a, b, c \in \mathbb{Z}$, we have*

1. $a \mid a$, $1 \mid a$, *and* $a \mid 0$;
2. $0 \mid a$ *if and only if* $a = 0$;
3. $a \mid b$ *and* $b \mid c$ *implies* $a \mid c$;
4. $a \mid b$ *implies* $a \mid bc$;
5. $a \mid b$ *and* $a \mid c$ *implies* $a \mid b + c$;
6. $a \mid b$ *and* $b \mid a$ *if and only if* $a = \pm b$.

Proof. Exercise. □

We say that an integer p is **prime** if $p > 1$ and the only divisors of p are ± 1 and $\pm p$. Conversely, and integer n is called **composite** if $n > 1$ and it is not prime. So an integer $n > 1$ is composite if and only if $n = ab$ for some integers a, b with $1 < a, b < n$.

A fundamental fact is that any integer can be written as a signed product of primes in an essentially unique way. More precisely:

Theorem 1.2 *Every non-zero integer n can be expressed as*

$$n = \pm \prod_p p^{\nu_p(n)},$$

where the product is over all primes, and all but a finite number of the exponents are zero. Moreover, the exponents and sign are uniquely determined by n.

To do prove this theorem, we may clearly assume that n is positive.

The proof of the existence part of Theorem 1.2 is easy. If n is 1 or prime, we are done; otherwise, there exist $a, b \in \mathbb{Z}$ with $1 < a, b < n$ and $n = ab$, and we apply an inductive argument with a and b.

The proof of the uniqueness part of Theorem 1.2 is not so simple, and most of the rest of this chapter is devoted to developing the ideas behind such a proof. The essential ingredient in the proof is the following:

Theorem 1.3 (Division with Remainder Property) *For $a, b \in \mathbb{Z}$ with $b > 0$, there exist unique $q, r \in \mathbb{Z}$ such that $a = bq + r$ and $0 \leq r < b$.*

Proof. Consider the set S of non-negative integers of the form $a - xb$ with $x \in \mathbb{Z}$. This set is clearly non-empty, and so contains a minimum. Let $r = a - qb$ be the smallest integer in this set. By definition, we have $r \geq 0$. Also, we must have $r < b$, since otherwise, we would have $r - b \in S$, contradicting the minimality of r.

That proves the existence of r and q. For uniqueness, suppose that $a = bq + r$ and $a = bq' + r'$, where $0 \leq r, r' < b$. Then subtracting these two equations and rearranging terms, we obtain

$$r' - r = b(q - q'). \tag{1.1}$$

Now observe that by assumption, the left-hand side of (1.1) is less than b in absolute value. However, if $q \neq q'$, then the right-hand side of (1.1) would be at least b in absolute value; therefore, we must have $q = q'$. But then by (1.1), we must have $r = r'$. □

In the above theorem, it is easy to see that $q = \lfloor a/b \rfloor$, the greatest integer less than or equal to a/b. We shall write $r = a \text{ rem } b$. For $a \in \mathbb{Z}$ and a positive integer b, it is clear that $b \mid a$ if and only if $a \text{ rem } b = 0$.

1.2 Ideals and Greatest Common Divisors

To carry on with the proof of Theorem 1.2, we introduce the notion of an **ideal** in $\mathbb{Z}$, which is a non-empty set of integers that is closed under addition and subtraction, and closed under multiplication by integers. That is, a non-empty set $I \subset \mathbb{Z}$ is an ideal if and only if for all $a, b \in I$ and all $z \in \mathbb{Z}$, we have

$$a + b \in I, \quad a - b \in I, \quad \text{and} \quad az \in I.$$

Note that in fact closure under addition and subtraction already implies closure under multiplication by integers, and so the definition is a bit redundant.

For $a_1, \ldots, a_k \in \mathbb{Z}$, define

$$a_1\mathbb{Z} + \cdots + a_n\mathbb{Z} := \{a_1 z_1 + \cdots a_k z_k : z_1, \ldots, z_k \in \mathbb{Z}\}.$$

We leave it to the reader to verify that $a_1\mathbb{Z} + \cdots + a_n\mathbb{Z}$ is an ideal, and this ideal clearly contains $a_1, \ldots, a_k$. An ideal of the form $a\mathbb{Z}$ is called a **principal ideal**.

Theorem 1.4 *For any ideal $I \subset \mathbb{Z}$, there exists a unique non-negative integer d such that $I = d\mathbb{Z}$.*

Proof. We first prove the existence part of the theorem. If $I = \{0\}$, then $d = 0$ does the job, so let us assume that $I \neq \{0\}$. Since I contains non-zero integers, it must contain positive integers, since if $x \in I$ then so is $-x$. Let d be the smallest positive integer in I. We want to show that $I = d\mathbb{Z}$.

We first show that $I \subset d\mathbb{Z}$. To this end, let c be any element in I. It suffices to show that $d \mid c$. Using the Division with Remainder Property, write $c = qd + r$, where $0 \leq r < d$. Then by the closure properties of ideals, one sees that $r = c - qd$ is also an element of I, and by the minimality of the choice of d, we must have $r = 0$. Thus, $d \mid c$.

We next show that $d\mathbb{Z} \subset I$. This follows immediately from the fact that $d \in I$ and the closure properties of ideals.

That proves the existence part of the theorem. As for uniqueness, note that if $d\mathbb{Z} = d'\mathbb{Z}$, we have $d \mid d'$ and $d' \mid d$, from which it follows that $d' = \pm d$. □

For $a, b \in \mathbb{Z}$, we call $d \in \mathbb{Z}$ a **common divisor** of a and b if $d \mid a$ and $d \mid b$; moreover, we call d the **greatest common divisor** of a and b if d is non-negative and all other common divisors of a and b divide d. It is immediate from the definition of a greatest common divisor that it is unique if it exists at all.

Theorem 1.5 *For any $a, b \in \mathbb{Z}$, there exists a greatest common divisor d of a and b, and moreover, $a\mathbb{Z} + b\mathbb{Z} = d\mathbb{Z}$; in particular, $as + bt = d$ for some $s, t \in \mathbb{Z}$.*

Proof. We apply the previous theorem to the ideal $I = a\mathbb{Z} + b\mathbb{Z}$. Let $d \in \mathbb{Z}$ with $I = d\mathbb{Z}$, as in that theorem. Note that $a, b, d \in I$.

Since $a \in I = d\mathbb{Z}$, we see that $d \mid a$; similarly, $d \mid b$. So we see that d is a common divisor of a and b.

Since $d \in I = a\mathbb{Z} + b\mathbb{Z}$, there exist $s, t \in \mathbb{Z}$ such that $as + bt = d$. Now suppose $a = a'd'$ and $b = b'd'$ for $a', b', d' \in \mathbb{Z}$. Then the equation $as + bt = d$ implies that $d'(a's + b't) = d$, which says that $d' \mid d$. Thus, d is the greatest common divisor of a and b. □

For $a, b \in \mathbb{Z}$, we denote by $\gcd(a, b)$ the greatest common divisor of a and b.

We say that a and b are **relatively prime** if $\gcd(a, b) = 1$. Notice that a and b are relatively prime if and only if $a\mathbb{Z} + b\mathbb{Z} = \mathbb{Z}$, i.e., if and only if there exist $s, t \in \mathbb{Z}$ such that $as + bt = 1$.

Theorem 1.6 *For $a, b, c \in \mathbb{Z}$ such that $c \mid ab$ and $\gcd(a, c) = 1$, we have $c \mid b$.*

Proof. Suppose that $c \mid ab$ and $\gcd(a, c) = 1$. Then since $\gcd(a, c) = 1$, by Theorem 1.5 we have $as + ct = 1$ for some $s, t \in \mathbb{Z}$. Multiplying this equation by b, we obtain

$$abs + cbt = b. \tag{1.2}$$

Since c divides ab by hypothesis, and since c clearly divides cbt, it follows that c divides the left-hand side of (1.2), and hence that c divides b. □

As a consequence of this theorem, we have:

Theorem 1.7 *Let p be prime, and let $a, b \in \mathbb{Z}$. Then $p \mid ab$ implies that $p \mid a$ or $p \mid b$.*

Proof. The only divisors of p are ± 1 and $\pm p$. Thus, $\gcd(p, a)$ is either 1 or p. If $p \mid a$, we are done; otherwise, if $p \nmid a$, we must have $\gcd(p, a) = 1$, and by the previous theorem, we conclude that $p \mid b$. □

1.3 Finishing the Proof of Theorem 1.2

Theorem 1.7 is the key to proving the uniqueness part of Theorem 1.2. Indeed, suppose we have

$$p_1 \cdots p_r = p_1' \cdots p_s',$$

where the p_i and p_i' are primes (duplicates are allowed among the p_i and among the p_i'). If $r = 0$, we must have $s = 0$ and we are done. Otherwise, as p_1 divides the right-hand side, by inductively applying Theorem 1.7, one sees that p_1 is equal to some p_i'. We can cancel these terms and proceed inductively (on r). That proves the uniqueness part of Theorem 1.2.

1.4 Further Observations

For non-zero integers a and b, it is easy to see that

$$\gcd(a, b) = \prod_p p^{\min(\nu_p(a), \nu_p(b))},$$

where the function $\nu_p(\cdot)$ is as implicitly defined in Theorem 1.2.

For $a, b \in \mathbb{Z}$ a **common multiple** of a and b is an integer m such that $a \mid m$ and $b \mid m$; moreover, m is a **least common multiple** of a and b if m is non-negative and m divides all common multiples of a and b. In light of Theorem 1.2, it is clear that the least common multiple exists and is unique; indeed, if we denote the least common multiple of a and b as $\operatorname{lcm}(a, b)$, then for non-zero integers a and b, we have

$$\operatorname{lcm}(a, b) = \prod_p p^{\max(\nu_p(a), \nu_p(b))}.$$

Moreover, for all $a, b \in \mathbb{Z}$, we have

$$\gcd(a, b) \cdot \operatorname{lcm}(a, b) = ab.$$

Finally, we recall the basic fact that that there in infinitely many primes. For a proof of this, suppose that there were only finitely many primes, call them $p_1, \ldots, p_k$. Then set $x = 1 + \prod_{i=1}^k p_i$, and consider any prime p that divides x. Clearly, p cannot equal any of the p_i, since if it did, we would would have $p \mid 1$, which is impossible. Therefore, the prime p is not among $p_1, \ldots, p_k$, which contradicts our assumption that these are the only primes.

Chapter 2

Congruences

This chapter reviews the notion of congruences.

2.1 Definitions and Basic Properties

For positive integer n and for $a, b \in \mathbb{Z}$, we say that a **is congruent to** b **modulo** n if $n \mid (a - b)$, and we write $a \equiv b \pmod{n}$. If $n \nmid (a - b)$, then we write $a \not\equiv b \pmod{n}$. The number n appearing in such congruences is called the **modulus**.

A trivial observation is that $a \equiv b \pmod{n}$ if and only if there exists an integer c such that $a = b + cn$. Another trivial observation is that if $a \equiv b \pmod{n}$ and $n' \mid n$, then $a \equiv b \pmod{n'}$.

A key property of congruences is that they are "compatible" with integer addition and multiplication, in the following sense:

Theorem 2.1 *For all positive integers n, and all $a, a', b, b' \in \mathbb{Z}$, if $a \equiv a' \pmod{n}$ and $b \equiv b' \pmod{n}$, then*

$$a + b \equiv a' + b' \pmod{n}$$

and

$$a \cdot b \equiv a' \cdot b' \pmod{n}.$$

Proof. Suppose that $a \equiv a' \pmod{n}$ and $b \equiv b' \pmod{n}$. This means that there exist integers c and d such that $a' = a + cn$ and $b' = b + dn$. Therefore,

$$a' + b' = a + b + (c + d)n,$$

which proves the first equality of the theorem, and

$$a'b' = (a + cn)(b + dn) = ab + (ad + bc + cdn)n,$$

which proves the second equality. □

2.2 Solving Linear Congruences

For a positive integer n, and $a \in \mathbb{Z}$, we say that a is a **unit modulo** n if there exists $a' \in \mathbb{Z}$ such that $aa' \equiv 1 \pmod{n}$, in which case we say that a' is a **multiplicative inverse of** a **modulo** n.

Theorem 2.2 *An integer a is a unit modulo n if and only if a and n are relatively prime.*

Proof. This follows immediately from the fact that a and n are relatively prime if and only if there exist $s, t \in \mathbb{Z}$ such that $as + bt = 1$. $\square$

We now prove a simple a "cancellation law" for congruences:

Theorem 2.3 *If a is relatively prime to n, then $ax \equiv ax' \pmod{n}$ if and only if $x \equiv x' \pmod{n}$. More generally, if $d = \gcd(a, n)$, then $ax \equiv ax' \pmod{n}$ if and only if $x \equiv x' \pmod{n/d}$.*

Proof. For the first statement, assume that $\gcd(a, n) = 1$, and let a' be a multiplicative inverse of a modulo n. Then, $ax \equiv ax' \pmod{n}$ implies $a'ax \equiv a'ax' \pmod{n}$, which implies $x \equiv x' \pmod{n}$, since $a'a \equiv 1 \pmod{n}$. Conversely, if $x \equiv x' \pmod{n}$, then trivially $ax \equiv ax' \pmod{n}$. That proves the first statement.

For the second statement, let $d = \gcd(a, n)$. Simply from the definition of congruences, one sees that in general, $ax \equiv ax' \pmod{n}$ holds if and only if $(a/d)x \equiv (a/d)x' \pmod{n/d}$. Moreover, since a/d and n/d are relatively prime, the first statement of the theorem implies that $(a/d)x \equiv (a/d)x' \pmod{n}$ holds if and only if $x \equiv x' \pmod{n/d}$. That proves the second statement. $\square$

We next look at solutions x to congruences of the form $ax \equiv b \pmod{n}$, for given integers n, a, b.

Theorem 2.4 *Let n be a positive integer and let $a, b \in \mathbb{Z}$. If a is relatively prime to n, then the congruence $ax \equiv b \pmod{n}$ has a solution x; moreover, any integer x' is a solution if and only if $x \equiv x' \pmod{n}$.*

Proof. The integer $x = ba'$, where a' is a multiplicative inverse of a modulo n, is clearly a solution. For any integer x', we have $ax' \equiv b \pmod{n}$ if and only if $ax' \equiv ax \pmod{n}$, which by Theorem 2.3 holds if and only if $x \equiv x' \pmod{n}$. $\square$

In particular, this theorem implies that multiplicative inverses are uniquely determined modulo n.

More generally, we have:

Theorem 2.5 *Let n be a positive integer and let $a, b \in \mathbb{Z}$. Let $d = \gcd(a, n)$. If $d \mid b$, then the congruence $ax \equiv b \pmod{n}$ has a solution x, and any integer x' is also a solution if and only if $x \equiv x' \pmod{n/d}$. If $d \nmid b$, then the congruence $ax \equiv b \pmod{n}$ has no solution x.*

Proof. Let n, a, b, d be as defined above.

For the first statement, suppose that $d \mid b$. In this case, by Theorem 2.3, we have $ax \equiv b \pmod{n}$ if and only if $(a/d)x \equiv (b/d) \pmod{n/d}$, and so the statement follows immediately from Theorem 2.4.

For the second statement, assume that $ax \equiv b \pmod{n}$ for some integer x. Then since $d \mid n$, we have $ax \equiv b \pmod{d}$. However, $ax \equiv 0 \pmod{d}$, since $d \mid a$, and hence $b \equiv 0 \pmod{d}$, i.e., $d \mid b$. $\square$

Next, we consider systems of congruences with respect to moduli that that are relatively prime in pairs. The result we state here is known as the Chinese Remainder Theorem, and is extremely useful in a number of contexts.

Theorem 2.6 (Chinese Remainder Theorem) *Let $k > 0$, and let $a_1, \ldots, a_k \in \mathbb{Z}$, and let $n_1, \ldots, n_k$ be positive integers such that $\gcd(n_i, n_j) = 1$ for all $1 \le i < j \le k$. Then there exists an integer x such that*

$$x \equiv a_i \pmod{n_i} \quad (i = 1, \ldots, k).$$

Moreover, any other integer x' is also a solution of these congruences if and only if $x \equiv x' \pmod{n}$, where $n := \prod_{i=1}^k n_i$.

Proof. Let $n := \prod_{i=1}^k n_i$, as in the statement of the theorem. Let us also define

$$n_i' := n/n_i \quad (i = 1, \ldots, k).$$

It is clear that $\gcd(n_i, n_i') = 1$ for $1 \le i \le k$, and so let m_i be a multiplicative inverse of n_i' modulo n_i for $1 \le i \le k$, and define

$$z_i := n_i' m_i \quad (i = 1, \ldots, k).$$

By construction, one sees that for $1 \le i \le k$, we have

$$z_i \equiv 1 \pmod{n_i}$$

and

$$z_i \equiv 0 \pmod{n_j} \quad \text{for } 1 \le j \le k \text{ with } j \ne i.$$

That is to say, for $1 \le i, j \le k$, $z_i \equiv \delta_{ij} \pmod{n_j}$, where $\delta_{ij} := 1$ for $i = j$ and $\delta_{ij} := 0$ for $i \ne j$.

Now define

$$x := \sum_{i=1}^k z_i a_i.$$

One then sees that for $1 \le j \le k$,

$$x \equiv \sum_{i=1}^k z_i a_i \equiv \sum_{i=1}^k \delta_{ij} a_i \equiv a_j \pmod{n_j}.$$

Therefore, this x solves the given system of congruences.

Moreover, if $x' \equiv x \pmod{n}$, then since $n_i \mid n$ for $1 \le i \le k$, we see that $x' \equiv x \equiv a_i \pmod{n_i}$ for $1 \le i \le k$, and so x' also solves the system of congruences.

Finally, if x' solves the system of congruences, then $x' \equiv x \pmod{n_i}$ for $1 \le i \le k$. That is, $n_i \mid (x' - x)$ for $1 \le i \le k$. Since $\gcd(n_i, n_j) = 1$ for $i \ne j$, this implies that $n \mid (x' - x)$, i.e., $x' \equiv x \pmod{n}$. □

2.3 Residue Classes

It is easy to see that for a fixed value of n, the relation $\cdot \equiv \cdot \pmod{n}$ is an *equivalence relation* on the set $\mathbb{Z}$; that is, for all $a, b, c \in \mathbb{Z}$, we have

- $a \equiv a \pmod{n}$,
- $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$, and
- $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ implies $a \equiv c \pmod{n}$.

As such, this relation partitions the set $\mathbb{Z}$ into equivalence classes. We denote the equivalence class containing the integer a by $[a \bmod n]$, or when n is clear from context, we may simply write $[a]$. Historically, these equivalence classes are called **residue classes modulo n**, and we shall adopt this terminology here as well.

It is easy to see from the definitions that

$$[a \bmod n] = a + n\mathbb{Z} := \{a + nz : z \in \mathbb{Z}\}.$$

Note that a given residue class modulo n has many different "names"; e.g., the residue class $[1]$ is the same as the residue class $[1+n]$. For any integer a in a residue class, we call a a **representative** of that class.

Theorem 2.7 *For a positive integer n, there are precisely n distinct residue classes modulo n, namely, $[a]$ for $0 \le a < n$. Moreover, for any $k \in \mathbb{Z}$, the residue classes $[k + a]$ for $0 \le a < n$ are distinct and therefore include all residue classes modulo n.*

Proof. Exercise. □

Fix a positive integer n. Let us define $\mathbb{Z}_n$ as the set of residue classes modulo n. We can "equip" $\mathbb{Z}_n$ with binary operators defining addition and multiplication in a natural way as follows: for $a, b \in \mathbb{Z}$, we define

$$[a] + [b] := [a + b],$$

and we define

$$[a] \cdot [b] := [a \cdot b].$$

Of course, one has to check this definition is unambiguous, i.e., that the addition and multiplication operators are well defined, in the sense that the sum or product of two residue classes does not depend on which particular representatives of the classes are chosen in the above definitions. More precisely, one must check that if $[a] = [a']$ and $[b] = [b']$, then $[a \text{ op } b] = [a' \text{ op } b']$, for op $\in \{+, \cdot\}$. However, this property follows immediately from Theorem 2.1.

These definitions of addition and multiplication operators on $\mathbb{Z}_n$ yield a very natural algebraic structure whose salient properties are as follows:

Theorem 2.8 *Let n be a positive integer, and consider the set $\mathbb{Z}_n$ of residue classes modulo n with addition and multiplication of residue classes as defined above.*

For all $a, b, c \in \mathbb{Z}$, we have

1. $[a] + [b] = [b] + [a]$ *(addition is commutative),*
2. $([a] + [b]) + [c] = [a] + ([b] + [c])$ *(addition is associative),*
3. $[a] + [0] = [a]$ *(existence of additive identity),*
4. $[a] + [-a] = [0]$ *(existence of additive inverses),*
5. $[a] \cdot [b] = [b] \cdot [a]$ *(multiplication is commutative),*
6. $([a] \cdot [b]) \cdot [c] = [a] \cdot ([b] \cdot [c])$ *(multiplication is associative),*
7. $[a] \cdot ([b] + [c]) = [a] \cdot [b] + [a] \cdot [c]$ *(multiplication distributes over addition)*

8. $[a] \cdot [1] = [a]$ *(existence of multiplicative identity).*

Proof. Exercise. $\square$

An algebraic structure satisfying the conditions in the above theorem is known more generally as a "commutative ring with unity," a notion that we will discuss in §5.

Note that while all elements of $\mathbb{Z}_n$ have an additive inverses, not all elements of $\mathbb{Z}_n$ have a multiplicative inverse; indeed, by Theorem 2.2, $[a \bmod n]$ has a multiplicative inverse if and only if $\gcd(a, n) = 1$. One denotes by $\mathbb{Z}_n^*$ the set of all residue classes $[a]$ of $\mathbb{Z}_n$ that have a multiplicative inverse; it is easy to see that $\mathbb{Z}_n^*$ is closed under multiplication.

Chapter 3

Computing with Large Integers

3.1 Complexity Theory

When presenting an algorithm, we shall always use a high-level, and somewhat informal, notation. However, all of our high-level descriptions can be routinely translated into the machine-language of an actual computer. So that our theorems on the running-times of algorithms have a precise mathematical meaning, we formally define an "idealized" computer: the **Random Access Machine** or **RAM**.

A RAM consists of an unbounded sequence of **memory cells**

$$m[0], m[1], m[2], \ldots$$

each of which can store an arbitrary integer, together with a **program**. A program consists of a finite sequence of instructions $I_0, I_1, \ldots$, where each instruction is of one of the following types:

arithmetic This type of instruction is of the form $x \leftarrow y \circ z$, where $\circ$ represents one of the operations addition, subtraction, multiplication, or integer division. The values y and z are of the form c, $m[a]$, or $m[m[a]]$, and x is of the form $m[a]$ or $m[m[a]]$, where c is an integer constant and a is a nonnegative integer constant. Execution of this type of instruction causes the value $y \circ z$ to be evaluated and then stored in x.

branching This type of instruction is of the form IF $y \sim z$ GOTO i, where i is the index of an instruction, and where $\sim$ is one of the comparison operators $=, \neq, <, >, \leq, \geq$, and y and z are as above. Execution of this type of instruction causes the "flow of control" to pass conditionally to instruction I_i.

halt The HALT instruction halts the execution of the program.

A RAM executes by executing instruction I_0, and continues to execute instructions, following branching instructions as appropriate, until a HALT instruction is executed.

We do not specify input or output instructions, and instead assume that the input and output are to be found in memory at some prescribed location, in some prescribed format.

To determine the running-time of a program on a given input, we charge 1 unit of time to each instruction executed.

This model of computation closely resembles a typical modern-day computer, except that we have abstracted away many annoying details. However, there are two details of real machines that cannot be ignored; namely, any real machine has a finite number of memory cells, and each cell can store numbers only in some fixed range.

The first limitation must be dealt with by either purchasing sufficient memory or designing more space-efficient algorithms.

The second limitation is especially annoying, as we will want to perform computations with quite large integers—much larger than will fit into any single memory cell of an actual machine. To deal with this limitation, we shall represent such large integers as vectors of digits to some base, so that each digit is bounded so as to fit into a memory cell. This is discussed in more detail in the next section. Using this strategy, the only other numbers we actually need to store in memory cells are "small" numbers representing array indices, addresses, and the like, which hopefully will fit into the memory cells of actual machines.

Thus, whenever we speak of an algorithm, we shall mean an algorithm that can be implemented on a RAM, such that all numbers stored in memory cells are "small" numbers, as discussed above. Admittedly, this is a bit imprecise. For the reader who demands more precision, we can make a restriction, such as the following: after the execution of m steps, all numbers stored in memory cells are bounded by $m^c + d$ in absolute value, for constants c and d — in making this formal requirement, we assume that the value m includes the number of memory cells of the input.

Even with these caveats and restrictions, the running time as we have defined it for a RAM is still only a rough predictor of performance on an actual machine. On a real machine, different instructions may take significantly different amounts of time to execute; for example, a division instruction may take much longer than an addition instruction. Also, on a real machine, the behavior of the cache may significantly affect the time it takes to load or store the operands of an instruction. However, despite all of these problems, it still turns out that measuring the running time on a RAM as we propose here is nevertheless a good "first order" predictor of performance on real machines in many cases.

If we have an algorithm for solving a certain class of problems, we expect that "larger" instances of the problem will require more time to solve that "smaller" instances. Theoretical computer scientists sometimes equate the notion of an "efficient" algorithm with that of a "polynomial-time" algorithm (although not everyone takes theoretical computer scientists very seriously, especially on this point). A polynomial-time algorithm is one whose running time on inputs of length n is bounded by $n^c + d$ for some constants c and d (a "real" theoretical computer scientist will write this as $n^{O(1)}$). To make this notion mathematically precise, one needs to define the *length* of an algorithm's input.

To define the length of an input, one chooses a "reasonable" scheme to encode all possible inputs as a string of symbols from some finite alphabet, and then defines the length of an input as the number of symbols in its encoding.

We will be dealing with algorithms whose inputs consist of arbitrary integers, or lists of such integers. We describe a possible encoding scheme using the alphabet consisting of the six symbols '0', '1', '-', ',', '(', and ')'. An integer is encoded in binary, with possibly a negative sign. Thus, the length of an integer x is approximately equal to $\log_2 |x|$. We can encode a list of integers $x_1, \ldots, x_n$ of numbers as "$(\bar{x}_1, \ldots, \bar{x}_n)$", where $\bar{x}_i$ is the encoding of x_i. We can also encode lists of lists, etc., in the obvious way. All of the mathematical objects we shall wish to compute with can be encoded in this way. For example, to encode an $n \times n$ matrix of rational numbers, we may encode each rational number as a pair of integers (the numerator and denominator), each row of the matrix as a list of n encodings of rational numbers, and the matrix as a list of n encodings of rows.

It is clear that other coding schemes are possible, giving rise to different definitions of input length. For example, we could encode inputs in some base other than 2 (but not unary!) or use a different alphabet. However, such an alternative encoding scheme would change the definition of input length by at most a constant multiplicative factor, and so would not affect the notion of a polynomial-time algorithm.

We stress that algorithms may use data structures for representing mathematical objects that look quite different from whatever encoding scheme one might choose.

3.2 Basic Integer Arithmetic

We will need algorithms to manipulate integers of arbitrary length. Since such integers will exceed the word-size of actual machines, we represent large integers as vectors of digits to some base B, along with a bit indicating the sign. Thus, for $x \in \mathbb{Z}$, we write

$$x = \pm(\sum_{i=0}^{k-1} x_i B^i) = \pm(x_{k-1} \cdots x_1 x_0)_B,$$

where $0 \leq x_i < B$ for $0 \leq i < k$, and usually, we shall have $x_{k-1} \neq 0$. The integer x will be represented in memory as a data structure consisting of a vector of digits and a sign-bit. For our purposes, we shall consider B to be a constant, and moreover, a power of 2. The choice of B as a power of 2 allows us to extract an arbitrary bit in the binary representation of a number in time $O(1)$.

We discuss basic arithmetic algorithms for positive integers; they can be very easily adapted to deal with signed integers. All of these algorithms can be implemented directly in a programming language that provides a "built-in" signed integer type that can represent all integers whose absolute value is less than B^2, and that provides the basic arithmetic operations (addition, subtraction, multiplication, integer division). So, for example, using the `C` programming language's `int` type on a typical 32-bit computer, we could take $B = 2^{15}$. The resulting algorithms would be reasonably efficient, but not nearly as efficient as algorithms that are much more carefully implemented, and which take advantage of low-level "assembly language" codes specific to a particular machine's architecture (e.g., the GNU Multi-Precision library GMP, available as `http://www.swox.com/gmp`).

Suppose we have two positive integers a and b, represented with k and ℓ base-B digits, respectively, with $k \geq \ell$. So we have $a = (a_{k-1} \cdots a_0)_B$ and $b = (b_{\ell-1} \cdots b_0)_B$. Then using the standard "paper-and-pencil" method (adapted from base-10 to base-B, of course), we can compute the base-B representation of $a + b$ or $a - b$ in time $O(k)$.

Using the standard paper-and-pencil technique, we can compute the $k + 1$ digit product $a \cdot b_0$ in time $O(k)$. We can then compute the $k + \ell$ digit product $c = a \cdot b$ as follows:

$c \leftarrow 0$
for $i \leftarrow 0$ to $\ell - 1$ do
    $c \leftarrow c + B^i \cdot ab_i$

As each loop-iteration of this algorithm takes time $O(k)$, the total running-time is $O(k\ell)$.

We now consider division with remainder. We want to compute q and r such that $a = bq + r$ and $0 \leq r < b$. Let us assume that $a \geq b$; otherwise, we can just set $q = 0$ and $r = a$. Also, let us assume that $b_{\ell-1} \neq 0$. The quotient q will have at most $m = k - \ell + 1$ base-B digits. Write $q = (q_{m-1} \cdots q_0)_B$. We compute the digits of q and the value r with the following division with remainder algorithm.

$r \leftarrow a$
for $i \leftarrow m-1$ down to 0 do
    $q_i \leftarrow \lfloor r/B^i b \rfloor$
    $r \leftarrow r - B^i \cdot q_i b$

To verify that this procedure is correct, one easily verifies by induction that in each loop iteration, $r < B^{i+1}b$.

It is perhaps not immediately clear how to efficiently implement the step $q_i \leftarrow \lfloor r/B^i b \rfloor$. As in the paper-and-pencil one has to make a reasonable "guess" at q_i, and then correct the guess if it turns out to be wrong.

More generally, consider the following situation. Let x and y be positive integers with $x/y < B$, and let $d = \lfloor x/y \rfloor$. Suppose $B = 2^t$ and that y is has $n+t$ bits in its binary representation, where $n \geq 0$. Then we can write $y = \hat{y}2^n + y'$, where $2^{t-1} \leq \hat{y} < 2^t$ and $0 \leq y' < 2^n$. We can also write $x = \hat{x}2^n + x'$, where $0 \leq \hat{x} < 2^{2t}$ and $0 \leq x' < 2^n$. Then we can approximate d by $\hat{d} = \lfloor \hat{x}/\hat{y} \rfloor$.

Theorem 3.1 *With notation as in the previous paragraph, we have $d \leq \hat{d} \leq d+2$.*

Proof. To prove the first inequality, it suffices to show that $x - \hat{d}y < y$. Using the fact that $\hat{x} = \hat{d}\hat{y} + \hat{z}$, where $0 \leq \hat{z} < \hat{y}$, we have

$$\begin{aligned} x - \hat{d}y &\leq x - \hat{d}\hat{y}2^n \leq x - (\hat{x} - (\hat{y}-1))2^n = x - \hat{x}2^n + (\hat{y}-1)2^n \\ &< 2^n + (\hat{y}-1)2^n = \hat{y}2^n \leq y. \end{aligned}$$

That proves the first inequality.

To prove the second inequality, it suffices to show that $x - \hat{d}y \geq -2y$. We have

$$x - \hat{d}y \geq x - \hat{d}(\hat{y}2^n + 2^n) = x - \hat{d}\hat{y}2^n - \hat{d}2^n \geq x - \hat{x}2^n - \hat{d}2^n \geq -\hat{d}2^n.$$

So it suffices to show that $\hat{d}2^n/y \leq 2$. Using the fact that $\hat{y} \geq 2^{t-1}$, we have

$$\frac{\hat{d}2^n}{y} \leq \frac{\hat{x}2^n}{\hat{y}y} \leq \frac{x}{\hat{y}y} \leq \frac{2^t}{\hat{y}} \leq \frac{2^t}{2^{t-1}} = 2.$$

That proves the second inequality. □

Now, going back to our division with remainder algorithm, consider executing one iteration of the main loop. If $i = 0$ and $b < B$, then we can compute $\lfloor r/b \rfloor$ in a single step, using the "built-in" division instruction; otherwise, we apply the above theorem with $y := B^i b \geq B$ and $x := r < Bx$. We can extract the high-order t bits from y and the corresponding high-order bits of x, and with one division of a number with less than $2t$-bits by a t-bit number, we get an approximation $\hat{q}_i$ to q_i. All of this can be carried out in time $O(1)$. With the above theorem, $q_i \leq \hat{q}_i \leq q_i + 2$. We perform the subtraction step $r \leftarrow r - B^i \cdot \hat{q}_i b$, which takes time $O(\ell)$. At this point, we can easily detect if our approximation was too large. Correcting the values of $\hat{q}_i$ and r then can be done in time $O(\ell)$. Thus, each loop iteration takes time $O(\ell)$, and hence the total running-time of this division with remainder algorithm is $O(m\ell)$.

We now summarize the above observations. For an integer n, we define $\mathcal{L}(n)$ to be the number of bits in the binary representation of $|n|$; more precisely,

$$\mathcal{L}(n) = \begin{cases} \lfloor \log_2 |n| \rfloor + 1 & \text{if } n \neq 0, \\ 1 & \text{if } n = 0. \end{cases}$$

Notice that for $n > 0$, $\log_2 n < \mathcal{L}(n) \le \log_2 n + 1$.

Theorem 3.2 *Let a and b be arbitrary integers, represented using the data structures described above.*

(i) We can determine an arbitrary bit in the binary representation of $|a|$ in time $O(1)$.

(ii) We can compute $a \pm b$ in time $O(\mathcal{L}(a) + \mathcal{L}(b))$.

(iii) We can compute $a \cdot b$ in time $O(\mathcal{L}(a)\mathcal{L}(b))$.

(iv) If $b > 0$, we can compute q and r such that $a = bq + r$ and $0 \le r < b$ in time $O(\mathcal{L}(b)\mathcal{L}(q))$.

From now on, we shall not worry about the implementation details of long-integer arithmetic, and will just refer directly this theorem.

Note the bound $O(\mathcal{L}(b)\mathcal{L}(q))$ in part (iv) of this theorem, which may be significantly less than the bound $O(\mathcal{L}(a)\mathcal{L}(b))$.

This theorem does not refer to the base B in the underlying implementation. The choice of B affects the values of the implied big-'O' constants; while in theory, this is of no significance, it does have a significant impact in practice.

We should point out that the algorithms discussed here for integer multiplication and division with remainder are by no means the best possible. If a and b are two integers whose length in bits is bounded by k, then the fastest known algorithm for computing ab runs in time $O(k \log k \log\log k)$. The fastest known algorithm to divide a by b also runs in time $O(k \log k \log\log k)$. We shall not discuss such fast algorithms any further here, even though in practice, they do indeed play a significant role, at least for numbers of more than a few hundred bits in length.

3.3 Greatest Common Divisors

We consider the following problem: given two positive integers a and b, compute $\gcd(a, b)$. We can do this using the well-known algorithm of Euclid, which is described in the following theorem.

Theorem 3.3 *Let $a \ge b > 0$. Define the numbers $r_0, r_1, \ldots, r_{\ell+1}$, and $q_1, \ldots, q_\ell$, where $\ell \ge 1$, as follows:*

$$
\begin{aligned}
r_0 &= a, \\
r_1 &= b, \\
r_0 &= r_1 q_1 + r_2 \quad (0 < r_2 < r_1), \\
&\vdots \\
r_{i-1} &= r_i q_i + r_{i+1} \quad (0 < r_{i+1} < r_i), \\
&\vdots \\
r_{\ell-2} &= r_{\ell-1} q_{\ell-1} + r_\ell \quad (0 < r_\ell < r_{\ell-1}), \\
r_{\ell-1} &= r_\ell q_\ell \quad (r_{\ell+1} = 0).
\end{aligned}
$$

Then $r_\ell = \gcd(a, b)$ and $\ell \le \log b / \log \phi + 1$, where $\phi = (1 + \sqrt{5})/2 \approx 1.62$.

Proof. For the first statement, one sees that for $1 \le i \le \ell$, the common divisors of r_{i-1} and r_i are the same as the common divisors of r_i and r_{i+1}, and hence $\gcd(r_{i-1}, r_i) = \gcd(r_i, r_{i+1})$. From this, it follows that $\gcd(a, b) = \gcd(r_0, r_1) = \gcd(r_\ell, 0) = r_\ell$.

To prove the second statement, we claim that for $0 \le i \le \ell - 1$, $r_{\ell-i} \ge \phi^i$. The statement will then follow by setting $i = \ell - 1$ and taking logarithms.

If $\ell = 1$, the claim is obviously true, so assume $\ell > 1$. We have $r_\ell \ge 1 = \phi^0$ and $r_{\ell-1} \ge r_\ell + 1 \ge 2 \ge \phi^1$. For $2 \le i \le \ell - 1$, using induction and applying the fact the $\phi^2 = \phi + 1$, we have

$$r_{\ell-i} \ge r_{\ell-(i-1)} + r_{\ell-(i-2)} \ge \phi^{i-1} + \phi^{i-2} = \phi^{i-2}(1 + \phi) = \phi^i,$$

which proves the claim. $\Box$

Example 3.1 Suppose $a = 100$ and $b = 35$. Then the numbers appearing in Theorem 3.3 are easily computed as follows:

i	0	1	2	3	4
r_i	100	35	30	5	0
q_i		2	1	6	

So we have $\gcd(a, b) = r_3 = 5$. $\Box$

We can easily turn the scheme described in Theorem 3.3 into a simple algorithm as follows:

```
while b ≠ 0 do
    Compute q, r such that a = bq + r, with 0 ≤ r < b
    (a, b) ← (b, r)
output a
```

By Theorem 3.3, this algorithm, known as **Euclid's algorithm**, outputs the greatest common divisor of a and b.

Theorem 3.4 *Euclid's algorithm runs in time $O(\mathcal{L}(a)\mathcal{L}(b))$.*

Proof. The running time is $O(\tau)$, where $\tau = \sum_{i=1}^{\ell} \mathcal{L}(r_i)\mathcal{L}(q_i)$. We have

$$\tau \le \mathcal{L}(b) \sum_i \mathcal{L}(q_i) \le \mathcal{L}(b) \sum_i (\log_2 q_i + 1) = \mathcal{L}(b)(\ell + \log_2(\prod_i q_i)).$$

Note that

$$a = r_0 \ge r_1 q_1 \ge r_2 q_2 q_1 \ge \cdots \ge r_\ell q_\ell \cdots q_1 \ge q_\ell \cdots q_1.$$

We also have $\ell \le \log b / \log \phi + 1$. Combining this with the above, we have

$$\tau \le \mathcal{L}(b)(\log b / \log \phi + 1 + \log_2 a) = O(\mathcal{L}(a)\mathcal{L}(b)),$$

which proves the theorem. $\Box$

Let $d = \gcd(a, b)$. We know that there exist integers s and t such that $as + bt = d$. The **extended Euclidean algorithm**, which we now describe, allows us to compute s and t.

Theorem 3.5 *Let a, b, $r_0, r_1, \ldots, r_{\ell+1}$, and $q_1, \ldots, q_\ell$ be as in Theorem 3.3. Define integers $s_0, s_1, \ldots, s_{\ell+1}$ and $t_0, t_1, \ldots, t_{\ell+1}$ as follows:*

$$s_0 := 1, \quad t_0 := 0,$$

$$s_1 := 0, \quad t_1 := 1,$$

and for $1 \leq i \leq \ell$,

$$s_{i+1} := s_{i-1} - s_i q_i, \quad t_{i+1} := t_{i-1} - t_i q_i.$$

Then

(i) for $0 \leq i \leq \ell+1$, we have $s_i a + t_i b = r_i$; in particular, $s_\ell a + t_\ell b = \gcd(a,b)$;

(ii) for $0 \leq i \leq \ell$, we have $s_i t_{i+1} - t_i s_{i+1} = (-1)^i$;

(iii) for $0 \leq i \leq \ell+1$, we have $\gcd(s_i, t_i) = 1$;

(iv) we have $|s_{\ell+1}| \leq b$ and $|t_{\ell+1}| \leq a$;

(v) for $0 \leq i \leq \ell$, we have $|t_i| \leq |t_{i+1}|$, and for $1 \leq i \leq \ell$, we have $|s_i| \leq |s_{i+1}|$;

(vi) for $0 \leq i \leq \ell+1$, we have $|s_i| \leq b$ and $|t_i| \leq a$.

Proof. (i) and (ii) are easily proved by induction on i (exercise).

(iii) follows directly from (ii).

To prove (iv), note that $s_{\ell+1}a + t_{\ell+1}b = r_{\ell+1} = 0$. We have $t_{\ell+1} \neq 0$, since otherwise, both $s_{\ell+1}$ and $t_{\ell+1}$ would be zero, contradicting (ii). This implies that $t_{\ell+1}/s_{\ell+1} = -a/b$, and then (iv) follows from the fact (iii) that $\gcd(s_{\ell+1}, t_{\ell+1}) = 1$.

For (v), one proves the statement for the t_i by induction, but with the stronger hypothesis that $t_i t_{i+1} \leq 0$ (i.e., the sign alternates) and $|t_i| \leq |t_{i+1}|$ for $0 \leq i \leq \ell$ (exercise). One argues similarly for the statement for the s_i.

(vi) follows immediately from (iv) and (v). □

Example 3.2 We continue with Example 3.1. The numbers s_i and t_i are easily computed from the q_i:

i	0	1	2	3	4
r_i	100	35	30	5	0
q_i		2	1	6	
s_i	1	0	1	-1	7
t_i	0	1	-2	3	-20

□

We can easily turn the scheme described in Theorem 3.5 into a simple algorithm, as follows:

$s \leftarrow 1,\ t \leftarrow 0$
$s' \leftarrow 0,\ t' \leftarrow 1$
while $b \neq 0$ do
  Compute q, r such that $a = bq + r$, with $0 \leq r < b$
  $(s, t, s', t') \leftarrow (s', t', s - s'q, t - t'q)$
  $(a, b) \leftarrow (b, r)$
output a, s, t

This algorithm, known as the *extended Euclidean algorithm*, computes the greatest common divisor d of a and b, together with s and t such that $as + bt = d$.

Theorem 3.6 *The extended Euclidean algorithm runs in time $O(\mathcal{L}(a)\mathcal{L}(b))$.*

Proof. It suffices to analyze the cost of computing the sequences $\{s_i\}$ and $\{t_i\}$. Consider first the cost of computing all of the t_i, which is $O(\tau)$, where $\tau = \sum_{i=1}^{\ell} \mathcal{L}(t_i)\mathcal{L}(q_i)$. By Theorem 3.5 part (vi), and arguing as in the proof of Theorem 3.4, we have

$$\begin{aligned}\tau &= \mathcal{L}(q_1) + \sum_{i=2}^{\ell} \mathcal{L}(t_i)\mathcal{L}(q_i) \leq \mathcal{L}(q_1) + \mathcal{L}(a)(\ell - 1 + \log_2(\prod_{i=2}^{\ell} q_i)) \\ &= O(\mathcal{L}(a)\mathcal{L}(b)),\end{aligned}$$

using the fact that $\prod_{i=2}^{\ell} q_i \leq b$. An analogous argument shows that one can compute all of the s_i also in time $O(\mathcal{L}(a)\mathcal{L}(b))$, and in fact, in time $O(\mathcal{L}(b)^2)$. □

We should point out that the Euclidean algorithm is not the fastest known algorithm for computing greatest common divisors. The asymptotically fastest known algorithm for computing the greatest common divisor of two numbers of bit length at most k runs in time $O(k(\log k)^2 \log\log k)$. One can also compute the corresponding values s and t within this time bound as well. Fast algorithms for greatest common divisors are not of much practical value, unless the integers involved are *very* large — at least several tens of thousands of bits in length.

3.4 Computing in $\mathbb{Z}_n$

Let $n > 1$. For computational purposes, we may represent elements of $\mathbb{Z}_n$ as elements of the set $\{0, \ldots, n-1\}$.

Addition and subtraction in $\mathbb{Z}_n$ can be performed in time $O(\mathcal{L}(n))$. Multiplication can be performed in time $O(\mathcal{L}(n)^2)$ with an ordinary integer multiplication, followed by a division with remainder.

Given $a \in \{0, \ldots, n-1\}$, we can determine if $[a \bmod n]$ has a multiplicative inverse in $\mathbb{Z}_n$, and if so, determine this inverse, in time $O(\mathcal{L}(n)^2)$ by applying the extended Euclidean algorithm. More precisely, we run the extended Euclidean algorithm to determine integers d, s, and t, such that $d = \gcd(n, a)$ and $ns + at = d$. If $d \neq 1$, then $[a \bmod n]$ is not invertible; otherwise, $[a \bmod n]$ is invertible, and $[t \bmod n]$ is its inverse. In the latter case, by part (vi) of Theorem 3.5, we know that $|t| \leq n$; we cannot have $t = \pm n$, and so either $t \in \{0, \ldots, n-1\}$, or $t + n \in \{0, \ldots, n-1\}$.

Another interesting problem is exponentiation modulo n: given $a \in \{0, \ldots, n-1\}$ and a non-negative integer e, compute $y = a^e \operatorname{rem} n$. Perhaps the most obvious way to do this is to iteratively multiply by a modulo n, e times, requiring time $O(e\,\mathcal{L}(n)^2)$. A much faster algorithm, the **repeated-squaring algorithm**, computes $y = a^e \operatorname{rem} n$ using just $O(\mathcal{L}(e))$ multiplications modulo n, thus taking time $O(\mathcal{L}(e)\,\mathcal{L}(n)^2)$.

This method works as follows. Let $e = (b_{\ell-1} \cdots b_0)_2$ be the binary expansion of e (where b_0 is the low-order bit). For $0 \leq i \leq \ell$, define $e_i = (b_{\ell-1} \cdots b_i)_2$. Also define, for $0 \leq i \leq \ell$, $y_i = a^{e_i} \operatorname{rem} n$, so $y_\ell = 1$ and $y_0 = y$. Then we have

$$e_i = 2e_{i+1} + b_i \quad (0 \leq i < \ell),$$

and hence

$$y_i = y_{i+1}^2 \cdot a^{b_i} \text{ rem } n \quad (0 \le i < \ell).$$

This idea yields the following algorithm:

$y \leftarrow 1$
for $i \leftarrow \ell - 1$ down to 0 do
  $y \leftarrow y^2$ rem n
  if $b_i = 1$ then $y \leftarrow y \cdot a$ rem n
output y

It is clear that when this algorithm terminates, $y = a^e$ rem n, and that the running-time estimate is as claimed above.

We close this chapter by observing that the Chinese Remainder Theorem (Theorem 2.6) can be made computationally effective as well. Indeed, by just using the formulas in the proof of that theorem, we see that given integers $n_1, \ldots, n_k$, and $a_1, \ldots, a_k$, with $n_i > 1$, $\gcd(n_i, n_j) = 1$ for $i \neq j$, and $0 \le a_i < n_i$, we can compute x such that $0 \le x < n$ and $x \equiv a_i \pmod{n_i}$ in time $O(\mathcal{L}(n)^2)$, where $n = \prod_i n_i$. We leave the details of this as an easy exercise.

Chapter 4

Abelian Groups

This chapter reviews the notion of an abelian group.

4.1 Definitions, Basic Properties, and Some Examples

Definition 4.1 *An **abelian group** is a set G together with a binary operation $\star$ on G such that*

1. *for all $a, b \in G$, $a \star b = b \star a$ (commutivity property),*
2. *for all $a, b, c \in G$, $a \star (b \star c) = (a \star b) \star c$ (associativity property),*
3. *there exists $e \in G$ (called the **identity element**) such that for all $a \in G$, $a \star e = a$ (identity property),*
4. *for all $a \in G$ there exists $a' \in G$ such that $a \star a' = e$ (inverse property).*

Before looking at examples, let us state some very basic properties of abelian groups that follow directly from the definition.

Theorem 4.2 *Let G be an abelian group with operator $\star$. Then we have*

1. *the identity element is unique, i.e., there is only one element $e \in G$ such that $a \star e = a$ for all $a \in G$;*
2. *inverses are unique, i.e., for all $a \in G$, there is only one element $a' \in G$ such that $a \star a'$ is the identity.*

Proof. Suppose e, e' are identities. Then since e is an identity, by the identity property in the definition, we have $e' \star e = e'$. Similarly, since e' is an identity, we have $e \star e' = e$. By the commutivity property, we have $e \star e' = e' \star e$. Thus,

$$e' = e' \star e = e,$$

and so we see that there is only one identity.

Now let $a \in G$, and suppose that $a \star a' = e$ and $a \star a'' = e$. Now, $a \star a' = e$ implies $a'' \star (a \star a') = a'' \star e$. Using the associativity and commutivity properties, the left-hand side can be written $a' \star (a \star a'')$, and by the identity property, the right-hand side can be written a''. Thus, we have

$a' \star (a \star a'') = a''$. This, together with the equation $a \star a'' = e$ implies that $a' \star e = a''$, and again applying the identity property, we have $a' = a''$. That proves a has only one inverse. $\Box$

The above proof was very straightforward, yet quite tedious if one fills in all the details. In the sequel, we shall leave proofs of this type as exercises for the reader.

There are many examples of abelian groups.

Example 4.1 The set of integers $\mathbb{Z}$ under addition forms an abelian group, with 0 being the identity, and $-a$ being the inverse of $a \in \mathbb{Z}$. $\Box$

Example 4.2 For integer n, the set $n\mathbb{Z} = \{nz : z \in \mathbb{Z}\}$ under addition forms as abelian group, again, with 0 being the identity, and $n(-z)$ being the inverse of nz. $\Box$

Example 4.3 The set of non-negative integers under addition does not form an abelian group, since inverses do not exist for integers other than 0. $\Box$

Example 4.4 The set of integers under multiplication does not form an abelian group, since inverses do not exist for integers other than ± 1. $\Box$

Example 4.5 The set of integers $\{\pm 1\}$ under multiplication forms an abelian group, with 1 being the identity, and -1 is its own inverse. $\Box$

Example 4.6 The set of rational numbers $\mathbb{Q} = \{a/b : a, b \in \mathbb{Z},\ b \neq 0\}$ under addition forms an abelian group, with 0 being the identity, and $(-a)/b$ being the inverse of a/b. $\Box$

Example 4.7 The set of non-zero rational numbers $\mathbb{Q}^*$ under multiplication forms a group, with 1 being the identity, and b/a being the inverse of a/b. $\Box$

Example 4.8 The set $\mathbb{Z}_n$ under addition forms an abelian group, where $[0 \bmod n]$ is the identity, and where $[-a \bmod n]$ is the inverse of $[a \bmod n]$. $\Box$

Example 4.9 The set $\mathbb{Z}_n^*$ of residue classes $[a \bmod n]$ with $\gcd(a, n) = 1$ under multiplication forms an abelian group, where $[1 \bmod n]$ is the identity, and if $as + nt = 1$, then $[s \bmod n]$ is the inverse of $[a \bmod n]$. $\mathbb{Z}_n^*$ is called the **multiplicative group of units modulo** n. $\Box$

Example 4.10 Continuing the previous example, let us set $n = 15$, and enumerate the elements of $\mathbb{Z}_{15}^*$. They are

$$[1], [2], [4], [7], [8], [11], [13], [14].$$

An alternative enumeration is

$$[\pm 1], [\pm 2], [\pm 4], [\pm 7].$$

$\Box$

Example 4.11 As another special case, consider $\mathbb{Z}_5^*$. We can enumerate the elements of this groups as

$$[1], [2], [3], [4]$$

or alternatively as

$$[\pm 1], [\pm 2].$$

$\Box$

Example 4.12 For any positive integer n, the set of n-bit strings under the "exclusive or" operator forms an abelian group, where every bit string is its own inverse. $\Box$

From the above examples, one can see that a group may be infinite or finite. In any case, the **order** of a group is defined to be the cardinality $|G|$ of the underlying set G defining the group.

Example 4.13 The order of $\mathbb{Z}_n$ is n. $\Box$

Example 4.14 The order of $\mathbb{Z}_p^*$ for prime p is $p-1$. $\Box$

Note that in specifying a group, one must specify both the underlying set G as well as the binary operation; however, in practice, the binary operation is often implicit from context, and by abuse of notation, one often refers to G itself as the group.

Usually, instead of using a special symbol like $\star$ for an abelian group operator, one instead uses the usual addition ("+") or multiplication ("·") operators.

If an abelian group G is written additively, then the identity element is denoted by 0_G (or just 0 if G is clear from context), and the inverse of an element $a \in G$ is denoted by $-a$. For $a, b \in G$, $a - b$ denotes $a + (-b)$. If n is a positive integer, then $n \cdot a$ denotes $a + a + \cdots + a$, where there are n terms in the sum. Moreover, if $n = 0$, then $n \cdot a$ denotes 0, and if n is a negative integer then $n \cdot a$ denotes $-((-n) \cdot a)$.

If an abelian group G is written multiplicatively, then the identity element is denoted by 1_G (or just 1 if G is clear from context), and the inverse of an element $a \in G$ is denoted by a^{-1} or $1/a$. As usual, one may write ab in place of $a \cdot b$. For $a, b \in G$, a/b denotes $a \cdot b^{-1}$. If n is a positive integer, then a^n denotes $a \cdot a \cdot \cdots \cdot a$, where there are n terms in the product. Moreover, if $n = 0$, then a^n denotes 1, and if n is a negative integer, then a^n denotes $(a^{-n})^{-1}$.

For any particular, concrete abelian group, the most natural choice of notation is clear; however, for a "generic" group, the choice is largely a matter of taste. By convention, whenever we consider a "generic" abelian group, we shall use *additive* notation for the group operation, unless otherwise specified.

We now record a few simple but useful properties of abelian groups.

Theorem 4.3 *Let G be an abelian group. Then*

1. *for all $a, b, c \in G$, if $a + b = a + c$, then $b = c$;*
2. *for all $a, b \in G$, the equation $a + x = b$ in x has a unique solution in G;*
3. *for all $a, b \in G$, $-(a + b) = (-a) + (-b)$;*
4. *for all $a \in G$, $-(-a) = a$;*
5. *for all $a \in G$ and all $n \in \mathbb{Z}$, $(-n)a = -(na) = n(-a)$.*

Proof. Exercise. $\Box$

If $G_1, \ldots, G_k$ are abelian groups, we can form the **direct product** $G_1 \times \cdots \times G_k$, which consists of all k-tuples $(a_1, \ldots, a_k)$ for $a_1 \in G_1, \ldots, a_k \in G_k$. We can view $G_1 \times \cdots \times G_k$ in a natural way as an abelian group if we define the group operation "component wise":

$$(a_1, \ldots, a_k) + (b_1, \ldots, b_k) := (a_1 + b_1, \ldots, a_k + b_k).$$

Of course, the groups $G_1, \ldots, G_k$ may be different, and the group operation applied in the ith component corresponds to the group operation associated with G_i. We leave it to the reader to verify that $G_1 \times \cdots \times G_k$ is in fact an abelian group.

In these notes, we have chosen only to discuss the notion of an abelian group. There is a more general notion of a **group**, which may be defined simply by dropping the commutivity condition in Definition 4.1, but we shall not need this notion in these notes, and restricting to abelian groups helps to simplify the discussion significantly. Nevertheless, many of the notions and results we discuss here regarding abelian groups extend (sometimes with slight modification) to general groups.

Example 4.15 The set of 2×2 integer matrices with determinant ± 1 with respect to matrix multiplication forms a group, but not an abelian group. □

4.2 Subgroups

We next introduce the notion of a subgroup.

Definition 4.4 *Let G be an abelian group, and let H be a non-empty subset of G such that*

- *for all $a, b \in H$, $a + b \in H$, and*
- *for all $a \in H$, $-a \in H$.*

*Then H is called a **subgroup of** G.*

Theorem 4.5 *If G is an abelian group, and H is a subgroup, then the binary operation of G defines a binary operation on H, and with respect to this binary operation, H forms an abelian group whose identity is the same as that of G.*

Proof. Exercise. □

Clearly, for an abelian group G, the subsets G and $\{0_G\}$ are subgroups. An easy way to find other, more interesting, subgroups within an abelian group is by using the following theorem:

Theorem 4.6 *Let G be an abelian group, and let m be an integer. Then $mG := \{ma : a \in G\}$ is a subgroup of G.*

Proof. For $ma, mb \in mG$, we have $ma + mb = m(a + b) \in mG$, and $-(ma) = m(-a) \in mG$. □

Multiplicative notation: if the abelian group G in the above theorem is written using multiplicative notation, then we write the subgroup of that theorem $G^m := \{a^m : a \in G\}$.

Example 4.16 For every integer m, the set $m\mathbb{Z}$ is a subgroup of the group $\mathbb{Z}$. □

Example 4.17 Let n be a positive integer, and let $m \in \mathbb{Z}$. By the above theorem, $m\mathbb{Z}_n$ is a subgroup of $\mathbb{Z}_n$; however, we wish to give an explicit description of this subgroup. Consider a fixed residue class $[a]$ for $a \in \mathbb{Z}$. Now, $[a] \in m\mathbb{Z}_n$ if and only if there exists $x \in \mathbb{Z}$ such that $mx \equiv a \pmod{n}$. By Theorem 2.5, such an x exists if and only if $d \mid a$, where $d = \gcd(m, n)$. Thus, $m\mathbb{Z}_n = d\mathbb{Z}_n$, and consists precisely of the n/d distinct residue classes

$$[i \cdot d] \quad (i = 0, \ldots, n/d - 1).$$

□

Because the abelian groups $\mathbb{Z}$ and $\mathbb{Z}_n$ are of such importance, it is a good idea to completely characterize all subgroups of these abelian groups. As the following two theorems show, the subgroups in the above examples are the *only* subgroups of these groups.

Theorem 4.7 *If G is a subgroup of $\mathbb{Z}$, then there exists a unique non-negative integer m such that $G = m\mathbb{Z}$.*

Proof. Actually, we have already proven this. One only needs to observe that a subset G is a subgroup if and only if it is an ideal (as defined in §1.2), and then apply Theorem 1.4. □

Theorem 4.8 *If G is a subgroup of $\mathbb{Z}_n$, then there exists a unique positive integer m dividing n such that $G = m\mathbb{Z}_n$.*

Proof. Let G be a subgroup of $\mathbb{Z}_n$. Define $G' := \{a \in \mathbb{Z} : [a] \in G\}$. It is easy to see that $G = \{[a] : a \in G'\}$.

First, we claim that G' is a subgroup of $\mathbb{Z}$. Suppose that $a, b \in G'$. This means that $[a] \in G$ and $[b] \in G$, which implies that $[a + b] = [a] + [b] \in G$, and hence $a + b \in G'$. Similarly, if $[a] \in G$, then $[-a] = -[a] \in G$, and hence $-a \in G'$.

By the previous theorem, it follows that G' is of the form $m\mathbb{Z}$ for some non-negative integer m. Moreover, note that $n \in G'$, since $[n] = [0]$ is the identity element of $\mathbb{Z}_n$, and hence belongs to G. Therefore, $m \mid n$.

So we have $G = \{[a] : a \in m\mathbb{Z}\} = m\mathbb{Z}_n$.

From the observations in Example 4.17, the uniqueness of m is clear. □

Of course, not all abelian groups have such a simple subgroup structure.

Example 4.18 Consider the group $G = \mathbb{Z}_2 \times \mathbb{Z}_2$. For any non-zero $\alpha \in G$, $\alpha + \alpha = 0_G$. From this, it is easy to see that the set $H = \{0_G, \alpha\}$ is a subgroup of G. However, for any integer m, $mG = G$ if m is odd, and $mG = \{0_G\}$ if m is even. Thus, the subgroup H is not of the form mG for any m. □

Example 4.19 Consider the group $\mathbb{Z}_n^*$ discussed in Example 4.9. The subgroup $(\mathbb{Z}_n^*)^2$ plays an important role in some situations. Integers a such that $[a] \in (\mathbb{Z}_n^*)^2$ are called **quadratic residues modulo** n. □

Example 4.20 Consider again the group $\mathbb{Z}_n^*$, for $n = 15$, discussed in Example 4.10. As discussed there, we have $\mathbb{Z}_{15}^* = \{[\pm 1], [\pm 2], [\pm 4], [\pm 7]\}$. Therefore, the elements of $(\mathbb{Z}_{15}^*)^2$ are

$$[1]^2 = [1],\ [2]^2 = [4],\ [4]^2 = [16] = [1],\ [7]^2 = [49] = [4];$$

thus, $(\mathbb{Z}_{15}^*)^2$ has order 2, consisting as it does of the two distinct elements [1] and [4].

Going further, one sees that $(\mathbb{Z}_{15}^*)^4 = \{[1]\}$. Thus, $\alpha^4 = [1]$ for all $\alpha \in \mathbb{Z}_{15}^*$.

By direct calculation, one can determine that $(\mathbb{Z}_{15}^*)^3 = \mathbb{Z}_{15}^*$; that is, cubing simply permutes $\mathbb{Z}_{15}^*$.

For any integer m, write $m = 4q + r$, where $0 \leq r < 4$. Then for any $\alpha \in \mathbb{Z}_{15}^*$, we have $\alpha^m = \alpha^{4q+r} = \alpha^{4q}\alpha^r = \alpha^r$. Thus, $(\mathbb{Z}_{15}^*)^m$ is either $\mathbb{Z}_{15}^*$, $(\mathbb{Z}_{15}^*)^2$, or $\{[1]\}$.

However, there are certainly other subgroups of $\mathbb{Z}_{15}^*$ — for example, the subgroup $\{[\pm 1]\}$. □

Example 4.21 Consider again the group $\mathbb{Z}_5^*$ from Example 4.11. As discussed there, $\mathbb{Z}_5^* = \{[\pm 1], [\pm 2]\}$. Therefore, the elements of $(\mathbb{Z}_5^*)^2$ are

$$[1]^2 = [1],\ [2]^2 = [4] = [-1];$$

thus, $(\mathbb{Z}_5^*)^2 = \{[\pm 1]\}$ and has order 2.

There are in fact no other subgroups of $\mathbb{Z}_5^*$ besides $\mathbb{Z}_5^*$, $\{[\pm 1]]\}$, and $\{[1]\}$. Indeed, if H is a subgroup containing $[2]$, then we must have $H = \mathbb{Z}_5^*$: $[2] \in H$ implies $[2]^2 = [4] = [-1] \in H$, which implies $[-2] \in H$ as well. The same holds if H is a subgroup containing $[-2]$. □

If G is an abelian group, and H_1 and H_2 are subgroups, we define $H_1 + H_2 := \{h_1 + h_2 : h_1 \in H_1,\ h_2 \in H_2\}$. Note that $H_1 + H_2$ contains $H_1 \cup H_2$.

Multiplicative notation: if G is written multiplicatively, then we write $H_1 \cdot H_2 := \{h_1 h_2 : h_1 \in H_1,\ h_2 \in H_2\}$.

Theorem 4.9 *If H_1 and H_2 are subgroups of an abelian group G, then so is $H_1 + H_2$. Moreover, any subgroup H of G that contains $H_1 \cup H_2$ contains $H_1 + H_2$, and $H_1 \subset H_2$ if and only if $H_1 + H_2 = H_2$.*

Proof. Exercise. □

Theorem 4.10 *If H_1 and H_2 are subgroups of an abelian group G, then so is $H_1 \cap H_2$.*

Proof. Exercise. □

Theorem 4.11 *If H' is a subgroup of an abelian group G, then a set $H \subset H'$ is a subgroup of G if and only if H is a subgroup of H'.*

Proof. Exercise. □

4.3 Cosets and Quotient Groups

We now generalize the notion of a congruence relation.

Let G be an abelian group, and let H be a subgroup. For $a, b \in G$, we write $a \equiv b \pmod{H}$ if $a - b \in H$.

It is easy to verify that the relation $\cdot \equiv \cdot \pmod{H}$ is an equivalence relation; that is, for all $a, b, c \in G$, we have

- $a \equiv a \pmod{H}$,
- $a \equiv b \pmod{H}$ implies $b \equiv a \pmod{H}$,
- and $a \equiv b \pmod{H}$ and $b \equiv c \pmod{H}$ implies $a \equiv c \pmod{H}$.

Therefore, this relation partitions G into equivalence classes. It is easy to see that for any $a \in G$, the equivalence class containing a is precisely $a + H := \{a + h : h \in H\}$; indeed, $a \equiv b \pmod{H}$ $\iff$ $b - a = h$ for some $h \in H$ $\iff$ $b = a + h$ for some $h \in H$ $\iff$ $b \in a + H$. The equivalence class $a + H$ is called the **coset of H in G containing** a, and an element of such a coset is called a **representative** of the coset.

Multiplicative notation: if G is written multiplicatively, then $a \equiv b \pmod{H}$ means $a/b \in H$, and the coset of H in G containing a is $aH := \{ah : h \in H\}$.

Example 4.22 Let $G = \mathbb{Z}$ and $H = n\mathbb{Z}$ for some positive integer n. Then $a \equiv b \pmod{H}$ if and only if $a \equiv b \pmod{n}$. $\Box$

Example 4.23 Let $G = \mathbb{Z}_4$ and let H be the subgroup $2\mathbb{Z}_4 = \{[0], [2]\}$. The coset of H containing $[1]$ is $\{[1], [3]\}$. These are all the cosets of H in G. $\Box$

Theorem 4.12 *Any two cosets of a subgroup H in an abelian group G have equal cardinality; i.e., there is a bijective map from one coset to the other.*

Proof. Let $a + H$ and $b + H$ be two cosets, and consider the map $f : G \to G$ that sends $x \in G$ to $x - a + b \in G$. The reader may verify that f is injective and carries $a + H$ onto $b + H$. $\Box$

An incredibly useful consequence of the above theorem is:

Theorem 4.13 *If G is a finite abelian group, and H is a subgroup of G, then the order of H divides the order of G.*

Proof. This is an immediate consequence of the previous theorem, and the fact that the cosets of H in G partition G. $\Box$

Analogous to Theorem 2.1, we have:

Theorem 4.14 *Let G be an abelian group and H a subgroup. For $a, a', b, b' \in G$, if $a \equiv a' \pmod{H}$ and $b \equiv b' \pmod{H}$, then $a + b \equiv a' + b' \pmod{H}$.*

Proof. Now, $a \equiv a' \pmod{H}$ and $b \equiv b' \pmod{H}$ means that $a' = a + h_1$ and $b' = b + h_2$ for $h_1, h_2 \in H$. Therefore, $a' + b' = (a + h_1) + (b + h_2) = (a + b) + (h_1 + h_2)$, and since $h_1 + h_2 \in H$, this means that $a + b \equiv a' + b' \pmod{H}$. $\Box$

Let G be an abelian group and H a subgroup. Theorem 4.14 allows us to define a group operation on the collection of cosets of H in G in the following natural way: for $a, b \in G$, define

$$(a + H) + (b + H) := (a + b)H.$$

The fact that this definition is unambiguous follows immediately from Theorem 4.14. Also, one can easily verify that this operation defines an abelian group. The resulting group is called the **quotient group of G modulo H**, and is denoted G/H.

The order of the group G/H is sometimes denoted $[G : H]$ and is called the **index of H in G**. If G is of finite order, then by Theorem 4.12, $[G : H] = |G|/|H|$.

Multiplicative notation: if G is written multiplicatively, then the definition of the group operation of G/H is expressed

$$(aH) \cdot (bH) := (ab)H.$$

Theorem 4.15 *If $H \subset H'$ are subgroups of an abelian group G, and $[G : H]$ is finite, then $[G : H] = [G : H'] \cdot [H' : H]$.*

Proof. Exercise. $\Box$

Example 4.24 For the additive group of integers $\mathbb{Z}$ and the subgroup $n\mathbb{Z}$ for $n > 0$, the quotient group $\mathbb{Z}/n\mathbb{Z}$ is precisely the same as the additive group $\mathbb{Z}_n$ that we have already defined. For $n = 0$, $\mathbb{Z}/n\mathbb{Z}$ is essentially just a "renaming" of $\mathbb{Z}$. □

Example 4.25 Let us return to Example 4.20. The multiplicative group $\mathbb{Z}_{15}^*$, as we saw, is of order 8. The subgroup $(\mathbb{Z}_{15}^*)^2$ has order 2. Therefore, the quotient group has order 4. Indeed, the cosets are $\alpha_{00} = \{[1], [4]\}$, $\alpha_{01} = \{[-1], [-4]\}$, $\alpha_{10} = \{[2], [-7]\}$, and $\alpha_{11} = \{[7], [-2]\}$. In the group $\mathbb{Z}_{15}^*/(\mathbb{Z}_{15}^*)^2$, α_{00} is the identity; moreover, we have

$$\alpha_{01}^2 = \alpha_{10}^2 = \alpha_{11}^2 = \alpha_{00}$$

and

$$\alpha_{01}\alpha_{10} = \alpha_{11},\ \alpha_{10}\alpha_{11} = \alpha_{01},\ \alpha_{10}\alpha_{11} = \alpha_{01}.$$

This completely describes the behavior of the group operation of the quotient group. Note that this group is essentially just a "renaming" of the group $\mathbb{Z}_2 \times \mathbb{Z}_2$. □

Example 4.26 As we saw in Example 4.21, $(\mathbb{Z}_5^*)^2 = \{[\pm 1]\}$. Therefore, the quotient group $\mathbb{Z}_5^*/(\mathbb{Z}_5^*)^2$ has order 2. The cosets of $(\mathbb{Z}_5^*)^2$ in $\mathbb{Z}_5^*$ are $\{[\pm 1]\}$ and $\{[\pm 2\}$. □

4.4 Group Homomorphisms and Isomorphisms

Definition 4.16 *A* ***homomorphism*** *from an abelian group G to an abelian group G' is a function $f : G \to G$ such that $f(a + b) = f(a) + f(b)$ for all $a, b \in G$.*

The set $f^{-1}(1_{G'})$ is called the ***kernel*** *of f, and is denoted $\ker(f)$. The set $f(G)$ is called the* ***image*** *of f.*

If f is bijective, then f is called an ***isomorphism*** *of G with G'.*

It is easy to see that if f is an isomorphism of G with G', then the inverse function f^{-1} is an isomorphism of G' with G. If such an isomorphism exists, we say that G and G' are **isomorphic**, and write $G \cong G'$. we stress that an isomorphism of G with G' is essentially just a "renaming" of the group elements — all structural properties of the group are preserved.

Theorem 4.17 *Let f be a homomorphism from an abelian group G to an abelian group G'.*

1. *$f(0_G) = 0_{G'}$.*
2. *$f(-a) = -f(a)$ for all $a \in G$.*
3. *$f(na) = nf(a)$ for all $n \in \mathbb{Z}$ and $a \in G$.*
4. *For any subgroup H of G, $f(H)$ is a subgroup of G'.*
5. *$\ker(f)$ is a subgroup of G.*
6. *For all $a, b \in G$, $f(a) = f(b)$ if and only if $a \equiv b \pmod{\ker(f)}$.*
7. *f is injective if and only if $\ker(f) = \{0_G\}$.*
8. *For any subgroup H' of G', $f^{-1}(H')$ is a subgroup of G containing $\ker(f)$.*
9. *For any subgroup H of G, $f^{-1}(f(H)) = H + \ker(f)$.*

Proof. Exercise. □

Part (7) of the above theorem is particular useful: to check that a homomorphism is injective, it suffices to determine if $\ker(f) = \{0_G\}$.

The following theorems, while very simple to prove, are also very useful.

Theorem 4.18 *If H is a subgroup of an abelian group G, then the map $f : G \to G/H$ given by $f(a) = a + H$ is a surjective homomorphism whose kernel is H. This is sometimes called the "natural" map from G to G/H.*

Proof. Exercise. □

Theorem 4.19 *Let G and G' be abelian groups. Let f be a homomorphism from G into G'. Then the map $\bar{f} : G/\ker(f) \to f(G)$ that sends the coset $a + \ker(f)$ for $a \in G$ to $f(a)$ is unambiguously defined and is an isomorphism of $G/\ker(f)$ with $f(G)$.*

Proof. Exercise. □

Theorem 4.20 *Let G and G' be abelian groups. Let f be a homomorphism from G into G'. The subgroups of G containing $\ker(f)$ are in one-to-one correspondence with the subgroups of $f(G)$, where the the subgroup H in G containing $\ker(f)$ corresponds to the subgroup $f(H)$ in $f(G)$.*

Proof. Exercise. □

Theorem 4.21 *Let G be an abelian group with subgroups H_1, H_2 such that $H_1 \cap H_2 = \{0_G\}$. Then the map that sends $(h_1, h_2) \in H_1 \times H_2$ to $h_1 + h_2 \in H_1 + H_2$ is an isomorphism of $H_1 \times H_2$ with $H_1 + H_2$.*

Proof. Exercise. □

Example 4.27 For any abelian group G and any integer m, the map that sends $a \in G$ to $ma \in G$ is clearly a homomorphism from G into G. The image of this homomorphism is mG. We call this map the **m-multiplication map on G**. If G is written multiplicatively, we call this the **m-power map on G**. □

Example 4.28 Consider the m-multiplication map on $\mathbb{Z}$. The image of this map is $m\mathbb{Z}$, and the kernel is $\{0\}$ if $m \neq 0$, and is $\mathbb{Z}$ if $m = 0$. □

Example 4.29 Consider the m-multiplication map on $\mathbb{Z}_n$. The image of this map is $m\mathbb{Z}_n$, which as we saw above in Example 4.17 is a subgroup of $\mathbb{Z}_n$ of order n/d, where $d = \gcd(n, m)$. Thus, this map is bijective if and only if $d = 1$, in which case it is an isomorphism of $\mathbb{Z}_n$ with itself. □

Example 4.30 For positive integer n, consider the natural map $f : \mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} = \mathbb{Z}_n$. Theorem 4.20 says that this map gives a one-to-one correspondence between the subgroups of $\mathbb{Z}$ containing $n\mathbb{Z}$ and the subgroups of $\mathbb{Z}_n$. Moreover, it follows from Theorem 4.7 that the subgroups of $\mathbb{Z}$ containing $n\mathbb{Z}$ are precisely $m\mathbb{Z}$ for $m \mid n$. From this, it follows that the subgroups of $\mathbb{Z}_n$ are precisely $m\mathbb{Z}_n$ for $m \mid n$. We already proved this in Theorem 4.8. □

Example 4.31 As was demonstrated in Example 4.25, the quotient group $\mathbb{Z}_{15}^*/(\mathbb{Z}_{15}^*)^2$ is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$. □

Example 4.32 Let G_1, G_2 be abelian groups. The map that sends $(a_1, a_2) \in G_1 \times G_2$ to $a_1 \in G_1$ is a homomorphism from $G_1 \times G_2$ to G_1. Its image is G_1, and its kernel is $\{0_{G_1}\} \times G_2$. □

Example 4.33 If $G = G_1 \times G_2$ for abelian groups G_1 and G_2, and H_1 is a subgroup of G_1 and H_2 is a subgroup of G_2, then $H := H_1 \times H_2$ is a subgroup of G, and $G/H \cong G_1/H_1 \times G_2/H_2$. □

4.5 Cyclic Groups

Let G be an abelian group. For $a \in G$, define $\langle a \rangle := \{za : z \in \mathbb{Z}\}$. It is clear that $\langle a \rangle$ is a subgroup of G, and moreover, that any subgroup H of G that contains a must also contain $\langle a \rangle$. The subgroup $\langle a \rangle$ is called **the subgroup generated by** a. Also, one defines the **order** of a to be the order of the subgroup $\langle a \rangle$, which is denoted $\mathrm{ord}(a)$.

More generally, for $a_1, \ldots, a_k \in G$, we define $\langle a_1, \ldots, a_k \rangle := \{z_1 a_1 + \cdots + z_k a_k : z_1, \ldots, z_k \in \mathbb{Z}\}$. One also verifies that $\langle a_1, \ldots, a_k \rangle$ is a subgroup of G, and that any subgroup H of G that contains $a_1, \ldots, a_k$ must contain $\langle a_1, \ldots, a_k \rangle$. The subgroup $\langle a_1, \ldots, a_k \rangle$ is called the **subgroup generated by** $a_1, \ldots, a_k$.

An abelian group G is called a **cyclic group** if $G = \langle a \rangle$ for some $a \in G$, in which case, a is called a **generator for** G.

Multiplicative notation: if G is written multiplicatively, then $\langle a \rangle := \{a^z : z \in \mathbb{Z}\}$, and $\langle a_1, \ldots, a_k \rangle := \{a_1^{z_1} \cdots a_k^{z_k} : z_1, \ldots, z_k \in \mathbb{Z}\}$.

Example 4.34 $\mathbb{Z}$ is a cyclic group generated by 1. The only other generator is -1. □

Example 4.35 $\mathbb{Z}_n$ is a cyclic group generated by $[1 \bmod n]$. More generally, $\langle [m \bmod n] \rangle = m\mathbb{Z}_n$, and so is cyclic of order n/d, where $d = \gcd(m, n)$. □

We can very quickly characterize all cyclic groups, up to isomorphism. Suppose that G is a cyclic group with generator a. Consider the map $f : \mathbb{Z} \to G$ that sends $z \in \mathbb{Z}$ to $za \in G$. This map is clearly a surjective homomorphism. Now, $\ker(f)$ is a subgroup of $\mathbb{Z}$, and by Theorem 4.7, it must be of the form $n\mathbb{Z}$ for some non-negative integer n. Also, by Theorem 4.19, we have $\mathbb{Z}/n\mathbb{Z} \cong G$.

Case 1: $n = 0$. In this case, $\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}$, and so we see $G \cong \mathbb{Z}$. Moreover, by Theorem 4.17, the only integer z such that $za = 0_G$ is the integer 0, and more generally, $z_1 a = z_2 a$ if and only if $z_1 = z_2$.

Case 2: $n > 0$. In this case, $\mathbb{Z}/n\mathbb{Z} = \mathbb{Z}_n$, and so we see that $G \cong \mathbb{Z}_n$. Moreover, by Theorem 4.17, $za = 0_G$ if and only if $n \mid z$, and more generally, $z_1 a = z_2 a$ if and only if $z_1 \equiv z_2 \pmod{n}$. The order of G is evidently n, and G consists of the distinct elements

$$0 \cdot a, 1 \cdot a, \ldots, (n-1) \cdot a.$$

From this characterization, we immediately have:

Theorem 4.22 *Let G be an abelian group and let $a \in G$. If there exists a positive integer m such that $ma = 0_G$, then the least such integer is the order of a. Moreover, if G of finite order n, then $\mathrm{ord}(a) \mid n$, and in particular $na = 0_G$.*

Proof. The first statement follows from the above characterization. For the second statement, since $\langle a \rangle$ is a subgroup of G, by Theorem 4.13, its order must divide that of G. Of course, if $ma = 0_G$, then for any multiple m' of m (in particular, $m' = n$), we also have $m'a = 0_G$. □

Based on the this theorem, we can trivially derive a classical result:

Theorem 4.23 (Fermat's Little Theorem) *For any prime p, and any integer $x \not\equiv 0 \pmod{p}$, we have $x^{p-1} \equiv 1 \pmod{p}$. Moreover, for any integer x, we have $x^p \equiv x \pmod{p}$.*

Proof. The first statement follows from Theorem 4.22, and the fact that Since $\mathbb{Z}_p^*$ is an abelian group of order $p-1$. The second statement is clearly true if $x \equiv 0 \pmod{p}$, and if $x \not\equiv 0 \pmod{p}$, we simply multiply both sides of the congruence $x^{p-1} \equiv 1 \pmod{p}$ by x. □

It also follows from the above characterization of cyclic groups that that any subgroup of a cyclic group is cyclic — indeed, we have already characterized the subgroups of $\mathbb{Z}$ and $\mathbb{Z}_n$ in Theorems 4.7 and 4.8, and it is clear that these subgroups are cyclic. Indeed, it is worth stating the following:

Theorem 4.24 *Let G be a cyclic group of finite order n. Then the subgroups of G are in one-to-one correspondence with the positive divisors of n, where each such divisor d corresponds to a cyclic subgroup of G_d of order d. Moreover:*

- *G_d is the image of the (n/d)-multiplication map (or (n/d)-power map).*
- *G_d contains precisely those elements in G whose order divides d; i.e., G_d is the kernel of the d-multiplication map (or d-power map, for multiplicative groups).*
- *$G_d \supset G_{d'}$ if and only if $d \mid d'$.*

Proof. Since $G \cong \mathbb{Z}_n$, this follows immediately from Theorem 4.8, and the discussion in Example 4.17. We leave the details to the reader. □

Example 4.36 Since $m\mathbb{Z}_n$ is cyclic of order n/d, where $d = \gcd(m, n)$, we have $m\mathbb{Z}_n \cong \mathbb{Z}_{n/d}$. □

Example 4.37 Consider the group $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$. For $m \in \mathbb{Z}$, then the element $m([1 \bmod n_1], [1 \bmod n_2]) = ([0 \bmod n_1], [0 \bmod n_2])$ if and only if $n_1 \mid m$ and $n_2 \mid m$. This implies that $([1 \bmod n_1], [1 \bmod n_2])$ has order $\operatorname{lcm}(n_1, n_2)$. In particular, if $\gcd(n_1, n_2) = 1$, then $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ is cyclic of order $n_1 n_2$, and so $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \cong \mathbb{Z}_{n_1 n_2}$. Moreover, if $\gcd(n_1, n_2) = d > 1$, then all elements of $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ have order dividing $n_1 n_2 / d$, and so $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ cannot be cyclic. □

Example 4.38 As we saw in Example 4.20, all elements of $\mathbb{Z}_{15}^*$ have order dividing 4, and since $\mathbb{Z}_{15}^*$ has order 8, we conclude that $\mathbb{Z}_{15}^*$ is not cyclic. □

Example 4.39 The group $\mathbb{Z}_5^*$ is cyclic, with $[2]$ being a generator:

$$[2]^2 = [4] = [-1], \quad [2]^3 = [-2], \quad [2]^4 = [1].$$

□

Theorem 4.25 *If G is a cyclic group, and $f : G \to G'$ is a homomorphism from G to the abelian group G', then $f(G)$ is cyclic.*

Proof. Exercise. □

Theorem 4.26 *If G is a finite abelian group of order n, and m is an integer relatively prime to n, then $mG = G$.*

Proof. Consider the m-multiplication map on G.

We claim that the kernel of this map is $\{0_G\}$. Indeed, $ma = 0_G$, implies $\mathrm{ord}(a)$ divides m, and since $\mathrm{ord}(a)$ also divides n and $\gcd(m, n) = 1$, we must have $\mathrm{ord}(a) = 1$, i.e., $a = 0_G$. That proves the claim.

Thus, the m-multiplication map is injective, and because G is finite, it must be surjective as well. □

Theorem 4.27 *If G is an abelian group of prime order, then G is cyclic.*

Proof. Let $|G| = p$. Let $a \in G$ with $a \neq 0_G$. Since $\mathrm{ord}(a) \mid p$, we have $\mathrm{ord}(a) = 1$ or $\mathrm{ord}(a) = p$. Since $a \neq 0_G$, we must have $\mathrm{ord}(a) \neq 1$, and so $\mathrm{ord}(a) = p$, which implies a generates G. □

Theorem 4.28 *Suppose that a is an element of an abelian group, and for some prime p and $e \geq 1$, we have $p^e a = 0_G$ and $p^{e-1}a \neq 0_G$. Then a has order p^e.*

Proof. If m is the order of a, then since $p^e a = 0_G$, we have $m \mid p^e$. So $m = p^f$ for some $0 \leq f \leq e$. If $f < e$, then $p^{e-1}a = 0_G$, contradicting the assumption that $p^{e-1}a \neq 0_G$. □

Theorem 4.29 *Suppose G is an abelian group with $a_1, a_2 \in G$ such that the a_1 is of finite order n_1 and a_2 is of finite order n_2, and $d = \gcd(n_1, n_2)$. Then $\mathrm{ord}(a_1 + a_2) \mid n_1 n_2/d$. Moreover, $\mathrm{ord}(a_1 + a_2) = n_1 n_2$ if and only if $d = 1$.*

Proof. Since $(n_1n_2/d)(a_1 + a_2) = (n_2/d)(n_1a_1) + (n_1/d)(n_2a_1) = 0_G + 0_G = 0_G$, the order of $a_1 + a_2$ must divide n_1n_2/d. On the one hand, if $d > 1$, then clearly $a_1 + a_2$ cannot have order n_1n_2. On the other hand, if $d = 1$, then $m(a_1 + a_2) = 0_G$ implies $ma_1 = -ma_2$; since $-ma_2$ has order dividing n_2, so does ma_1; also, ma_1 has order dividing n_1, and so we conclude that the order of ma_1 is 1, since n_1 and n_2 are relatively prime. That is, $ma_1 = 0_G$, from which it follows that $n_1 \mid m$. By a symmetric argument, one finds $n_2 \mid m$, and again, as n_1 and n_2 are relatively prime, this implies that $n_1n_2 \mid m$. That proves that $\mathrm{ord}(a_1 + a_2) = n_1n_2$. □

For an abelian group G, the **exponent** of G is defined to be the least positive integer m such that $mG = \{0_G\}$ if such an integer exists, and is defined to be 0 otherwise.

We first state some basic properties.

Theorem 4.30 *Let G be an abelian group of exponent m.*

1. *For any integer m' such that $m'G = \{0_G\}$, we have $m \mid m'$.*

2. *If G has finite order, then m divides $|G|$.*

3. *If $m \neq 0$, for any $a \in G$, the order of a is finite, and $\mathrm{ord}(a) \mid m$.*

Proof. Exercise. □

Theorem 4.31 *For finite abelian groups G_1, G_2 whose exponents are m_1 and m_2, the exponent of $G_1 \times G_2$ is $\mathrm{lcm}(m_1, m_2)$.*

Proof. Exercise. □

Theorem 4.32 *If a finite abelian group G has exponent m, then G contains an element of order m. In particular, a finite abelian group is cyclic if and only if its order equals its exponent.*

Proof. The second statement follows immeditely from the first. For the first statement, assume that $m > 1$, and let $m = \prod_{i=1}^{r} p_i^{e_i}$ be the prime factorization of m.

First, we claim that for each $1 \leq i \leq r$, there exists $a_i \in G$ such that $(m/p_i)a_i \neq 0_G$. Suppose the claim were false: then for some i, $(m/p_i)a = 0_G$ for all $a \in G$; however, this contradicts the minimality property in the definition of the exponent m. That proves the claim.

Let $a_1, \ldots, a_r$ be as in the above claim. Then by Theorem 4.28, $(m/p_i^{e_i})a_i$ has order $p_i^{e_i}$ for each $1 \leq i \leq r$. Finally, by Theorem 4.29, the group element

$$(m/p_1^{e_1})a_1 + \cdots + (m/p_r^{e_r})a_r$$

has order m. □

Theorem 4.33 *If G is a finite abelian group of order n, and p is a prime dividing n, then G contains an element of order p.*

Proof. First, note that if G contains an element whose order is divisible by p, then it contains an element of order p; indeed, if a has order mp, then ma has order p.

Let $a_1, \ldots, a_n$ be an enumeration of all the elements of G, and consider the tower of subgroups

$$H_0 := \{0_G\}, \quad H_i := \langle a_1, \ldots, a_i \rangle \quad (i = 1, \ldots, n).$$

We have

$$n = |H_n|/|H_0| = \prod_{i=1}^{n} |H_i|/|H_{i-1}| = \prod_{i=1}^{n} |H_i/H_{i-1}|,$$

and therefore, for some $1 \leq i \leq n$, $p \mid |H_i/H_{i-1}|$. Let $k = |H_i/H_{i-1}|$. Now, the quotient group H_i/H_{i-1} is clearly cyclic and is generated by the coset $a_i + H_{i-1}$. Let $k' = \mathrm{ord}(a_i)$. Then $k'(a_i + H_{i-1}) = k'a_i + H_{i-1} = 0_G + H_{i-1}$. Therefore, $k \mid k'$. That proves that $p \mid \mathrm{ord}(a_i)$, so we are done. □

With this last theorem, we can prove the converse of Theorem 4.26.

Theorem 4.34 *If G is a finite abelian group of order n, and $mG = G$, then m is relatively prime to n.*

Proof. To the contrary, suppose that p is a prime dividing m and n. Then G contains an element of order p by Theorem 4.33, and this element is in the kernel of the m-multiplication map. Therefore, this map is not injective, and hence not surjective since G is finite. Thus, $mG \neq G$, a contradiction. □

4.6 The Structure of Finite Abelian Groups

We next state a theorem that characterizes all finite abelian groups up to isomorphism.

Theorem 4.35 (Fundamental Theorem of Finite Abelian Groups) *A finite abelian group (with more than one element) is isomorphic to a direct product of cyclic groups*

$$\mathbb{Z}_{p_1^{e_1}} \times \cdots \times \mathbb{Z}_{p_r^{e_r}},$$

where the p_i are primes (not necessarily distinct) and the e_i are positive integers. This direct product of cyclic groups is unique up to the order of the factors.

An alternative characterization of this theorem is the following:

Theorem 4.36 *A finite abelian group (with more than one element) is isomorphic to a direct product of cyclic groups*

$$\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_t},$$

where all $m_i > 1$ and $m_1 \mid m_2 \mid \cdots \mid m_t$. Moreover, the integers $m_1, \ldots, m_t$ are unique, and m_t is the exponent of the group.

Proof. This follows easily from Theorem 4.35 and Example 4.37. Details are left to the reader. □

Note that Theorems 4.32 and 4.33 follow as easy corollaries of Theorem 4.35; however, the direct proofs of those theorems are much simpler than the proof of Theorem 4.35.

The proof of Theorem 4.35 is a bit tedious, and we break it into three lemmas.

Lemma 4.37 *Let G be a finite abelian group. Then G is isomorphic to a direct product of abelian groups, each of whose exponent is a prime power.*

Proof. Let m be the exponent of G. If m is a prime power, we are done. Otherwise, write $m = m_1 m_2$, where $\gcd(m_1, m_2) = 1$, and $1 < m_1, m_2 < m$. Consider the subgroups $m_1 G$ and $m_2 G$. Clearly, $m_1 G$ has exponent m_2 and $m_2 G$ has exponent m_1. Since $\gcd(m_1, m_2) = 1$, $m_1 G \cap m_2 G = \{0_G\}$. By Theorem 4.21, $m_1 G_1 \times m_2 G_2 \cong m_1 G + m_2 G$. Again, since $\gcd(m_1, m_2) = 1$, there exist integers x_1, x_2 such that $m_1 x_1 + m_2 x_2 = 1$. For any $a \in G$,

$$a = (m_1 x_1 + m_2 x_2) a = m_1(x_1 a) + m_2(x_2 a) \in m_1 G_1 \times m_2 G,$$

and hence $G = m_1 G_1 \times m_2 G_2$. Thus, we have an isomorphism of G with $m_1 G_1 \times m_2 G_2$. The lemma follows by induction on m. □

Lemma 4.38 *Let G be a finite abelian group of exponent p^e for prime p and positive integer e. Then there exist positive integers $e_1, \ldots, e_k$ such that $G \cong \mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_k}}$.*

Proof. The proof is a bit long.

Consider a sequence of group elements $(a_1, \ldots, a_k)$, with $k \geq 0$, along with a corresponding "tower" of subgroups

$$H_0 := \{0_G\}, \quad H_i := \langle a_1, \ldots, a_i \rangle \quad (i = 1, \ldots, k).$$

Let us call the sequence of $(a_1, \dots, a_k)$ "good" if for $1 \le i \le k$, there exists a positive integer e_i such that

$$p^{e_i} a_i = 0_G, \;\; p^{e_i - 1} a_i \notin H_{i-1}, \;\; \text{and} \;\; p^{e_i} a \in H_{i-1} \text{ for all } a \in G. \tag{4.1}$$

Let us study some of the properties of a good sequence $(a_1, \dots, a_k)$. To this end, for $0 \le i \le k$, and for $a \in G$, let us define $\mathrm{ord}_i(a)$ to be the least positive integer m such that $ma \in H_i$. Clearly, $\mathrm{ord}_0(a) = \mathrm{ord}(a)$, and $\mathrm{ord}_i(a)$ is the order of the coset $a + H_i$ in the quotient group G/H_i. Since $\mathrm{ord}(a)a = 0_G \in H_i$, it follows that $\mathrm{ord}_i(a) \mid \mathrm{ord}(a) \mid p^e$.

From the definitions, it is clear that condition (4.1) above is equivalent to the condition

$$p^{e_i} = \mathrm{ord}(a_i) = \mathrm{ord}_{i-1}(a_i) = \max\{\mathrm{ord}_{i-1}(a) : a \in G\}. \tag{4.2}$$

Assume now that $k > 0$, and consider expressions of the form $x_1 a_1 + \cdots + x_k a_k$, for $x_1, \dots, x_k \in \mathbb{Z}$. It is clear from the definition that every element of H_k can be expressed in this way.

Claim 1: $x_1 a_1 + \cdots + x_k a_k = 0_G$ implies $p^{e_i} \mid x_i$ for $1 \le i \le k$.

To prove this claim, assume that $x_1 a_1 + \cdots + x_k a_k = 0_G$, and $p^{e_j} \nmid x_j$ for some j. Moreover, assume that the index j is maximal, i.e., $p^{e_{j'}} \mid x_{j'}$ for $j < j' \le k$. Write $x_j = p^f y$, where $p \nmid y$ and $0 \le f < e_j$, and let y' be a multiplicative inverse of y modulo p^{e_j}. Then we have $p^f a_j = -(y' x_1 a_1 + \cdots + y' x_{j-1} a_{j-1}) \in H_{j-1}$, contradicting the assumption that $p^{e_j - 1} a_j \notin H_{j-1}$. That proves the claim.

From this claim, it is easy to see that the map sending $([x_1 \bmod p^{e_1}], \dots, [x_k \bmod p^{e_k}])$ to $x_1 a_1 + \cdots + x_k a_k$ is an isomorphism of $\mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_k}}$ with H_k. That the definition of this map is unambiguous follows from the fact that $\mathrm{ord}(a_i) = p^{e_i}$ for $1 \le i \le k$. Moreover, it is clear that the the map is a surjective homomorphism, and by Claim 1, the kernel is trivial.

It is also clear that under this isomorphism, the subgroup H_i of H_k, for $1 \le i \le k$, corresponds to the subgroup of $\mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_k}}$ consisting of all k-tuples whose last $k - i$ components are zero.

Next, assume that $H_k \subsetneq G$. We show how to extend a good sequence $(a_1, \dots, a_k)$ to a good sequence $(a_1, \dots, a_k, a_{k+1})$ for some $a_{k+1} \in G$.

If $k = 0$, this is trivial: simply choose a_1 to be an element of maximal order in G.

Now assume $k > 0$. Let us choose $b \in G$ such that $\mathrm{ord}_k(b)$ is maximal. Let $\mathrm{ord}_k(b) = p^f$. Note that $f \le e_i$ for $1 \le i \le k$, since by definition, $p^{e_i} b \in H_{i-1} \subset H_k$. In general, we have $p^f \mid \mathrm{ord}(b)$, and if $p^f = \mathrm{ord}(b)$, then we can set $a_{k+1} := b$, and we are done. However, in general, we cannot expect that $p^f = \mathrm{ord}(b)$. Note, however, that $\mathrm{ord}_k(b + h) = \mathrm{ord}_k(b)$ for all $h \in H_k$, so if we can find $h \in H_k$ such that $p^f(b + h) = 0_G$, we will also be done.

Write $p^f b = \sum_{i=1}^k x_i a_i$ for some integers x_i. If we can find integers $z_1, \dots, z_k$ such that $p^f z_i + x_i \equiv 0 \pmod{p^{e_i}}$ for $1 \le i \le k$, then setting $h := \sum_{i=1}^k z_i a_i$, we see that

$$p^f(b + h) = \sum_{i=1}^k (p^f z_i + x_i) a_i = 0_G,$$

and we will be done. Moreover, by Theorem 2.5, such integers $z_1, \dots, z_k$ exist provided $p^f \mid x_i$ for $1 \le i \le k$.

Claim 2: If $p^f b = \sum_{i=1}^k x_i a_i$ as above, then $p^f \mid x_i$ for all $1 \le i \le k$.

To prove this claim, assume that $p^f \nmid x_j$ for some j. Multiplying the equation $p^f b = \sum_{i=1}^k x_i a_i$ by $p^{e_j - f}$, we see that $p^{e_j} b$ can be expressed as $\sum_{i=1}^k x'_i a_i$, where $p^{e_j} \nmid x'_j$. By Claim 1, it follows that $p^{e_j} b \notin H_{j-1}$, which contradicts the assumption that $p^{e_j} a \in H_{j-1}$ for all $a \in G$. That proves the claim.

So we see that we can always extend a good sequence. Since G is finite, by starting with the empty sequence, and extending it one element at a time, we will eventually find a good sequence $(a_1, \ldots, a_k)$ such that $H_k = G$, and as we have seen above, H_k is isomorphic to $\mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_k}}$.

That proves the lemma. $\Box$

These two lemmas prove the existence part of Theorem 4.35. The following lemma proves the uniqueness part.

Lemma 4.39 *Suppose that* $G \cong \times_i \mathbb{Z}_{{p_i}^{e_i}}$ *and* $G \cong \times_j \mathbb{Z}_{{q_j}^{f_j}}$, *for primes* p_i *and* q_j. *Then the prime powers* $p_i^{e_i}$ *and* $q_j^{e_j}$ *are the same, after re-ordering.*

Proof. Clearly, $|G| = \prod_i p_i^{e_i} = \prod_j q_j^{f_j}$, and so the distinct primes appearing among the p_i are the same as the distinct primes appearing among the q_j.

Now, fix a prime p dividing $|G|$, and for $a \geq 0$, let $D_a(p)$ be the number of elements of G whose order divides p^a. Also, for $k \geq 1$, let us define $M_k(p)$ to be the number of indices i such that $p = p_i$ and $e_i \geq k$, and similarly, $N_k(p)$ to be the number of indices j such that $p = q_j$ and $e_j \geq k$. From the isomorphism $G \cong \times_i \mathbb{Z}_{{p_i}^{e_i}}$, it is easy to verify (exercise) that

$$D_a(p) = p^{\sum_{k=1}^{a} M_k(p)},$$

and similarly, from the isomorphism $G \cong \times_j \mathbb{Z}_{{q_j}^{f_j}}$, that

$$D_a(p) = p^{\sum_{k=1}^{a} N_k(p)}.$$

It follows that for all $a \geq 0$,

$$\sum_{k=1}^{a} M_k(p) = \sum_{k=1}^{a} N_k(p),$$

from which it follows from a simple induction argument that $M_k(p) = N_k(p)$ for all $k \geq 1$. From this, it is easy to verify (exercise) that the prime powers $p_i^{e_i}$ and $q_j^{e_j}$ are the same, after re-ordering. $\Box$

Chapter 5

Rings

This chapter reviews the notion of a ring, more specifically, a commutative ring with unity.

5.1 Definitions, Basic Properties, and Examples

Definition 5.1 *A **commutative ring with unity** is a set R together with addition and multiplication operators on R, such that*

1. *the set R under addition forms an abelian group, and we denote the additive identity by 0_R;*
2. *multiplication is commutative, i.e., for all $a, b \in R$, we have $ab = ba$;*
3. *multiplication is associative, i.e., for all $a, b, c \in R$, we have $a(bc) = (ab)c$;*
4. *multiplication distributes over addition, i.e., for all $a, b, c \in R$, $a(b + c) = ab = ac$;*
5. *there exists a non-zero multiplicative identity, i.e., there exists an element $1_R \in R$, with $1_R \neq 0_R$, such that $1_R \cdot a = a$ for all $a \in R$.*

There are other, more general (and less convenient) types of rings, but we will not be discussing them here. Therefore, to simplify terminology, from now on we will refer to a commutative ring with unity simply as a **ring**.

When there is no possibility for confusion, one may write "0" instead of "0_R" and "1" instead of "1_R."

We first state some simple facts which follow directly from the definition.

Theorem 5.2 *Let R be a ring. Then*

1. *the multiplicative identity is unique;*
2. *$0_R \cdot a = 0_R$ for all $a \in R$;*
3. *$(-a)b = a(-b) = -(ab)$ for all $a, b \in R$;*
4. *$(-a)(-b) = ab$ for all $a, b \in R$;*
5. *$(na)b = a(nb) = n(ab)$ for all $n \in \mathbb{Z}$ and $a, b \in R$;*
6. *$(\sum_{i=1}^{n} a_i)(\sum_{j=1}^{m} b_j) = \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j$ for all $a_i, b_j \in R$.*

Proof. Exercise. □

Example 5.1 The set $\mathbb{Z}$ under the usual rules of multiplication and addition forms a ring. □

Example 5.2 For $n > 1$, the set $\mathbb{Z}_n$ under the rules of multiplication and addition defined in §2.3 forms a ring. Note that $\mathbb{Z}_n$ with $n = 1$ does not satisfy our definition, since our definition requires that in a ring R, $1_R \neq 0_R$, and in particular, R must contain at least two elements. Actually, if we have an algebraic structure that satisfies all the requirements of a ring except that $1_R = 0_R$, then it is easy to see that R consists of the single element 0_R, where $0_R + 0_R = 0_R$ and $0_R \cdot 0_R = 0_R$. □

Example 5.3 The set $\mathbb{Q}$ of rational numbers under the usual rules of multiplication and addition forms a ring. □

Let R be a ring.

The **characteristic** of R is defined as the exponent of the underlying additive group. Alternatively, the characteristic if the least positive integer m such that $m \cdot 1_R = 0_R$, if such an m exists, and is zero otherwise.

For $a, b \in R$, we say that b **divides** a, written $b \mid a$, if there exists $c \in R$ such that $a = bc$, in which case we say that b is a **divisor** of a.

Note that parts 1-5 of Theorem 1.1 holds for an arbitrary ring.

5.1.1 Units and Fields

Let R be a ring. We call $u \in R$ a **unit** if it has a multiplicative inverse, i.e., if $uu' = 1_R$ for some $u' \in R$. It is easy to see that the multiplicative inverse of u, if its exists, is unique, and we denote it by u^{-1}; also, for $a \in R$, we may write a/u to denote au^{-1}. It is clear that a unit u divides every $a \in R$.

We denote the set of units R^*. It is easy to verify that the set R^* is closed under multiplication, from which it follows that R^* is an abelian group, called the **multiplicative group of units** of R.

If R^* contains all non-zero elements of R, then R is called a **field**.

Example 5.4 The only units in the ring $\mathbb{Z}$ are ± 1. Hence, $\mathbb{Z}$ is not a field. □

Example 5.5 For $n > 1$, the units in $\mathbb{Z}_n$ are the residue classes $[a \bmod n]$ with $\gcd(a, n) = 1$. In particular, if n is prime, all non-zero residue classes are units, and conversely, if n is composite, some non-zero residue classes are not units. Hence, $\mathbb{Z}_n$ is a field if and only if n is prime. □

Example 5.6 Every non-zero element of $\mathbb{Q}$ is a unit. Hence, $\mathbb{Q}$ is a field. □

5.1.2 Zero divisors and Integral Domains

Let R be a ring. An element $a \in R$ is called a **zero divisor** if $a \neq 0$ and there exists non-zero $b \in R$ such that $ab = 0_R$.

If R has no zero divisors, then it is called an **integral domain**. Put another way, R is an integral domain if and only if $ab = 0_R$ implies $a = 0_R$ or $b = 0_R$ for all $a, b \in R$.

Note that if u is a unit in R, it cannot be a zero divisor (if $ub = 0_R$, then multiplying both sides of this equation by u^{-1} yields $b = 0_R$). In particular, it follows that any field is an integral domain.

Example 5.7 $\mathbb{Z}$ is an integral domain. $\Box$

Example 5.8 For $n > 1$, $\mathbb{Z}_n$ is an integral domain if and only if n is prime. In particular, if n is composite, so $n = n_1 n_2$ with $1 < n_1, n_2 < n$, then $[n_1]$ and $[n_2]$ are zero divisors: $[n_1][n_2] = [0]$, but $[n_1] \neq [0]$ and $[n_2] \neq [0]$. $\Box$

Example 5.9 $\mathbb{Q}$ is an integral domain. $\Box$

We have the following "cancellation law":

Theorem 5.3 *If R is a ring, and $a, b, c \in R$ such that $a \neq 0_R$ and a is not a zero divisor, then $ab = ac$ implies $b = c$.*

Proof. $ab = bc$ implies $a(b - c) = 0_R$. The fact that $a \neq 0$ and a is not a zero divisor implies that we must have $b - c = 0_R$, i.e., $b = c$. $\Box$

Theorem 5.4 *If D is an integral domain, then*

1. *for all $a, b, c \in D$, $a \neq 0_D$ and $ab = ac$ implies $b = c$;*
2. *for all $a, b \in D$, $a \mid b$ and $b \mid a$ if and only if $a = bc$ for $c \in D^*$.*

Proof. The first statement follows immediately from the previous theorem and the definition of an integral domain.

For the second statement, if $a = bc$ for $c \in D^*$, then we also have $b = ac^{-1}$; thus, $b \mid a$ and $a \mid b$. Conversely, $a \mid b$ implies $b = ax$ for $x \in D$, and $b \mid a$ implies $a = by$ for $y \in D$, and hence $b = bxy$. Cancelling b, we have $1_D = xy$, and so x and y are units. $\Box$

It follows from the above theorem that in an integral domain D, if $a, b \in D$ with $b \neq 0_D$ and $b \mid a$, then there is a unique $c \in D$ such that $a = bc$, which we may denote as a/b.

5.1.3 Subrings

A subset R' of a ring R is called a **subring** if

- R' is an additive subgroup of R,
- R' is closed under multiplication, i.e., $ab \in R'$ for all $a, b \in R$, and
- $1_R \in R'$.

Note that the requirement that $1_R \in R'$ is not redundant. Some authors do not make this restriction.

It is clear that the operations of addition and multiplication on R make R' itself into a ring, where 0_R is the additive identity of R' and 1_R is the multiplicative identity of R'.

Example 5.10 $\mathbb{Z}$ is a subring of $\mathbb{Q}$. $\Box$

5.1.4 Direct products of rings

If $R_1, \ldots, R_k$ are rings, then the set of all k-tuples $(a_1, \ldots, a_k)$ with $a_i \in R_i$ for $1 \leq i \leq k$, with addition and multiplication defined component-wise, forms a ring. The ring is denoted $R_1 \times \cdots \times R_k$, and is called the **direct product** of $R_1, \ldots, R_k$.

Clearly, $(a_1, \ldots, a_k)$ is a unit (resp., zero divisor) in $R_1 \times \cdots \times R_k$ if and only if each component a_i is a unit (resp., zero divisor) in R_i.

5.2 Polynomial rings

If R is a ring, then we can form the **ring of polynomials** $R[T]$, consisting of all polynomials $\sum_{i=0}^{k} a_i T^i$ in the indeterminate (or variable) T, with coefficients in R, with addition and multiplication being defined in the usual way: let $a = \sum_{i=0}^{k} a_i T^i$ and $b = \sum_{i=0}^{\ell} b_i T^i$; then

$$a + b := \sum_{i=0}^{\max(k,\ell)} (a_i + b_i) T^i,$$

where one interprets a_i as 0_R if $i > k$ and b_i as 0_R if $j > \ell$, and

$$a \cdot b := \sum_{i=0}^{k+\ell} c_i T^i,$$

where $c_i := \sum_{j=0}^{i} a_j b_{i-j}$, and one interprets a_j as 0_R if $j > k$ and b_{i-j} as 0_R if $i - j > \ell$.

For $a = \sum_{i=0}^{k} a_i T^i \in R[T]$, if $k = 0$, we call a a **constant** polynomial, and if $k > 0$ and $a_k \neq 0_R$, we call a a **non-constant** polynomial.

Clearly, R is a subring of $R[T]$, and consists precisely of the constant polynomials of $R[T]$. In particular, 0_R is the additive identity of $R[T]$, and 1_R is the multiplicative identity of $R[T]$.

5.2.1 Polynomials versus polynomial functions

Of course, a polynomial $a = \sum_{i=0}^{k} a_i T^i$ defines a polynomial function on R that sends $x \in R$ to $\sum_{i=0}^{k} a_i x^i$, and we denote the value of this function as $a(x)$. However, it is important to to regard polynomials over R as formal expressions, and not to identify them with their corresponding functions. In particular, a polynomial $a = \sum_{i=0}^{k} a_i T^i$ is zero if and only if $a_i = 0_R$ for $0 \leq i \leq k$, and two polynomials are equal if and only if their difference is zero. This distinction is important, since there are rings R over which two different polynomials define the same function. One can of course define the ring of polynomial functions on R, but in general, that ring has a different structure from the ring of polynomials over R.

Example 5.11 In the ring $\mathbb{Z}_p$, for prime p, we have $x^p - x = [0]$ for all $x \in \mathbb{Z}_p$. But consider the polynomial $a = T^p - T \in \mathbb{Z}_p[T]$. We have $a(x) = 0_R$ for all $x \in 0_R$, and hence the function defined by a is the zero function, yet a is *not* the zero polynomial. $\Box$

5.2.2 Basic properties of polynomial rings

Let R be a ring.

For non-zero $a \in R[T]$, if $a = \sum_{i=0}^{k} a_i T^i$ with $a_k \neq 0_R$, we call k the **degree** of a, denoted $\deg(a)$, and we call a_k the **leading coefficient** of a, denoted $\mathrm{lc}(a)$, and we call a_0 the **constant term** of a. If $\mathrm{lc}(a) = 1_R$, then a is called **monic**.

Note that if $a, b \in R[T]$, both non-zero, and their leading coefficients are not both zero divisors, then the product ab is non-zero and $\deg(ab) = \deg(a) + \deg(b)$. However, if the leading coefficients of a and b are both zero divisors, then we could get some "collapsing": we could have $ab = 0_R$, or $ab \neq 0_R$ but $\deg(ab) < \deg(a) + \deg(b)$.

For the zero polynomial, we establish the following conventions: its leading coefficient and constant term are defined to be 0_R, and its degree is defined to be "$-\infty$", where it is understood that for all integers $x \in \mathbb{Z}$, $-\infty < x$, and $(-\infty) + x = x + (-\infty) = -\infty$, and $(-\infty) + (-\infty) = -\infty$.

This notion of "negative infinity" should not be construed as a useful algebraic notion — it is simply a convenience of notation; for example, it allows us to succinctly state that

> for all $a, b \in R[T]$, $\deg(ab) \leq \deg(a) + \deg(b)$, with equality holding if the leading coefficients of a and b are not both zero divisors.

Theorem 5.5 *Let D be an integral domain. Then*

1. *for all $a, b \in D[T]$, $\deg(ab) = \deg(a) + \deg(b)$;*
2. *$D[T]$ is an integral domain;*
3. *$(D[T])^* = D^*$.*

Proof. Exercise. □

5.2.3 Division with remainder

An extremely important property of polynomials is a division with remainder property, analogous to that for the integers:

Theorem 5.6 (Division with Remainder Property) *Let R be a ring. For $a, b \in R[T]$ with $\mathrm{lc}(b) \in R^*$, there exist unique $q, r \in R[T]$ such that $a = bq + r$ and $\deg(r) < \deg(b)$.*

Proof. Consider the set S of polynomials of the form $a - xb$ with $x \in R[T]$. Let $r = a - qb$ be an element of S of minimum degree. We must have $\deg(r) < \deg(b)$, since otherwise, we would have $r' := r - (\mathrm{lc}(r)\mathrm{lc}(b)^{-1} T^{\deg(r) - \deg(b)}) \cdot b \in S$, and $\deg(r') < \deg(r)$, contradicting the minimality of $\deg(r)$.

That proves the existence of r and q. For uniqueness, suppose that $a = bq + r$ and $a = bq' + r'$, where $\deg(r) < \deg(b)$ and $\deg(r') < \deg(b)$. This implies $r' - r = b(q - q')$. However, if $q \neq q'$, then

$$\deg(b) > \deg(r' - r) = \deg(b(q - q')) = \deg(b) + \deg(q - q') \geq \deg(b),$$

which is impossible. Therefore, we must have $q = q'$, and hence $r = r'$. □

If $a = bq + r$ as in the above theorem, we define $a \text{ rem } b := r$.

Theorem 5.7 *If K is field, then for $a, b \in K[T]$ with $b \neq 0_K$, there exist unique $q, r \in K[T]$ such that $a = bq + r$ and $\deg(r) < \deg(b)$.*

Proof. Clear. □

Theorem 5.8 *For a ring R and $a \in R[T]$ and $x \in R$, $a(x) = 0_R$ if and only if $(T - x)$ divides a.*

Proof. Let us write $a = (T - x)q + r$, with $q, r \in R[T]$ and $\deg(r) < 1$, which means that $r \in R$. Then we have $a(x) = (x - x)q(x) + r = r$. Thus, $a(x) = 0$ if and only if $T - x$ divides a. $\square$

With R, a, x as in the above theorem, we say that x is a **root** of a if $a(x) = 0_R$.

Theorem 5.9 *Let D be an integral domain, and let $a \in D[T]$, with $\deg(a) = k \geq 0$. Then a has at most k roots.*

Proof. We can prove this by induction. If $k = 0$, this means that a is a non-zero element of D, and so it clearly has no roots.

Now suppose that $k > 0$. If a has no roots, we are done, so suppose that a has a root x. Then we can write $a = q(T - x)$, where $\deg(q) = k - 1$. Now, for any root y of a with $y \neq x$, we have $0_D = a(y) = q(y)(y - x)$, and using the fact that D is an integral domain, we must have $q(y) = 0$. Thus, the only roots of a are x and the roots of q. By induction, q has at most $k - 1$ roots, and hence a has at most k roots. $\square$

Theorem 5.10 *Let D be an infinite integral domain, and let $a \in D[T]$. If $a(x) = 0_D$ for all $x \in D$, then $a = 0_D$.*

Proof. Exercise. $\square$

With this last theorem, one sees that for an infinite integral domain D, there is a one-to-one correspondence between polynomials over D and polynomial functions on D.

5.3 Ideals and Quotient Rings

Throughout this section, let R denote a ring.

Definition 5.11 *An **ideal** of R is a additive subgroup I of R that is closed under multiplication by element of R, that is, for all $x \in I$ and $a \in R$, $xa \in I$.*

Clearly, $\{0\}$ and R are ideals of R.

Example 5.12 For $m \in \mathbb{Z}$, the set $m\mathbb{Z}$ is not only an additive subgroup of $\mathbb{Z}$, it is also an ideal of the ring $\mathbb{Z}$. $\square$

Example 5.13 For $m \in \mathbb{Z}$, the set $m\mathbb{Z}_n$ is not only an additive subgroup of $\mathbb{Z}_n$, it is also an ideal of the ring $\mathbb{Z}_n$. $\square$

If $d_1, \ldots, d_k \in R$, then the set

$$d_1 R_1 + \cdots + d_k R := \{d_1 a_1 + \cdots + d_k a_k : a_1, \ldots, a_k \in R\}$$

is clearly an ideal, and contains $d_1, \ldots, d_k$. It is called the **ideal generated by** $d_1, \ldots, d_k$. Clearly, any ideal I that contains $d_1, \ldots, d_k$ must contain $d_1 R_1 + \cdots + d_k R$. If an ideal I is equal to dR for some $d \in R$, then we say that I is a **principal ideal**.

Note that if I and J are ideals, then so are $I + J := \{x + y : x \in I, y \in J\}$ and $I \cap J$.

Throughout the rest of this section, I denotes an ideal of R.

Since I is an additive subgroup, we may adopt the congruence notation in §4.3, writing $a \equiv b \pmod{I}$ if and only if $a - b \in I$.

Note that if $I = dR$, then $a \equiv b \pmod{I}$ if and only if $d \mid (a - b)$, and as a matter of notation, we may simply write this congruence as $a \equiv b \pmod{d}$.

If we just consider R as an additive group, then as we saw in §4.3, we can form the additive group R/I of cosets, where $(a + I) + (b + I) := (a + b) + I$. By considering also the multiplicative structure of R, we can also view R/I as a ring. To do this, we need the following fact.

Theorem 5.12 *If $a \equiv a' \pmod{I}$ and $b \equiv b' \pmod{I}$, then $ab \equiv a'b' \pmod{I}$.*

Proof. If $a' = a + x$ for $x \in I$ and $b' = b + y$ for $y \in I$, then $a'b' = ab + ay + bx + xy$. Since I is closed under multiplication by elements of R, we see that $ay, bx, xy \in I$, and since it is closed under addition, $ay + bx + xy \in I$. Hence, $a'b' - ab \in I$. □

So we define multiplication on R/I as follows: for $a, b \in R$,

$$(a + I) \cdot (b + I) := ab + I.$$

The previous theorem is required to show that this definition is unambiguous. It is trivial to show that if $I \subsetneq R$, then R/I satisfies the properties defining of a ring, using the corresponding properties for R. Note that the restriction that $I \subsetneq R$ is necessary; otherwise R/I would consist of a single element and could not satisfy the requirement that the additive and multiplicative identities are distinct. This ring is called the **quotient ring** or **residue class ring of R modulo I**.

As a matter of notation, for $a \in R$, we define $[a \bmod I] := a + I$, and if $I = dR$, we may write this simply as $[a \bmod d]$. If I is clear from context, we may also just write $[a]$.

Example 5.14 For $n > 1$, the ring $\mathbb{Z}_n$ as we have defined it is precisely the quotient ring $\mathbb{Z}/n\mathbb{Z}$. □

Example 5.15 Let m be a monic polynomial over R with $\deg(m) = \ell > 0$, and consider the quotient ring $S = R[T]/mR[T]$. Every element of S can be written uniquely as $[a \bmod m]$, where a is a polynomial over R of degree less than ℓ. □

5.4 Ring homomorphisms and isomorphisms

Throughout this section, R and R' denote rings.

Definition 5.13 *A function f from R to R' is called a* ***homomorphism*** *if it is a homomorphism with respect to the underlying additive groups of R and R', and if in addition,*

1. *$f(ab) = f(a)f(b)$ for all $a, b \in R$, and*
2. *$f(1_R) = 1_{R'}$.*

Moreover, if f is a bijection, then it is called an ***isomorphism*** *of R with R'.*

Note that the requirement that $f(1_R) = 1_{R'}$ is not redundant. Some authors do not make this requirement.

It is easy to see that if f is an isomorphism of R with R', then the inverse function f^{-1} is an isomorphism of R' with R. If such an isomorphism exists, we say that R is **isomorphic** to R', and write $R \cong R'$. We stress that an isomorphism of R with R' is essentially just a "renaming" of elements.

A homomorphism f from R to R' is also a homomorphism from the additive group of R to the additive group of R'. We may therefore adopt the terminology of kernel and image, as defined in §4.4, and note that all the results of Theorem 4.17 apply as well here. In particular, $f(a) = f(b)$ if and only if $a \equiv b \pmod{\ker(f)}$, and f is injective if and only if $\ker(f) = \{0_R\}$. However, we may strengthen Theorem 4.17 as follows:

Theorem 5.14 *Let $f : R \to R'$ be a homomorphism.*

1. *For any subring S of R, $f(S)$ is a subring of R'.*
2. *For any ideal I of R, $f(I)$ is an ideal of $f(R)$.*
3. *$\ker(f)$ is an ideal of R.*
4. *For any ideal I' of R', $f^{-1}(I')$ is an ideal of R (and contains $\ker(f)$).*
5. *The restriction f^* of f to R^* is a homomorphism from the multiplicative group R^* into the multiplicative group $(R')^*$, and $\ker(f^*) = (1_R + \ker(f)) \cap R^*$.*

Proof. Exercise. □

An injective homomorphism $f : R \to R'$ is called an **embedding** of R in R'. In this case, $f(R)$ is a subring of R' and $R \cong f(R)$, and we say that "R is embedded in R'," or as an abuse of terminology, one might simply say that "R is a subring of R'."

Theorems 4.18, 4.19, and 4.20 also have natural analogs:

Theorem 5.15 *If I is an ideal R, then the map $f : R \to R/I$ given by $f(a) = a + I$ is a surjective homomorphism whose kernel is I. This is sometimes called the "natural" map from R to R/I.*

Proof. Exercise. □

Theorem 5.16 *Let f be a homomorphism from R into R'. Then the map $\bar{f} : R/\ker(f) \to f(R)$ that sends the coset $a + \ker(f)$ for $a \in R$ to $f(a)$ is unambiguously defined and is an isomorphism of $R/\ker(f)$ with $f(R)$.*

Proof. Exercise. □

Theorem 5.17 *Let f be a homomorphism from R into R'. The ideals of R containing $\ker(f)$ are in one-to-one correspondence with the ideals of $f(R)$, where the the ideal I in R containing $\ker(f)$ corresponds to the ideal $f(I)$ in $f(R)$.*

Proof. Exercise. □

Example 5.16 For $n > 1$, the natural map f from $\mathbb{Z}$ to $\mathbb{Z}_n$ sends $a \in \mathbb{Z}$ to the residue class $[a \bmod n]$. This is a surjective map with kernel $n\mathbb{Z}$. Consider the multiplicative group of units $\mathbb{Z}^* = \{\pm 1\}$ and the restriction f^* of f to $\mathbb{Z}^*$. This is a homomorphism from $\mathbb{Z}^*$ into $\mathbb{Z}_n^*$ with kernel $\ker(f^*) = (1 + n\mathbb{Z}) \cap \{\pm 1\}$. Thus, if $n = 2$, $\ker(f^*) = \{\pm 1\}$, and otherwise, $\ker(f^*) = \{1\}$. □

Example 5.17 We may restate the Chinese Remainder Theorem (see Theorem 2.6) in more algebraic terms. Let $n_1, \ldots, n_k$ be integers, all greater than 1, such that $\gcd(n_i, n_j) = 1$ for all $1 \leq i < j \leq k$. Consider the homomorphism from the ring $\mathbb{Z}$ to the ring $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$ that sends $x \in \mathbb{Z}$ to $([x \bmod n_1], \ldots, [x \bmod n_k])$. In our new language, Theorem 2.6 says that this homomorphism is surjective and the kernel is $n\mathbb{Z}$, where $n = \prod_{i=1}^{k} n_i$. Therefore, the map that sends $[x \bmod n] \in \mathbb{Z}_n$ to $([x \bmod n_1], \ldots, [x \bmod n_k])$ is an isomorphism of the ring $\mathbb{Z}_n$ with the ring $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$. The restriction of this map to $\mathbb{Z}_n^*$ yields an isomorphism of $\mathbb{Z}_n^*$ with $\mathbb{Z}_{n_1}^* \times \cdots \times \mathbb{Z}_{n_k}^*$.
□

Example 5.18 Let n_1, n_2 be positive integers with $n_1 > 1$ and $n_1 \mid n_2$. Then the map $f : \mathbb{Z}_{n_2} \to \mathbb{Z}_{n_1}$ that sends $[a \bmod n_2]$ to $[a \bmod n_1]$ is a surjective homomorphism, $[a \bmod n_2] \in \ker(f)$ if and only if $n_1 \mid a$, i.e., $\ker(f) = n_1\mathbb{Z}_{n_2}$. □

Example 5.19 Fix $x \in R$. The map that sends $a \in R[T]$ to $a(x)$ is a homomorphism from $R[T]$ onto R. The kernel is the ideal generated by $(T - x)$. Thus, $R[T]/(T - x) \cong R$. □

Example 5.20 Let us continue with Example 5.15. The map $f : R \to S$ that sends $r \in R$ to $[r \bmod m] \in S$ is an embedding of R in S. □

Example 5.21 For any ring R, consider the map $f : \mathbb{Z} \to R$ that sends $m \in \mathbb{Z}$ to $m \cdot 1_R$ in R. This is clearly a homomorphism of rings. If $\ker(f) = \{0\}$, then the ring $\mathbb{Z}$ is embedded in R, and R has characteristic zero. If $\ker(f) = n\mathbb{Z}$ for $n > 0$, then the ring $\mathbb{Z}_n$ is embedded in R, and R has characteristic n. □

Chapter 6

Polynomials over Fields

Throughout this chapter, K denotes a field, and D denotes the ring $K[T]$ of polynomials over K. Like the ring $\mathbb{Z}$, D is an integral domain, and as we shall see, because of the division with remainder property for polynomials, D has many other properties in common with $\mathbb{Z}$ as well. Indeed, essentially all the ideas and results from Chapters 1 and 2 carry over almost immediately from $\mathbb{Z}$ to D.

Recall that for $a, b \in D$, we write $b \mid a$ if $a = bc$ for some $c \in D$; note that $\deg(a) = \deg(b) + \deg(c)$. Also, recall that because of the cancellation law for an integral domain, if $b \mid a$ and $b \neq 0$, then the choice of c above is unique, and may be denoted a/b.

The units D^* of D are precisely the units K^* of K; i.e., the non-zero constants. We call two polynomials $a, b \in D$ **associates** if $a = bu$ for $u \in K^*$. Clearly, any non-zero polynomial a is associate to a unique monic polynomial, called the **monic associate** of a. Note that a polynomial a is a unit if and only if it is associate to 1. Let us call a polynomial **normalized** if it is either zero or monic.

We call a polynomial p **irreducible** if it is non-constant and all divisors of p are associate to 1 or p. Conversely, we call a polynomial n **reducible** if it is non-constant and is not irreducible. Equivalently, non-constant n is reducible if and only if there exist polynomials $a, b \in D$ of degree strictly less that n such that $n = ab$.

Clearly, if a and b are associate polynomials, then a is irreducible if and only if b is irreducible.

The irreducible polynomials play a role similar to that of the prime numbers. Just as it is convenient to work with only positive prime numbers, it is also convenient to restrict attention to monic irreducible polynomials.

Corresponding to Theorem 1.2, every non-zero polynomial can be expressed as a unit times a product of monic irreducibles in an essentially unique way:

Theorem 6.1 *Every non-zero polynomial n can be expressed as*

$$n = u \cdot \prod_p p^{\nu_p(n)},$$

where u is a unit, and the product is over all monic irreducible polynomials, with all but a finite number of the exponents zero. Moreover, the exponents and the unit are uniquely determined by n.

To prove this theorem, we may assume that n is monic, since the non-monic case trivially reduces to the monic case.

The proof of the existence part of Theorem 6.1 is just as for Theorem 1.2. If n is 1 or a monic irreducible, we are done. Otherwise, there exist $a, b \in D$ of degree strictly less than n such that $n = ab$, and again, we may assume that a and b are monic. We then apply an inductive argument with a and b.

The proof of the uniqueness part of Theorem 6.1 is almost identical to that of Theorem 1.2.

Analogous to Theorem 1.4, we have:

Theorem 6.2 *For any ideal $I \subset D$, there exists a unique normalized polynomial d such that $I = dD$.*

Proof. We first prove the existence part of the theorem. If $I = \{0\}$, then $d = 0$ does the job, so let us assume that $I \neq \{0\}$. Let d be a monic polynomial of minimal degree in I. We want to show that $I = dD$.

We first show that $I \subset dD$. To this end, let c be any element in I. It suffices to show that $d \mid c$. Using the Division with Remainder Property, write $c = qd + r$, where $\deg(r) < \deg(d)$. Then by the closure properties of ideals, one sees that $r = c - qd$ is also an element of I, and by the minimality of the choice of d, we must have $r = 0$. Thus, $d \mid c$.

We next show that $dD \subset I$. This follows immediately from the fact that $d \in I$ and the closure properties of ideals.

That proves the existence part of the theorem. As for uniqueness, note that if $dD = d'D$, we have $d \mid d'$ and $d' \mid d$, from which it follows that $d' = ud$ for a unit u. $\square$

For $a, b \in D$, we call $d \in D$ a **common divisor** of a and b if $d \mid a$ and $d \mid b$; moreover, we call d the **greatest common divisor** of a and b if d is normalized, and all other common divisors of a and b divide d. It is immediate from the definition of a greatest common divisor that it is unique if it exists at all.

Analogous to Theorem 1.5, we have:

Theorem 6.3 *For any $a, b \in D$, there exists a greatest common divisor d of a and b, and moreover, $aD + bD = dD$; in particular, $as + bt = d$ for some $s, t \in D$.*

Proof. Replace the symbol $\mathbb{Z}$ in the proof of Theorem 1.5 with the symbol D. $\square$

For $a, b \in D$, we denote by $\gcd(a, b)$ the greatest common divisor of a and b.

We say that a and b are **relatively prime** if $\gcd(a, b) = 1$. Notice that a and b are relatively prime if and only if $aD + bD = D$, i.e., if and only if there exist $s, t \in D$ such that $as + bt = 1$.

Analogous to Theorem 1.6, we have:

Theorem 6.4 *For $a, b, c \in D$ such that $c \mid ab$ and $\gcd(a, c) = 1$, we have $c \mid b$.*

Proof. Replace the symbol $\mathbb{Z}$ in the proof of Theorem 1.6 with the symbol D. $\square$

Analogous to Theorem 1.7, we have:

Theorem 6.5 *Let $p \in D$ be irreducible, and let $a, b \in D$. Then $p \mid ab$ implies that $p \mid a$ or $p \mid b$.*

Proof. The only divisors of p are associate to 1 or p. Thus, $\gcd(p, a)$ is either 1 or the monic associate of p. If $p \mid a$, we are done; otherwise, if $p \nmid a$, we must have $\gcd(p, a) = 1$, and by the previous theorem, we conclude that $p \mid b$. $\square$

Now to prove the uniqueness part of Theorem 6.1. Clearly, the choice of the unit u is uniquely determined: $u = \mathrm{lc}(n)$. Suppose we have

$$p_1 \cdots p_r = p'_1 \cdots p'_s,$$

where the p_i and p'_i are monic irreducible polynomials (duplicates are allowed among the p_i and among the p'_i). If $r = 0$, we must have $s = 0$ and we are done. Otherwise, as p_1 divides the right-hand side, by inductively applying Theorem 6.5, one sees that p_1 is equal to some p'_i. We can cancel these terms and proceed inductively (on r).

That completes the proof of Theorem 6.1.

For non-zero polynomials a and b, it is easy to see that

$$\gcd(a, b) = \prod_p p^{\min(\nu_p(a), \nu_p(b))},$$

where the function $\nu_p(\cdot)$ is as implicitly defined in Theorem 6.1.

For $a, b \in D$ a **common multiple** of a and b is a polynomial m such that $a \mid m$ and $b \mid m$; moreover, m is a **least common multiple** of a and b if m is normalized, and m divides all common multiples of a and b. In light of Theorem 6.1, it is clear that the least common multiple exists and is unique; indeed, if we denote the least common multiple of a and b as $\mathrm{lcm}(a, b)$, then for non-zero polynomials a and b, we have

$$\mathrm{lcm}(a, b) = \prod_p p^{\max(\nu_p(a), \nu_p(b))}.$$

Moreover, for all $a, b \in D$, we have

$$\gcd(a, b) \cdot \mathrm{lcm}(a, b) = ab.$$

Recall that for polynomials a, b, n, we write $a \equiv b \pmod{n}$ when $n \mid (a - b)$.

For a non-zero polynomial n, and $a \in D$, we say that a is a **unit modulo** n if there exists $a' \in D$ such that $aa' \equiv 1 \pmod{n}$, in which case we say that a' is a **multiplicative inverse of** a **modulo** n.

All of the results we proved in Chapter 2 for integer congruences carry over almost identically to polynomials. As such, we do not give proofs of any of the results here. The reader may simply check that the proofs of the corresponding results translate almost directly.

Theorem 6.6 *An polynomial a is a unit modulo n if and only if a and n are relatively prime.*

Theorem 6.7 *If a is relatively prime to n, then $ax \equiv ax' \pmod{n}$ if and only if $x \equiv x' \pmod{n}$. More generally, if $d = \gcd(a, n)$, then $ax \equiv ax' \pmod{n}$ if and only if $x \equiv x' \pmod{n/d}$.*

Theorem 6.8 *Let n be a non-zero polynomial and let $a, b \in D$. If a is relatively prime to n, then the congruence $ax \equiv b \pmod{n}$ has a solution x; moreover, any integer x' is a solution if and only if $x \equiv x' \pmod{n}$.*

Theorem 6.9 *Let n be a non-zero polynomial and let $a, b \in D$. Let $d = \gcd(a, n)$. If $d \mid b$, then the congruence $ax \equiv b \pmod{n}$ has a solution x, and any integer x' is also a solution if and only if $x \equiv x' \pmod{n/d}$. If $d \nmid b$, then the congruence $ax \equiv b \pmod{n}$ has no solution x.*

Theorem 6.10 (Chinese Remainder Theorem) *Let $k > 0$, and let $a_1, \ldots, a_k \in D$, and let $n_1, \ldots, n_k$ be non-zero polynomials such that $\gcd(n_i, n_j) = 1$ for all $1 \le i < j \le k$. Then there exists a polynomial x such that*

$$x \equiv a_i \pmod{n_i} \quad (i = 1, \ldots, k).$$

Moreover, any other polynomial x' is also a solution of these congruences if and only if $x \equiv x' \pmod{n}$, where $n := \prod_{i=1}^{k} n_i$.

If we set $R = D/nD$ and $R_i = D/n_iD$ for $1 \le i \le k$, then in ring-theoretic language, the Chinese Remainder Theorem says the homomorphism from the ring D to the ring $R_1 \times \cdots \times R_1$ that sends $x \in D$ to $([x \bmod n_1], \ldots, [x \bmod n_k])$ is a surjective homomorphism with kernel nD, and hence $R \cong R_1 \times \cdots \times R_k$.

Let us recall the formula for the solution x (see proof of Theorem 2.6). We have

$$x := \sum_{i=1}^{k} z_i a_i,$$

where

$$z_i := n_i' m_i, \quad n_i' := n/n_i, \quad m_i n_i' \equiv 1 \pmod{n_i} \quad (i = 1, \ldots, k).$$

Now, let us consider the special case of the Chinese Remainder Theorem where $a_i \in K$ and and $n_i = (T - b_i)$ with $b_i \in K$, for $1 \le i \le k$. The condition that $\gcd(n_i, n_j) = 1$ for all $i \ne j$ is equivalent to the condition that $b_i \ne b_j$ for all $i \ne j$. Then a polynomial x satisfies the system of congruences if and only if $x(b_i) = a_i$ for $1 \le i \le k$. Moreover, we have $n_i' = \prod_{j \ne i}(T - b_j)$, and $m_i := 1/\prod_{j \ne i}(b_i - b_j)$ is a multiplicative inverse of n_i' modulo n_i. So we get

$$x = \sum_{i=1}^{k} a_i \frac{\prod_{j \ne i}(T - b_j)}{\prod_{j \ne i}(b_i - b_j)}.$$

The reader will recognize this as the LaGrange Interpolation Formula. Thus, the Chinese Remainder Theorem for polynomials includes LaGrange Interpolation as a special case.

As we saw in Example 5.15, if m is a monic polynomial over K of degree $\ell > 0$, then the elements of quotient ring $S = D/mD$ are in one-to-one correspondence with the polynomials of degree less than ℓ. More precisely, every element of S can be expressed uniquely as $[a \bmod m]$, where a is a polynomial of degree less than ℓ. As we saw in Example 5.20, the ring S contains an isomorphic copy of K. Now, if m happens to be irreducible, then S is a *field*, since every $[a \bmod m]$ with $a \not\equiv 0 \pmod{m}$ has a multiplicative inverse. If $K = \mathbb{Z}_p$ for a prime number p, then we see that S is a field of cardinality p^ℓ.

Chapter 7

The Structure of $\mathbb{Z}_n^*$

We study the structure of the group of units $\mathbb{Z}_n^*$ of the ring $\mathbb{Z}_n$. As we know, $\mathbb{Z}_n^*$ consists of those elements $[a \bmod n] \in \mathbb{Z}_n$ such that a is an integer relatively prime to n.

Suppose $n = p_1^{e_1} \cdots p_r^{e_r}$ is the factorization of n into primes. By the Chinese Remainder Theorem, we have the ring isomorphism

$$\mathbb{Z}_n \cong \mathbb{Z}_{p_1^{e_1}} \times \cdots \times \mathbb{Z}_{p_r^{e_r}}$$

which induces a group isomorphism

$$\mathbb{Z}_n^* \cong \mathbb{Z}_{p_1^{e_1}}^* \times \cdots \times \mathbb{Z}_{p_r^{e_r}}^*.$$

Thus the problem of studying the group of units of modulo an arbitrary integer reduces to the studying the group of units modulo a prime power.

Define $\phi(n)$ to be the cardinality of $\mathbb{Z}_n^*$. This is equal to the number of integers in the interval $\{0, \ldots, n-1\}$ that are relatively prime to n. It is clear that $\phi(p) = p - 1$ for prime p.

Theorem 7.1 *If $n = p_1^{e_1} \cdots p_r^{e_r}$ is the prime factorization of n, then*

$$\phi(n) = p_1^{e_1-1}(p_1 - 1) \cdots p_r^{e_r-1}(p_r - 1).$$

Proof. By the Chinese Remainder Theorem, we have $\phi(n) = \phi(p_1^{e_1}) \cdots \phi(p_r^{e_r})$, so it suffices to show that for a prime power p^e, $\phi(p^e) = p^{e-1}(p-1)$. Now, $\phi(p^e)$ is equal to p^e minus the number of integers in the interval $\{0, \ldots, p^e - 1\}$ that are a multiple of p. The integers in the interval $\{0, \ldots, p^e - 1\}$ that are multiplies of p are precisely $0, p, 2p, 3p, \ldots, (p^{e-1} - 1)p$, of which there are p^{e-1}. Thus, $\phi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$. □

Next, we study the structure of the group $\mathbb{Z}_n^*$. Again, by the Chinese Remainder Theorem, it suffices to consider $\mathbb{Z}_{p^e}^*$ for prime p.

We consider first consider the simpler case $\mathbb{Z}_p^*$.

Theorem 7.2 *$\mathbb{Z}_p^*$ is a cyclic group.*

This theorem follows from the more general theorem:

Theorem 7.3 *Let K be a field and G a subgroup of K^* of finite order. Then G is cyclic.*

Proof. Let n be the order of G, and suppose G is not cyclic. Then by Theorem 4.32, we have that the exponent m of G is strictly less than n. It follows that for all $\alpha \in G$, $\alpha^m = 1_K$. That is, all the elements of G are roots of the polynomial $T^m - 1_K \in K[T]$. But since a polynomial of degree m over a field has at most m roots, this contradicts the fact that $m < n$. □

Now we consider more generally the structure of $\mathbb{Z}_{p^e}^*$. The situation for odd p is described by the following theorem.

Theorem 7.4 *Let p be an odd prime and $e \geq 1$. Then $\mathbb{Z}_{p^e}^*$ is cyclic.*

For $p = 2$, the situation is slightly more complicated:

Theorem 7.5 *The group $\mathbb{Z}_{2^e}^*$ is cyclic for $e = 1$ or 2, but not for $e \geq 3$. For $e \geq 3$, $\mathbb{Z}_{2^e}^*$ is isomorphic to the group $\mathbb{Z}_2 \times \mathbb{Z}_{2^{e-2}}$.*

Before proving these two theorems, we need a few simple facts.

Theorem 7.6 *If p is prime and $0 < k < p$, then the binomial coefficient $\binom{p}{k}$ is divisible by p.*

Proof. By definition

$$\binom{p}{k} = \frac{p!}{k!(p-k)!}.$$

One sees that p divides the numerator, but for $0 < k < p$, p does not divide the denominator. □

Theorem 7.7 *For $e \geq 1$, if $a \equiv b \pmod{p^e}$, then $a^p \equiv b^p \pmod{p^{e+1}}$.*

Proof. We have $a = b + cp^e$ for some $c \in \mathbb{Z}$. Thus, $a^p = b^p + pb^{p-1}cp^e + dp^{2e}$ for an integer d. It follows that $a^p \equiv b^p \pmod{p^{e+1}}$. □

Theorem 7.8 *Let $e \geq 1$ and assume $p^e > 2$. If $a \equiv 1 + p^e \pmod{p^{e+1}}$, then $a^p \equiv 1 + p^{e+1} \pmod{p^{e+2}}$.*

Proof. By Theorem 7.7, $a^p \equiv (1 + p^e)^p \pmod{p^{e+2}}$. Expanding $(1 + p^e)^p$, we have

$$(1 + p^e)^p = 1 + p \cdot p^e + \sum_{k=2}^{p-1} \binom{p}{k} p^{ek} + p^{ep}.$$

Applying Theorem 7.6, all of the terms in the sum on k are divisible by p^{1+2e}, and $1 + 2e \geq e + 2$ for all $e \geq 1$. For the term p^{ep}, the assumption that $p^e > 2$ means that either $p \geq 3$ or $e \geq 2$, which implies $ep \geq e + 2$. □

Now consider Theorem 7.4. Let p be odd and $e > 1$. Let $x \in \mathbb{Z}$ be chosen so that $[x \bmod p]$ generates $\mathbb{Z}_p^*$. Suppose the order of $[x \bmod p^e] \in \mathbb{Z}_{p^e}^*$ is m. Then as $x^m \equiv 1 \pmod{p^e}$ implies $x^m \equiv 1 \pmod{p}$, it must be the case that $p - 1$ divides m, and so $[x^{m/(p-1)} \bmod p^e]$ has order exactly $p - 1$. By Theorem 4.29, if we find an integer y such that $[y \bmod p^e]$ has order p^{e-1}, then $[x^{m/(p-1)} y \bmod p^e]$ has order $(p-1)p^{e-1}$, and we are done. We claim that $y = 1 + p$ does the job. Any integer between 0 and $p^e - 1$ can be expressed as an e-digit number in base p; for example,

$y = (0 \cdots 0\,1\,1)_p$. If we compute successive p-th powers of y modulo p^e, then by Theorem 7.8 we have:

$$\begin{array}{rcll} y \text{ rem } p^e & = & (0 \cdots & 0\,1\,1)_p \\ y^p \text{ rem } p^e & = & (* \cdots & *\,1\,0\,1)_p \\ y^{p^2} \text{ rem } p^e & = & (* \cdots & *\,1\,0\,0\,1)_p \\ & \vdots & & \\ y^{p^{e-2}} \text{ rem } p^e & = & (1\,0 \cdots & 0\,1)_p \\ y^{p^{e-1}} \text{ rem } p^e & = & (0 \cdots & 0\,1)_p \end{array}$$

Here, "$*$" indicates an arbitrary digit. From this table of values, it is clear (c.f., Theorem 4.28) that $[y \bmod p^e]$ has order p^{e-1}. That proves Theorem 7.4.

Now consider Theorem 7.5. For $e = 1$ and $e = 2$, the theorem is clear. Suppose $e \geq 3$. Consider the subgroup $G \subset \mathbb{Z}_{2^e}^*$ generated by $[5 \bmod 2^e]$. Expressing integers between 0 and $2^e - 1$ as e-digit binary numbers, and applying Theorem 7.8, we have:

$$\begin{array}{rcll} 5 \text{ rem } 2^e & = & (0 \cdots & 0\,1\,0\,1)_2 \\ 5^2 \text{ rem } 2^e & = & (* \cdots & *\,1\,0\,0\,1)_2 \\ & \vdots & & \\ 5^{2^{e-3}} \text{ rem } 2^e & = & (1\,0 \cdots & 0\,1)_2 \\ 5^{2^{e-2}} \text{ rem } 2^e & = & (0 \cdots & 0\,1)_2 \end{array}$$

So it is clear (c.f., Theorem 4.28) that $[5 \bmod 2^e]$ has order 2^{e-2}. We claim that $[-1 \bmod 2^e] \notin G$. If it were, then since it has order 2, and since any cyclic group of even order has precisely one element of order 2 (c.f., Theorem 4.24), it must be equal to $[5^{2^{e-3}} \bmod 2^e]$; however, it is clear from the above calculation that $5^{2^{e-3}} \not\equiv -1 \pmod{2^e}$. Let $H \subset \mathbb{Z}_{2^e}^*$ be the subgroup generated by $[-1 \bmod 2^e]$. Then from the above, $G \cap H = \{[1 \bmod 2^e]\}$, and hence by Theorem 4.21, $G \times H$ is isomorphic to the subgroup $G \cdot H$ of $\mathbb{Z}_{2^e}^*$. But since the orders of $G \times H$ and $\mathbb{Z}_{2^e}^*$ are equal, we must have $G \cdot H = \mathbb{Z}_{2^e}^*$. That proves Theorem 7.5.

Chapter 8

Computing Generators and Discrete Logarithms in $\mathbb{Z}_p^*$

As we have seen in the previous chapter, for a prime p, $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$. This means that there exists a generator $\gamma \in \mathbb{Z}_p^*$, such that for all $\alpha \in \mathbb{Z}_p^*$, α can be written uniquely as $\alpha = \gamma^x$ for $0 \leq x < p-1$; the integer x is called the **discrete logarithm** of α to the base γ, and is denoted $\log_\gamma \alpha$.

This chapter discusses some elementary considerations regarding the computational aspects of this situation; namely, how to efficiently find a generator γ, and given γ and α, how to compute $\log_\gamma \alpha$.

More generally, if γ generates a subgroup of $\mathbb{Z}_p^*$ of order q, where $q \mid (p-1)$, and $\alpha \in \langle\gamma\rangle$, then $\log_\gamma \alpha$ is defined to be the unique integer x with $0 \leq x < q$ and $\alpha = \gamma^x$. In some situations it is more convenient to view $\log_\gamma \alpha$ as an element of $\mathbb{Z}_q$. Also for $x \in \mathbb{Z}_q$, with $x = [a \bmod q]$, one may write γ^x to denote γ^a. There can be no confusion, since if $x = [a' \bmod q]$, then $\gamma^{a'} = \gamma^a$. However, in this chapter, we shall view $\log_\gamma \alpha$ as an integer.

8.1 Finding a Generator for $\mathbb{Z}_p^*$

There is no efficient algorithm known for this problem, unless the prime factorization of $p-1$ is given, and even then, we must resort to the use of a probabilistic algorithm.

8.1.1 Probabilistic algorithms

A **probabilistic algorithm** is one that during the course of its execution generates random integers (drawn, say, uniformly from some interval). Generally speaking, the behavior of a probabilistic algorithm depends not only on its input, but also on the particular values of the above-mentioned randomly generated numbers. The running time and output of the algorithm on a given input are properly regarded as random variables. An **efficient** probabilistic algorithm for solving a given problem is one which

- for all inputs, outputs the correct answer with probability very close to 1;
- for all inputs, its *expected* running time is bounded by a polynomial in the input length.

Note that we have not specified in the above requirement just how close to 1 the probability that the output is correct should be. However, it does not really matter (at least, as far as theoretical computer scientists are concerned). If this probability is at least, say, 2/3, then we can make it at least $1 - 2^{-t}$ by running the algorithm $t^{O(1)}$ times, and taking the majority output. The analysis of this "amplification" procedure relies on standard results on the tail of the binomial distribution, which we do not go into here.

A problem of both philosophical and practical interest is the problem of where we get random numbers from. In practice, no one cares: one just uses a reasonably good pseudo-random number generator, and ignores the problem.

8.1.2 Finding a generator

We now present an efficient probabilistic algorithm that takes as input an odd prime p, along with the prime factorization

$$p - 1 = \prod_{i=1}^{r} q_i^{e_i},$$

and outputs a generator for $\mathbb{Z}_p^*$. It runs as follows:

for $i \leftarrow 1$ to r do
    repeat
        choose $\alpha \in \mathbb{Z}_p^*$ at random
        compute $\beta \leftarrow \alpha^{(p-1)/q_i}$
    until $\beta \neq 1$

    $\gamma_i \leftarrow \alpha^{(p-1)/q_i^{e_i}}$

$\gamma \leftarrow \prod_{i=1}^{r} \gamma_i$
output γ

First, let us analyze the correctness of this algorithm. When the ith loop iteration terminates, by construction, we have

$$\gamma_i^{q_i^{e_i}} = 1 \quad \text{but} \quad \gamma_i^{q_i^{e_i - 1}} \neq 1.$$

It follows (c.f., Theorem 4.28) that γ_i has order $q_i^{e_i}$. From this, it follows (c.f., Theorem 4.29) that γ has order $p - 1$.

Thus, we have shown that if the algorithm terminates, its output is always correct.

Let us now analyze the running time of this algorithm. Consider the repeat/until loop in the ith iteration of the outer loop. Since the kernel of the $(p-1)/q_i$-power map on $\mathbb{Z}_p^*$ has order $(p-1)/q_i$, the probability that a random $\alpha \in \mathbb{Z}_p^*$ lies in the kernel is $1/q_i$. It follows that the expected number of iterations of the repeat/until loop is $O(1)$, and therefore, the expected running time of the entire algorithm is $O(r\mathcal{L}(p)^3)$, and since $r \leq \log_2 p$, this is $O(\mathcal{L}(p)^4)$.

Note that if we are not given the prime factorization of $p-1$, but rather, just a prime q dividing $p - 1$, and we want to find an element of order q in $\mathbb{Z}_p^*$, then the above algorithm is easily adapted to this problem. We leave the details as an exercise for the reader.

8.2 Computing Discrete Logarithms $\mathbb{Z}_p^*$

In this section, we consider algorithms for computing the discrete logarithm of $\alpha \in \mathbb{Z}_p^*$ to a given base γ. The algorithms we present here are in the worst case exponential-time algorithms, and are by no means the best possible; however, in some special cases, these algorithms are not so bad.

8.2.1 Brute-force search

Suppose that $\gamma \in \mathbb{Z}_p^*$ generates a subgroup of order q (not necessarily prime), and we are given p, q, γ, and $\alpha \in \langle\gamma\rangle$, and wish to compute $\log_\gamma \alpha$.

The simplest algorithm to solve the problem is **brute-force search**:

$\beta \leftarrow 1$
$i \leftarrow 0$
while $\beta \neq \alpha$ do
  $\beta \leftarrow \beta \cdot \gamma$
  $i \leftarrow i + 1$
output i

This algorithm is clearly correct, and the main loop will always halt after at most q iterations (assuming, as we are, that $\alpha \in \langle\gamma\rangle$). So the total running time is $O(q\mathcal{L}(p)^2)$.

8.2.2 Baby step/giant step method

As above, suppose that $\gamma \in \mathbb{Z}_p^*$ generates a subgroup of order q (not necessarily prime), and we are given p, q, γ, and $\alpha \in \langle\gamma\rangle$, and wish to compute $\log_\gamma \alpha$.

A faster algorithm than brute-force search is the **baby step/giant step method**. It works as follows.

Let us choose an approximation m to $q^{1/2}$. It does not have to be a very good approximation — we just need $m = \Theta(q^{1/2})$. Also, let $m' = \lfloor q/m \rfloor$, so that $m' = \Theta(q^{1/2})$ as well.

The idea is to compute all the values γ^i for $0 \leq i < m$ (the "baby steps") and to build a "lookup table" T that contains all the pairs (γ^i, i). Using an appropriate data structure, such as a *search trie*, we can build the table in time $O(m\mathcal{L}(p)^2)$, and we can perform a lookup in time $O(\mathcal{L}(p))$. By a lookup, we mean that given $\beta \in \mathbb{Z}_p^*$, we can determine if $\beta = \gamma^i$ for some i, and if so, determine the value of i. Let us define $T(\beta) := i$ if $\beta = \gamma^i$ for some i; and otherwise, $T(\beta) := -1$.

After building the lookup table, we execute the following procedure:

$\gamma' \leftarrow \gamma^{-m}$
$\beta \leftarrow \alpha;\ \ j \leftarrow 0;\ \ i \leftarrow T(\beta)$
while $i = -1$ do
  $\beta \leftarrow \beta \cdot \gamma';\ \ j \leftarrow j + 1;\ \ i \leftarrow T(\beta)$

$x \leftarrow jm + i$
output x

To analyze this procedure, suppose that $\alpha = \gamma^x$ for $0 \leq x < q$. Now, x can be written in a unique way as $x = vm + u$, where $0 \leq u < m$ and $0 \leq v \leq m'$. In the jth loop iteration, for

$j = 0, 1, \ldots$, we have

$$\beta = \alpha\gamma^{-mj} = \gamma^{(v-j)m+u}.$$

So we will find that $i \neq -1$ precisely when $j = v$, in which case $i = u$. Thus, the output will be correct, and the total running time of the algorithm is easily seen to be $O(q^{1/2}\mathcal{L}(p)^2)$.

While this algorithm is much faster than brute-force search, it has the drawback that it requires a table of size $O(q^{1/2})$. Of course, there is a "time/space trade-off" here: by choosing m smaller, we get a table of size $O(m)$, but the running time will be proportional to $O(q/m)$.

There are, in fact, algorithms that run (at least heuristically) in time proportional to $O(q^{1/2})$, but which require only a constant amount of space. We do not discuss these algorithms here.

8.2.3 Groups of order q^e

Suppose that $\gamma \in \mathbb{Z}_p^*$ generates a subgroup of order q^e, where $q > 1$ and $e \geq 1$, and we are given p, q, γ, and $\alpha \in \langle\gamma\rangle$, and wish to compute $\log_\gamma \alpha$.

There is a simple algorithm that allows one to reduce this problem to the problem of computing discrete logarithms in a subgroup of order q.

It is perhaps easiest to describe the algorithm recursively.

The base case is when $e = 1$, in which case, we use an algorithm for the subgroup of order q.

Suppose now that $e > 1$. We choose an integer f with $0 < f < e$. Different strategies for choosing f yield different algorithms — we discuss this below. Suppose $\alpha = \gamma^x$, where $0 \leq x < q^e$. Then we can write $x = q^f v + u$, where $0 \leq u < q^f$ and $0 \leq v < q^{e-f}$. Therefore,

$$\alpha^{q^{e-f}} = \gamma^{q^{e-f}u}.$$

Note that $\gamma^{q^{e-f}}$ has order q^f, and so if we recursively compute the discrete logarithm of $\alpha^{q^{e-f}}$ to the base $\gamma^{q^{e-f}}$, we obtain u.

Having obtained u, observe

$$\alpha/\gamma^u = \gamma^{q^f v}.$$

Note also that γ^{q^f} has order q^{e-f}, and so if we recursively compute the discrete logarithm of α/γ^u to the base γ^{q^f}, we obtain u, from which we then compute $x = q^f v + u$.

To analyze the running time of this algorithm, note that we recursively reduce the discrete logarithm problem to a base of order q^e to two discrete logarithm problems: one to a base of order q^f and the other to a base of order q^{e-f}. The running time of the body of one recursive invocation (not counting the running time of the recursive calls it makes) is $O(e \log q \cdot \mathcal{L}(p)^2)$.

To calculate the total running time, we have to sum up the running times of all the recursive calls plus the running times of all the base cases.

Regardless of the strategy for choosing f, the total number of base case invocations is e. Note that for $e > 1$, all the base cases compute discrete logarithms are to the base $\gamma^{q^{e-1}}$. Assuming we implement the base case using the baby step/giant step algorithm, the total running time for all the base cases is therefore $O(eq^{1/2}\mathcal{L}(p)^2)$.

The running time for the recursive calls depends on the strategy used to choose f. If we always choose $f = 1$ or $f = e - 1$, then the running time is for all the recursive calls is $O(e^2 \log q \cdot \mathcal{L}(p)^2)$. However, if we use a "balanced" divide-and-conquer strategy, choosing $f \approx e/2$, then we get $O(e \log e \log q \cdot \mathcal{L}(p)^2)$.

In summary, the total running time is:

$$O((eq^{1/2} + e \log e \log q) \cdot \mathcal{L}(p)^2).$$

8.2.4 Discrete logarithms in $\mathbb{Z}_p^*$

Suppose that we are given a prime p, along with the prime factorization

$$p - 1 = \prod_{i=1}^{r} q_i^{e_i},$$

a generator γ for $\mathbb{Z}_p^*$, and $\alpha \in \mathbb{Z}_p^*$. We wish to compute $\log_\gamma \alpha$.

Suppose that $\alpha = \gamma^x$, where $0 \leq x < p - 1$. Then for $1 \leq i \leq r$,

$$\alpha^{(p-1)/q_i^{e_i}} = \gamma^{(p-1)/q_i^{e_i} x}.$$

Note that $\gamma^{(p-1)/q_i^{e_i}}$ has order $q_i^{e_i}$, and if x_i is the discrete logarithm of $\alpha^{(p-1)/q_i^{e_i}}$ to the base $\gamma^{(p-1)/q_i^{e_i}}$, then we have $0 \leq x_i < q_i^{e_i}$ and $x \equiv x_i \pmod{q_i^{e_i}}$.

Thus, if we compute the values $x_1, \ldots, x_r$, using the algorithm in §8.2.3, we can obtain x using the algorithm of the Chinese Remainder Theorem. If we define $q := \max\{q_i : 1 \leq i \leq r\}$, then the running time of this algorithm will be bounded by $q^{1/2}\mathcal{L}(p)^{O(1)}$.

8.3 Further remarks

One conclusion to be drawn from the observations in this cahpter is that if all the prime factors of $p - 1$ are "small," then the discrete logarithm problem in $\mathbb{Z}_p^*$ is "easy."

The algorithm we have presented here is by no means the fastest. The fastest known algorithm for this problem is based on a technique called the **number field sieve**, and runs in time

$$\exp(\mathcal{L}(P)^{1/3}(\log \mathcal{L}(P))^{2/3}).$$

While this running time is still larger than any polynomial in $\mathcal{L}(P)$, it is still much smaller than that of the simple algorithm presented above.

Finally, we remark that all of the algorithms presented in this chapter work in *any* finite cyclic group — we really did not exploit any properties about $\mathbb{Z}_p^*$ other than the fact that it is a cyclic group. However, faster discrete logarithm algorithms, like those mentioned above based on the number field sieve, do not work in an arbitrary finite cyclic group; these algorithms only work for $\mathbb{Z}_p^*$, and more generally, for K^*, where K is a finite field.

Chapter 9

Quadratic Residues and Quadratic Reciprocity

9.1 Quadratic Residues

For positive integer n, an integer a is called a **quadratic residue modulo** n if $\gcd(a, n) = 1$ and $x^2 \equiv a \pmod{n}$ for some integer x; in this case, we say that x is a **square root of** a **modulo** n.

The quadratic residues modulo n correspond exactly to the subgroup of squares $(\mathbb{Z}_n^*)^2$ of $\mathbb{Z}_n^*$; that is, a is a quadratic residue modulo n if and only if $[a \bmod n] \in (\mathbb{Z}_n^*)^2$.

Let us first consider the case where $n = p$, where p is an odd prime. In this case, we know that $\mathbb{Z}_p^*$ is cyclic of order $p - 1$. Recall that the subgroups any finite cyclic group are in one-to-one correspondence with the divisors of the order of the group.

For any $d \mid (p - 1)$, consider the d-power map on $\mathbb{Z}_p^*$ that sends $\alpha \in \mathbb{Z}_p^*$ to α^d. The image of this map is the unique subgroup of $\mathbb{Z}_p^*$ of order $(p - 1)/d$, and the kernel of this map is the unique subgroup of order d (c.f., Theorem 4.24). This means that the image of the 2-power map is of order $(p - 1)/2$ and must be the same as the kernel of the $(p - 1)/2$-power map. Since the image of the $(p - 1)/2$-power map is of order 2, it must be equal to the subgroup $\{[\pm 1 \bmod p]\}$. The kernel of the 2-power map is of order 2, and so must also be equal to the subgroup $\{[\pm 1 \bmod p]\}$.

Translating from group-theoretic language to the language of congruences, we have shown:

Theorem 9.1 *For an odd prime p, the number of quadratic residues a modulo p, with $0 < a < p$, is $(p - 1)/2$. Moreover, if x is a square root of a modulo p, then so is $-x$, and any square root y of a modulo p satisfies $y \equiv \pm x \pmod{p}$. Also, for any integer $a \not\equiv 0 \pmod{p}$, we have $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$, and moreover, a is a quadratic residue modulo p if and only if $a^{(p-1)/2} \equiv 1 \pmod{p}$.*

Now consider the case where $n = p^e$, where p is an odd prime and $e > 1$. We also know that $\mathbb{Z}_{p^e}^*$ is a cyclic group of order $p^{e-1}(p - 1)$, and so everything that we said in discussing the case $\mathbb{Z}_p^*$ applies here as well. Thus, for $a \not\equiv 0 \pmod{p}$, a is a quadratic residue modulo p^e if and only if $a^{p^{e-1}(p-1)/2} \equiv 1 \pmod{p^e}$. However, we can simplify this a bit. Note that $a^{p^{e-1}(p-1)/2} \equiv 1 \pmod{p^e}$ implies $a^{p^{e-1}(p-1)/2} \equiv 1 \pmod{p}$, and by Theorem 4.23 (Fermat's Little Theorem), this implies $a^{(p-1)/2} \equiv 1 \pmod{p}$. Conversely, by Theorem 7.7, $a^{(p-1)/2} \equiv 1 \pmod{p}$ implies $a^{p^{e-1}(p-1)/2} \equiv 1 \pmod{p^e}$. Thus, we have shown:

Theorem 9.2 *For an odd prime p and positive integer e, the number of quadratic residues a modulo p^e, with $0 < a < p^e$, is $p^{e-1}(p - 1)/2$. Moreover, if x is a square root of a modulo p^e, then* *so is $-x$, and any square root y of a modulo p^e satisfies $y \equiv \pm x \pmod{p^e}$. Also, for any integer $a \not\equiv 0 \pmod{p}$, we have $a^{p^{e-1}(p-1)/2} \equiv \pm 1 \pmod{p}$, and moreover, a is a quadratic residue modulo p^e iff $a^{p^{e-1}(p-1)/2} \equiv 1 \pmod{p^e}$ iff $a^{(p-1)/2} \equiv 1 \pmod{p}$ iff a is a quadratic residue modulo p.*

Now consider an arbitary odd positive integer n. Let $n = \prod_{i=1}^{r} p_i^{e_i}$ be its prime factorization. Recall the group isomorphism implied by the Chinese Remainder Theorem:

$$\mathbb{Z}_n^* \cong \mathbb{Z}_{p_1^{e_1}}^* \times \cdots \times \mathbb{Z}_{p_r^{e_r}}^*.$$

Now,

$$(\alpha_1, \ldots, \alpha_r) \in \mathbb{Z}_{p_1^{e_1}}^* \times \cdots \times \mathbb{Z}_{p_r^{e_r}}^*$$

is a square if and only if there exist $\beta_1, \ldots, \beta_r$ with $\beta_i \in \mathbb{Z}_{p_i^{e_i}}^*$ and $\alpha_i = \beta_i^2$ for $1 \le i \le k$, in wich case, we see that the square roots of $(\alpha_1, \ldots, \alpha_r)$ comprise the 2^r elements $(\pm\beta_1, \ldots, \pm\beta_r)$. Thus we have:

Theorem 9.3 *Let n be odd positive integer n with prime factorization $n = \prod_{i=1}^{r} p_i^{e_i}$. The number of quadratic residues a modulo n, with $0 < a < n$, is $\phi(n)/2^r$. Moreover, if a is a quadratic residue modulo n, then there are precisely 2^r distinct integers x, with $0 < x < n$, such that $x^2 \equiv a \pmod{n}$. Also, an integer a is a quadratic resisdue modulo n if and only if it is a quadratic residue modulo p_i for $1 \le i \le r$.*

That completes our investigation of the case where n is an odd positive integer. We shall not investigate the case where n is even, as it is a bit cumbersome, and is not of particular importance.

9.2 The Legendre Symbol

For an odd prime p and an integer a with $\gcd(a, p) = 1$, the **Legendre symbol** $(a \mid p)$ is defined to be 1 if a is a quadratic residue modulo p, and -1 otherwise. For completeness, one defines $(a \mid p) = 0$ if $p \mid a$.

Theorem 9.4 *Let p be an odd prime, and let $a, b \in \mathbb{Z}$, both not divisible by p. Then*

1. *$(a \mid p) \equiv a^{(p-1)/2} \pmod{p}$; in particular, $(-1 \mid p) = (-1)^{(p-1)/2}$;*
2. *$(a \mid p)(b \mid p) = (ab \mid p)$;*
3. *$a \equiv b \pmod{p}$ implies $(a \mid p) = (b \mid p)$;*
4. *$(2 \mid p) = (-1)^{(p^2-1)/8}$;*
5. *if q is an odd prime different from p, then*

$$(p \mid q)(q \mid p) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}.$$

Part (5) of this theorem is called the Law of Quadratic Reciprocity.

Part (1) follows from Theorem 9.1. Part (2) is an immediate cosequence of part (1), and part (3) is clear from the definition.

The rest of this section is devoted to a proof of parts (4) and (5) of this theorem. The proof is completely elementary, although a bit technical. The proof we present here is taken almost verbatim from Niven and Zuckerman's book, *An Introduction to the Theory of Numbers.*

Theorem 9.5 (Gauss' Lemma) *Let p be an odd prime and a relatively prime to p. Define $\alpha_j := ja \text{ rem } p$ for $1 \le j \le (p-1)/2$, and let n be the number of indices j for which $\alpha_j > p/2$. Then $(a \mid p) = (-1)^n$.*

Proof. Let $r_1, \ldots, r_n$ denote the α_j's exceeding $p/2$, and let $s_1, \ldots, s_k$ denote the remaining α_j's. The r_i and s_i are all distinct and non-zero. We have $0 < p - r_i < p/2$ for $1 \le i \le n$, and no $p - r_i$ is an s_j; indeed, if $p - r_i = s_j$, then $s_j \equiv -r_j \pmod{p}$, and writing $s_j = k_1 a$ and $r_j = k_2 a$ for $1 \le k_1, k_2 \le (p-1)/2$, we have $k_1 a \equiv -k_2 a \pmod{p}$, which implies $k_1 \equiv -k_2 \pmod{p}$, which is impossible.

It follows that the sequence of numbers $s_1, \ldots, s_k, p - r_1, \ldots, p - r_n$ is just a re-ordering of $1, \ldots, (p-1)/2$. Then we have

$$((p-1)/2)! \equiv s_1 \cdots s_k (-r_1) \cdots (-r_n) \equiv (-1)^n s_1 \cdots s_k r_1 \cdots r_n \equiv (-1)^n ((p-1)/2)! a^{(p-1)/2} \pmod{p},$$

and cancelling the factor $((p-1)/2)!$, we obtain $a^{(p-1)/2} \equiv (-1)^n \pmod{p}$, and the result follows from the fact that $(a \mid p) \equiv a^{(p-1)/2} \pmod{p}$. □

Theorem 9.6 *If p is an odd prime and $\gcd(a, 2p) = 1$, then $(a \mid p) = (-1)^t$ where $t = \sum_{j=1}^{(p-1)/2} \lfloor ja/p \rfloor$. Also, $(2 \mid p) = (-1)^{(p^2-1)/8}$.*

Proof. Let a be an integer relatively prime to p (not necessarily odd), and let us adopt the same notation as in the proof of Theorem 9.5. Note that $ja = p\lfloor ja/p \rfloor + \alpha_j$, for $1 \le j \le k$, so we have

$$\sum_{j=1}^{(p-1)/2} ja = \sum_{j=1}^{(p-1)/2} p\lfloor ja/p \rfloor + \sum_{j=1}^{n} r_j + \sum_{j=1}^{k} s_j.$$

Also, we saw in the proof of Theorem 9.5 that the integers $s_1, \ldots, s_k, p - r_1, \ldots, p - p_n$ are a re-ordering of $1, \ldots, (p-1)/2$, and hence

$$\sum_{j=1}^{(p-1)/2} j = \sum_{j=1}^{n} (p - r_j) + \sum_{j=1}^{k} s_j = np - \sum_{j=1}^{n} r_j + \sum_{j=1}^{k} s_j.$$

Subtracting, we get

$$(a-1) \sum_{j=1}^{(p-1)/2} j = p \left(\sum_{j=1}^{(p-1)/2} \lfloor ja/p \rfloor - n \right) + 2 \sum_{j=1}^{n} r_j.$$

Note that

$$\sum_{j=1}^{(p-1)/2} j = \frac{p^2-1}{8},$$

which implies

$$(a-1) \frac{p^2-1}{8} \equiv \sum_{j=1}^{(p-1)/2} \lfloor ja/p \rfloor - n \pmod{2}.$$

If a is odd,this implies

$$n \equiv \sum_{j=1}^{(p-1)/2} \lfloor ja/p \rfloor \pmod{2}.$$

If $a = 2$, this — along with the fact that $\lfloor 2j/p \rfloor = 0$ for $1 \le j \le (p-1)/2$ — implies

$$n \equiv \frac{p^2-1}{8} \pmod{2}.$$

The theorem now follows from Theorem 9.5. □

Note that this last theorem proves part (4) of Theorem 9.4. The next theorem proves part (5).

Theorem 9.7 *If p and q are distinct odd primes, then*

$$(p \mid q)(q \mid p) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}.$$

Proof. Let S be the set of pairs of integers (x, y) with $1 \le x \le (p-1)/2$ and $1 \le y \le (q-1)/2$. Note that S contains no pair (x, y) with $qx = py$, so let us partition S into two subsets: S_1 contains all pairs (x, y) with $qx > py$, and S_2 contains all pairs (x, y) with $qx < py$. Note that $(x, y) \in S_1$ if and only if $1 \le x \le (p-1)/2$ and $1 \le y \le \lfloor qx/p \rfloor$. So $|S_1| = \sum_{x=1}^{(p-1)/2} \lfloor qx/p \rfloor$. Similarly, $|S_2| = \sum_{y=1}^{(q-1)/2} \lfloor py/q \rfloor$. So we have

$$\frac{p-1}{2}\frac{q-1}{2} = |S| = |S_1| + |S_2| = \sum_{x=1}^{(p-1)/2} \lfloor qx/p \rfloor + \sum_{y=1}^{(q-1)/2} \lfloor py/q \rfloor,$$

and Theorem 9.6 implies

$$(p \mid q)(q \mid p) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}.$$

That proves the first statement of the theorem. The second statement follows immediately. □

9.3 The Jacobi Symbol

Let a, n be integers, where n is positive and odd, so that $n = q_1 \cdots q_k$, where the q_i are odd primes, not necessarily distinct. Then the **Jacobi symbol** $(a \mid n)$ is defined as

$$(a \mid n) \;:=\; (a \mid q_1) \cdots (a \mid q_k),$$

where $(a \mid q_j)$ is the Legendre symbol. Note that $(a \mid 1) = 1$ for all $a \in \mathbb{Z}$. Thus, the Jacobi symbol essentially extends the domain of definition of the Legendre symbol. Note that $(a \mid n) \in \{0, \pm 1\}$.

Theorem 9.8 *Let m, n be positive, odd integers, an let a, b be integers. Then*

1. $(ab \mid n) = (a \mid n)(b \mid n)$;
2. $(a \mid mn) = (a \mid m)(a \mid n)$;
3. $a \equiv b \pmod{n}$ *imples* $(a \mid n) = (b \mid n)$;
4. $(-1 \mid n) = (-1)^{(n-1)/2}$;
5. $(2 \mid n) = (-1)^{(n^2-1)/8}$;
6. *if* $\gcd(m, n) = 1$, *then*
$$(m \mid n)(n \mid m) = (-1)^{\frac{m-1}{2}\frac{n-1}{2}}.$$

Proof. Parts (1)–(3) follow directly from the definition (exercise).

For parts (4) and (6), one can easily verify (exercise) that for odd integers $n_1, \ldots, n_k$,

$$\sum_{i=1}^{k} (n_i - 1)/2 \equiv (n_1 \cdots n_k - 1)/2 \pmod{2}.$$

Part (4) easily follows from this fact, along with part (2) of this theorem and part (1) of Theorem 9.4 (exercise). Part (6) easily follows from this fact, along with parts (1) and (2) of this theorem, and part (5) of Theorem 9.4 (exercise).

For part (5), one can easily verify (exercise) that for odd integers $n_1, \ldots, n_k$,

$$\sum_{1 \leq i \leq k} (n_i^2 - 1)/8 \equiv (n_1^2 \cdots n_k^2 - 1)/8 \pmod{2}.$$

Part (5) easily follows from this fact, along with part (2) of this theorem, and part (4) of Theorem 9.4 (exercise). □

As we shall see later, this theorem is extremely useful from a computational point of view — with it, one can efficiently compute $(a \mid n)$, without having to know the prime factorization of either a or n. Also, in applying this theorem it is useful to observe that for odd integers m, n,

- $(-1)^{(n-1)/2} = 1$ iff $n \equiv 1 \pmod{4}$;
- $(-1)^{(n^2-1)/8} = 1$ iff $n \equiv \pm 1 \pmod{8}$;
- $(-1)^{((m-1)/2)((n-1)/2)} = 1$ iff $m \equiv 1 \pmod{4}$ or $n \equiv 1 \pmod{4}$.

Finally, we note that if a is a quadratic residue modulo n, then $(a \mid n) = 1$; however, $(a \mid n) = 1$ does not imply that a is a quadratic residue modulo n.

Chapter 10

Computational Problems Related to Quadratic Residues

10.1 Computing the Jacobi Symbol

Suppose we are given an odd, positive integer n, along with an integer a, and we want to compute the Jacobi symbol $(a \mid n)$. Theorem 9.8 suggests the following algorithm:

```
t ← 1
repeat
    — loop invariant: n is odd and positive

    a ← a rem n
    if a = 0
        if n = 1 return t else return 0

    compute a′, h such that a = 2^h a′ and a′ is odd
    if h ≢ 0 (mod 2) and n ≢ ±1 (mod 8) then t ← −t
    if a′ ≢ 1 (mod 4) and n ≢ 1 (mod 4) then t ← −t
    (a, n) ← (n, a′)
forever
```

That this algorithm correctly computes the Jacobi symbol $(a \mid n)$ follows directly from Theorem 9.8. Using an analysis similar to that of Euclid's algorithm, one easily sees that the running time of this algorithm is $O(\mathcal{L}(a)\mathcal{L}(n))$.

10.2 Testing quadratic residuosity

10.2.1 Prime modulus

For an odd prime p, we can test if a is a quadratic residue modulo p by either performing the exponentiation $a^{(p-1)/2}$ rem p or by computing the Legendre symbol $(a \mid p)$. Using a standard repeated squaring algorithm, the former method takes time $O(\mathcal{L}(p)^3)$, while using the Euclidean-like algorithm of the previous section, the latter method takes time $O(\mathcal{L}(p)^2)$. So presumably, the latter method is to be preferred.

10.2.2 Prime-power modulus

For an odd prime p, we know that a is a quadratic residue modulo p^e if and only if a is a quadratic residue modulo p. So this case immediately reduces to the previous case.

10.2.3 Composite modulus

For odd, composite n, if we know the factorization of n, then we can also determine if a is a quadratic residue modulo n by determining if it is a quadratic residue modulo each prime divisor p of n. However, without knowledge of this factorization (which is in general believed to be hard to compute), there is no efficient algorithm known. We can compute the Jacobi symbol $(a \mid n)$; if this is -1 or 0, we can conclude that a is not a quadratic residue; otherwise, we cannot conclude much of anything.

10.3 Computing modular square roots

10.3.1 Prime modulus

Let p be an odd prime, and suppose that $(a \mid p) = 1$. Here is one way to compute a square root of a modulo p, assuming we have at hand an integer y such that $(y \mid p) = -1$.

Let $\alpha = [a \bmod p] \in \mathbb{Z}_p^*$ and $\gamma = [y \bmod p] \in \mathbb{Z}_p^*$. The above problem is equivalent to finding $\beta \in \mathbb{Z}_p^*$ such that $\beta^2 = \alpha$.

Let us write $p - 1 = 2^h m$, where m is odd. For any $\delta \in \mathbb{Z}_p^*$, δ^m has order dividing 2^h. Since $\alpha^{2^{h-1}m} = 1$, α^m has order dividing 2^{h-1}. Since $\gamma^{2^{h-1}m} = [-1 \bmod p]$, γ^m has order precisely 2^h. Since there is only one subgroup in $\mathbb{Z}_p^*$ of order 2^h, it follows that γ^m generates this subgroup, and that $\alpha^m = \gamma^{mx}$ for $0 \leq x < 2^h$ and x is even. We can find x by computing the discrete logarithm of α^m to the base γ^m, using the algorithm in §8.2.3. Setting $\kappa = \gamma^{mx/2}$, we have

$$\kappa^2 = \alpha^m.$$

We are not quite done, since we now have a square root of α^m, and not of α. However, because $\gcd(m, 2) = 1$, we can find integers s, t such that $ms + 2t = 1$. In fact, $s = 1$ and $t = -\lfloor m/2 \rfloor$ do the job. It then follows that

$$(\kappa^s \alpha^t)^2 = \kappa^{2s} \alpha^{2t} = \alpha^{ms} \alpha^{2t} = \alpha^{ms+2t} = \alpha.$$

Thus, $\kappa^s \alpha^t$ is a square root of α.

The total amount of work done outside the discrete logarithm calculation amounts to just a handful of exponentiations modulo p, and so takes time $O(\mathcal{L}(p)^3)$. The time to compute the discrete logarithm is $O(h \log h \mathcal{L}(p)^2)$. So the total running time of this procedure is

$$O(\mathcal{L}(p)^3 + h \log h \mathcal{L}(p)^2).$$

The above procedure assumed we had at hand a non-square γ. If $h = 1$, i.e., $p \equiv 3 \pmod 4$, then -1 is a quadratic residue modulo p, and so we are done. In fact, in this case, the the output of the above procedure is simply $\alpha^{(p+1)/4}$, no matter what value of γ is used. One can easily show directly that $\alpha^{(p+1)/4}$ is a square root of α, without analyzing the above procedure.

If $h > 1$, we can find a non-square γ using a probabilistic algorithm. Simply choose γ at random, test if it is a square, and repeat if not. The probability that a random element of $\mathbb{Z}_p^*$ is a square is $1/2$; thus, the expected number of trials is $O(1)$, and hence the expected running time of this probabilistic algorithm is $O(\mathcal{L}(p)^2)$.

10.3.2 Prime-power modulus

Again, for an odd prime p, we know that a is a quadratic residue modulo p^e if and only if a is a quadratic residue modulo p.

Suppose we have found an integer z such that $z^2 \equiv a \pmod{p}$, using, say, the procedure described above. From this, we can easily compute a square root of a modulo p^e using the following technique, which is known as **Hensel lifting**.

More generally, suppose we have integers a, z such that $z^2 \equiv a \pmod{p^f}$, for $f \geq 1$, and we want to find an integer $\hat{z}$ such that $\hat{z}^2 \equiv a \pmod{p^{f+1}}$. Clearly, if $\hat{z}^2 \equiv a \pmod{p^{f+1}}$, then $\hat{z}^2 equiva \pmod{p^f}$, and so $\hat{z} \equiv \pm z \pmod{p^f}$. So let us set $\hat{z} = z + up^f$, and solve for u. We have

$$\hat{z}^2 \equiv (z + up^f)^2 \equiv z^2 + 2p^f u + u^2 p^{2f} \equiv z^2 + 2p^f u \pmod{p^{f+1}}.$$

So we want to find integer u such that

$$2p^f u \equiv a - z^2 \pmod{p^{f+1}}.$$

Since $p^f \mid (z^2 - a)$, by Theorem 2.3, the above congruence holds if and only if

$$2u \equiv \frac{a - z^2}{p^f} \pmod{p}.$$

From this, we can easily compute the desired value u.

By iterating the above procedure, starting with a square root of a modulo p, we can quickly find a square root of a modulo p^e. We leave a detailed analysis of the running time of this procedure to the reader.

10.3.3 Composite modulus

To find square roots modulo n, where n is an odd composite modulus, if we know the prime factorization of n, then we can use the above procedures for finding square roots modulo primes and prime powers, and then use the algorithm of the Chinese Remainder Theorem to get a square root modulo n.

However, if the factorization of n is not known, then there is no efficient algorithm known for computing square roots modulo n. In fact, one can show that the problem of finding square roots modulo n is at least as hard as the problem of factoring n, in the sense that if there is an efficient algorithm for computing square roots modulo n, then there is an efficient (probabilistic) algorithm for factoring n.

Here is an algorithm to find a non-trivial divisor of n — it uses a square root-algorithm as a subroutine. Choose $z \in \{1, \ldots, n-1\}$ at random. If $\gcd(z, n) > 1$, then output $\gcd(z, n)$. Otherwise, set $a := z^2 \text{ rem } n$, and feed a and n to the square-root algorithm. If the square-root algorithm returns an integer z', and $z' \equiv \pm z \pmod{n}$, then output "failure"; otherwise, output $\gcd(z - z', n)$, which is a non-trivial divisor of n.

To analyze this algorithm, let us just consider the case where $n = pq$, and p and q are distinct primes. If $\gcd(z, n) > 1$, we split n, so assume that $\gcd(z, n) = 1$. In this case, $[z \bmod n]$ is uniformly distributed over $\mathbb{Z}_n^*$, and $[a \bmod n]$ is uniformly distributed over $(\mathbb{Z}_n^*)^2$. Let us condition on an *fixed* value of a. In this conditional probability space, $[z \bmod n]$ is uniformly distributed over the four square roots of a, which under the isomorphism of the Chinese Remainder Theorem, correspond to

$$([\pm z \bmod p], [\pm z \bmod q]) \in \mathbb{Z}_p \times \mathbb{Z}_q.$$

Since the square-root algorithm receives no information about z other than the value a, the probability that $z' \equiv \pm z \pmod{n}$ is $1/2$, in which case we output "failure"; however, if $z' \not\equiv \pm z$, then we have either

$$z' \equiv z \pmod{p} \quad \text{and} \quad z' \equiv -z \pmod{q}$$

or

$$z' \equiv -z \pmod{p} \quad \text{and} \quad z' \equiv z \pmod{q}.$$

In the first case, $\gcd(z - z', n) = p$, and in the second case $\gcd(z - z', n) = q$; in either case, we split n.

That completes the analysis in the case where $n = pq$. In general, one can show that for any odd n that is not a prime power, the above procedure will find a non-trivial factor of n into with probability at least $1/2$. With this, it is easy to obtain an efficient probabilistic algorithm that completely factors n.

Chapter 11

Primality Testing

In this chapter, we discuss some simple tests for primality, and also mention some results on the distribution of primes.

11.1 Trial Division

Suppose we are given a number n, and we want to determine if n is prime or composite. The simplest algorithm to describe and to program is **trial division**. We simply divide n by 2, 3, and so on, testing if any of these numbers evenly divide n. Of course, we don't need to go any farther than $\sqrt{n}$, since if n has any nontrivial factors, it must have one that is no greater than $\sqrt{n}$. Other small optimizations are also possible; for example, we don't have to test multiples of 2 other than 2, multiples of 3 other than 3, and so on.

This algorithm requires $O(\sqrt{n})$ arithmetic operations, which is exponential in the length of n. Thus, for practical purposes, this algorithm is limited to quite small n. Suppose, for example, that n has 100 decimal digits, and that a computer can perform 1 billion divisions per second (this is much faster than any computer existing today). Then it would take 3×10^{35} *years* to perform $\sqrt{n}$ divisions.

In the next section, we discuss a much faster primality test that allows 100 decimal digit numbers to be tested for primality less than a second. Unlike the above test, however, this test does not find a factor of n when n is composite.

11.2 A Fast Probabilistic Test

We describe in this section a fast (polynomial time) test for primality, known as the **Miller-Rabin algorithm**. The algorithm, however, is probabilistic, and may (with small probability) make a mistake.

We assume for the remainder of this section that the number n we are testing for primality is odd.

Several probabilistic primality tests, including the the Miller-Rabin algorithm, have the following general structure. Define $\mathbb{Z}_n^+$ to be the set of non-zero elements of $\mathbb{Z}_n$; thus, $|\mathbb{Z}_n^+| = n - 1$ and if n is prime, $\mathbb{Z}_n^+ = \mathbb{Z}_n^*$. Suppose also that we define a set $L_n \subset \mathbb{Z}_n^+$ such that

- there is an efficient algorithm that on input n and $\alpha \in \mathbb{Z}_n^+$, determines if $\alpha \in L_n$;
- if n is prime, then $L_n = \mathbb{Z}_n^*$;

- if n is composite, $|L_n| \leq (n-1)/2$.

To test n for primality, we set an "error parameter" t, and choose random elements $\alpha_1, \ldots, \alpha_t \in \mathbb{Z}_n^+$. If $\alpha_i \in L_n$ for all $1 \leq i \leq t$, then we output "prime"; otherwise, we output "composite."

It is easy to see that if n is prime, this algorithm always outputs "prime," and if n is composite this algorithm outputs "composite" with probability at least $1 - 2^t$. If t is chosen large enough, say $t = 100$, then the probability that the output is wrong is so small that for all practical purposes, it is "just as good as zero."

We now make a first attempt at defining a suitable set L_n. Let us define $L_n = \{\alpha \in \mathbb{Z}_n^+ : \alpha^{n-1} = 1\}$. Note that $L_n \subset \mathbb{Z}_n^*$, since if $\alpha^{n-1} = 1$, then α has a multiplicative inverse, namely, α^{n-2}. Using a repeated-squaring algorithm, we can test if $\alpha \in L_n$ in time $O(\lg(n)^3)$.

Theorem 11.1 *If n is prime, then $L_n = \mathbb{Z}_n^*$. If n is composite and $L_n \subsetneq \mathbb{Z}_n^*$, $|L_n| \leq (n-1)/2$.*

Proof. Note that L_n is the kernel of the $(n-1)$-power map on $\mathbb{Z}_n^*$, and hence is a subgroup of $\mathbb{Z}_n^*$.

If n is prime, then we know that $\mathbb{Z}_n^*$ is a group of order $n-1$. Hence, $\alpha^{n-1} = 1$ for all $\alpha \in \mathbb{Z}_n^*$. That is, $L_n = \mathbb{Z}_n^*$.

Suppose that n is composite and $L_n \subsetneq \mathbb{Z}_n^*$. Since the order of a subgroup divides the order of the group, we have $|\mathbb{Z}_n^*| = m|L_n|$ for some integer $m > 1$. From this, we conclude that

$$|L_n| = \frac{1}{m}|\mathbb{Z}_n^*| \leq \frac{1}{2}|\mathbb{Z}_n^*| \leq \frac{n-1}{2}.$$

□

Unfortunately, there are odd composite numbers n such that $L_n = \mathbb{Z}_n^*$. The smallest such number is

$$561 = 3 \cdot 11 \cdot 17.$$

Such numbers are called **Carmichael numbers**. They are extremely rare, but it is known that there are infinitely many of them, so we can not ignore them.

The following theorem characterizes Carmichael numbers.

Theorem 11.2 *A positive odd integer n is a Carmichael number if and only if it is square-free of the form $n = p_1 \cdots p_r$, where $(p_i - 1) \mid (n-1)$ for $1 \leq i \leq r$.*

Proof. Suppose $n = p_1^{e_1} \cdots p_r^{e_r}$. By the Chinese Remainder Theorem, we have an isomorphism of $\mathbb{Z}_n^*$ with the group

$$\mathbb{Z}_{p_1^{e_1}}^* \times \cdots \times \mathbb{Z}_{p_k^{e_k}}^*,$$

and we know that each group $\mathbb{Z}_{p_i^{e_i}}^*$ is cyclic of order $p_i^{e_i-1}(p_i - 1)$. Thus, the $(n-1)$-power map annihilates the group $\mathbb{Z}_n^*$ if and only if it annihilates each of the groups $\mathbb{Z}_{p_i^{e_i}}^*$, which occurs if and only if $p_i^{e_i-1}(p_i-1) \mid (n-1)$. Now, on the one hand, $n \equiv 0 \pmod{p_i}$. On the other hand, if $e_i > 1$, we would have $n \equiv 1 \pmod{p_i}$, which is clearly impossible. Thus, we must have $e_i = 1$. □

To obtain a good primality test, we need to define a different set L_n', which we do as follows. Let $n - 1 = 2^h m$, where m is odd (and $h \geq 1$ since n is assumed odd). Then $\alpha \in L_n'$ if and only if $\alpha^m = 1$ or $\alpha^{m2^i} = [-1 \bmod n]$ for some $0 \leq i < h$.

The Miller-Rabin algorithm uses this set L_n', in place of the set L_n defined above.

Note that L_n' is a subset of L_n: if $\alpha^m = 1$, then certainly $\alpha^{n-1} = (\alpha^m)^{2^h} = 1$, and if $\alpha^{m2^i} = [-1 \bmod n]$ for some $0 \leq i < h$, then $\alpha^{n-1} = (\alpha^{m2^i})^{2^{h-i}} = 1$.

As a first step in analyzing the Miller-Rabin algorithm, we prove the following:

Theorem 11.3 *Let n be a Carmichael number, and suppose $n = p_1 \cdots p_r$. Let $n-1 = 2^h m$, where m is odd, and for $1 \le i \le r$, let $p_i - 1 = 2^{h_i} m_i$, where m_i is odd. Let $h' = \max\{h_i\}$, and define $P_n := \{u \in \mathbb{Z}_n^* : u^{m2^{h'-1}} = [\pm 1 \bmod n]\}$. Then we have:*

(i) $h' \le h$;

(ii) for all $u \in \mathbb{Z}_n^*$, $u^{m2^{h'}} = 1$;

(iii) P_n *is a subgroup of* $\mathbb{Z}_n^*$, *and* $P_n \subsetneq \mathbb{Z}_n^*$.

Proof. As n is Carmichael, each $p_i - 1$ divides $n-1$. It follows that $h' \le h$. That proves (i). It also follows that $m_i \mid m$ for each i.

Again, by the Chinese Remainder Theorem, we have an isomorphism of $\mathbb{Z}_n^*$ with the group $\mathbb{Z}_{p_1}^* \times \cdots \times \mathbb{Z}_{p_r}^*$, where each $\mathbb{Z}_{p_i}^*$ is cyclic of order $p_i - 1$.

Since each $p_i - 1$ divides $m2^{h'}$, it follows that each $\mathbb{Z}_{p_i}^*$ is annihilated by the $(m2^{h'})$-power map. It follows from the Chinese Remainder Theorem that $\mathbb{Z}_n^*$ is also annihilated by the $(m2^{h'})$-power map. That proves (ii).

To prove (iii), first note that P_n is the pre-image of the subgroup $\{[\pm 1 \bmod n]\}$ under the $(m2^{h'-1})$-power map, and hence is itself a subgroup of $\mathbb{Z}_n^*$. Now, $h' = h_i$ for some i, and without loss of generality, assume $i = 1$. Let $\alpha = [a \bmod p_1] \in \mathbb{Z}_{p_1}^*$ be a generator for $\mathbb{Z}_{p_1}^*$. Since α has order $m_1 2^{h'}$, it follows that $\alpha^{m_1 2^{h'-1}}$ has order 2, which means that $\alpha^{m_1 2^{h'-1}} = [-1 \bmod p_1]$. Since $m_1 \mid m$ and m is odd, it follows that $\alpha^{m2^{h'-1}} = [-1 \bmod p_1]$. By the Chinese Remainder Theorem, there exists an integer b such that $b \equiv a \pmod{p_1}$ and $b \equiv 1 \pmod{p_j}$ for $j \ne 1$. We claim that $b^{m2^{h'-1}} \not\equiv \pm 1 \pmod{n}$. Indeed, if $b^{m2^{h'-1}} \equiv 1 \pmod{n}$, then we would have $b^{m2^{h'-1}} \equiv 1 \pmod{p_1}$, which is not the case, and if $b^{m2^{h'-1}} \equiv -1 \pmod{n}$, then we would have $b^{m2^{h'-1}} \equiv -1 \pmod{p_2}$, which is also not the case. That proves $P_n \subsetneq \mathbb{Z}_n^*$. □

From the above theorem, we can easily derive the following result:

Theorem 11.4 *If n is prime, then $L'_n = \mathbb{Z}_n^*$. If n is composite, then $|L'_n| \le (n-1)/2$.*

Proof. Let $n - 1 = m2^h$, where m is odd. For $\alpha \in \mathbb{Z}_n^*$, let define the sequence of group elements $s_i(\alpha) := \alpha^{m2^i}$ for $0 \le i \le h$. We can characterize the set L'_n as follows: it consists of all $\alpha \in \mathbb{Z}_n^*$ such that $s_h(\alpha) = [1 \bmod n]$, and for $1 \le i \le h$, $s_i(\alpha) = [1 \bmod n]$ implies $s_{i-1}(\alpha) = [\pm 1 \bmod n]$.

First, suppose n is prime. By Fermat's little theorem, for $\alpha \in \mathbb{Z}_n^*$, we know that $s_h(\alpha) = [1 \bmod n]$. Moreover, if $s_i(\alpha) = [1 \bmod n]$ for $1 \le i \le h$, then as $s_{i-1}(\alpha)^2 = [1 \bmod n]$, and the only square roots of $[1 \bmod n]$ are $[\pm 1 \bmod n]$, we have $s_{i-1}(\alpha) = [\pm 1 \bmod n]$.

Next, suppose n is composite but is not a Carmichael number. Then the theorem follows from Theorem 11.1 and the fact that $L'_n \subset L_n$.

Finally, suppose that n is a Carmichael number. We claim that $L'_n \subset P_n$, where P_n is as defined in Theorem 11.3. To prove this, let $\alpha \in \mathbb{Z}_n^*$. Then if h' is as defined in Theorem 11.3, we have $h' \le h$ and $s_i(\alpha) = [1 \bmod n]$ for $h' \le i \le h$. Now, if in addition, $\alpha \in L'_n$, then we have $s_{h'-1}(\alpha) = [\pm 1 \bmod n]$, which implies $\alpha \in P_n$. That proves the claim.

Thus, we have shown that L'_n is contained in a subgroup P_n of $\mathbb{Z}_n^*$, and by Theorem 11.3, $P_n \subsetneq \mathbb{Z}_n^*$. By the same argument as in the proof of Theorem 11.1, it follows that $|L'_n| \le (n-1)/2$. □

The above result is not the best possible. In particular, one can show without too much difficulty that $|L'_n| \leq (n-1)/4$. We do not present this result here. Even this result is overly pessimistic from an "average case" point of view. It turns out that for "most" odd integers n of a given length, $|L'_n|$ is much smaller than this.

The Miller-Rabin algorithm is widely used in practice. Of course, in a practical implementation, before applying this test, one would first perform a bit of trial division, testing if n is divisible by any primes up to some small bound B.

11.3 The Distribution of Primes

In this section, we discuss some facts relating to the distribution of prime numbers, and algorithmic methods for generating prime numbers.

For a real number x, the function $\pi(x)$ is defined to be the number of primes up to x. Thus, $\pi(1) = 0$, $\pi(2) = 1$, $\pi(7.5) = 4$, and so on.

The main theorem in the theory of the distribution of primes is the following.

Theorem 11.5 (Prime Number Theorem) *The number of primes up to x is asymptotic to $x/\log x$:*

$$\pi(x) \sim x/\log x.$$

A proof of the Prime Number Theorem is beyond the scope of these notes.

However, one consequence of the Prime Number Theorem is that a random k-bit number (i.e., a number chosen at random from the interval $\{2^{k-1}, \ldots, 2^k - 1\}$) is prime with probability $\Theta(1/k)$.

This fact suggests the following "generate and test" algorithm for generating a random k-bit prime: choose a k-bit number at random, test it for primality, and repeat until a number is found that passes the primality test.

We leave it to the reader to verify the following assertions regarding this algorithm:

- The expected number of iterations of this algorithm is $O(k)$.
- If we use a probabilistic primality test, such as the one in the previous section, that may erroneously report that composite number is prime with probability ϵ, the probability that this prime-generating algorithm erroneously outputs a composite number is $O(\epsilon k)$ (and not, in general, $O(\epsilon)$).

If we use as our primality test the Miller-Rabin algorithm with a given error parameter t, then we have proven that $\epsilon \leq 2^{-t}$, and in fact (although we did not prove it here) $\epsilon \leq 4^{-t}$. However, as we already mentioned, these results are quite pessimistic, and in fact, the above prime-generating algorithm errs with much smaller probability, so that for $t = 1$ and sufficiently large k, the error probability is acceptably small for most practical purposes.

Primes in arithmetic progressions

For some applications, one needs a prime number of a given bit-length k, but with additional special properties. One convenient property is that $p - 1$ should be divisible by a prime q of given length ℓ. So we want an algorithm takes as input k and ℓ, with $\ell < k$, and outputs p and q such that p is a k-bit prime, q is an ℓ-bit prime, and $p \equiv 1 \pmod{q}$.

One way to generate p and q is as follows:

Step 1: Generate an ℓ-bit prime q, using an algorithm such as the "generate and test" algorithm discussed above.

Step 2: Choose m at random from the interval

$$I = \{x \in \mathbb{Z} : (2^{k-1} - 1)/q < x < (2^k - 1)/q\},$$

set $p = mq + 1$ (which is a k-bit integer), and test if p is prime; if not, repeat this step; otherwise, output p and q.

For what values of k and ℓ will this algorithm perform reasonably well?

If we view ℓ as fixed and let k tend to infinity, then Dirichlet's theorem on primes in arithmetic progressions tells us that for any ℓ-bit prime q, the probability that m chosen at random from I yields a prime is $\Theta(1/k)$.

However, suppose we want to let both k and ℓ tend to infinity. Clearly, if $k = \ell + 1$, for a given q of length ℓ, there is only one possible value for p, namely $p = 2q + 1$. So if $2q + 1$ is not prime, the above algorithm will never terminate. But suppose that k and ℓ both tend to infinity, but we restrict ℓ so that it is not too big relative to k. For example, we may require that $\ell < k/3$. In this case, it turns out that there is strong mathematical evidence (namely, the Generalized Riemann Hypothesis) that the probability that m chosen at random from I yields a prime is $\Theta(1/k)$. Thus, in this case it is reasonable to conjecture, and it is born out in practice, that Step 2 of the above algorithm terminates on average after $\Theta(k)$ iterations.

Sophie Germain primes

Sometimes, one wants a prime p of a given length to satisfy a stronger property; namely, that $p = 2q + 1$, where q is prime. Mathemeticians call the prime q in this case a "Sophie Germain" prime, while cryptographers call the prime p in this case a "strong" or "safe" prime.

It is not known whether there exist an infinite number of strong primes. However, it is conjectured, and supported by experiment, that the probability that a random k-bit number is a strong prime is $\Theta(1/k^2)$. The intuition is that a random number p of length k is prime with probability $1/k$, and for a random prime p, it does not seem unreasonable to believe that $q = (p - 1)/2$ is also prime with roughly the same probability.

If we believe this conjecture, then a reasonable way to generate strong primes is the same "generate and test" procedure we used above; namely, generate a random k-bit number p, and test if both p and $q = (p - 1)/2$ are prime; if so, output p; otherwise, repeat.

11.4 Deterministic Primality Tests

In a very recent breakthrough, Agrawal, Kayal, and Saxena have shown how to test for primality in *deterministic* polynomial time (see `http://www.cse.iitk.ac.in/primality.pdf`). Prior to this result, no such deterministic, polynomial-time test was known to exist, despite many years of extensive research in this area. It is not yet clear if this new algorithm will have much impact on practice. We do not discuss this algorithm any further here.