Radboud Universiteit

Master Thesis

Mathematics

Expanding Compositeness Tests

Author:
Erik Mulder

Supervisor:
Dr. Wieb Bosma

Second Reader:
Prof. dr. Wadim Zudilin

July 23, 2020

Contents

Chapter 1

Introduction

For millennia mathematicians have been fascinated by prime numbers, integers n such that the only positive divisors of n are 1 and n itself. Certainly not all integers are prime numbers, which raises the question how many there are and how one can find and detect them. The most obvious way to test if an integer n is prime is to check if one of $2, 3, \ldots, n-1$ divides n. If not, then n is prime! This *primality test* works every time, but doing $n-1$ divisions is not ideal when n is large. This is why we try to use algebraic properties of the prime numbers to find faster tests. We can note for example that if n is *not* prime, then n has a divisor a such that $a \leq \sqrt{n}$. Using this fact, we now only have to check if one of $2, 3, \ldots, \lfloor\sqrt{n}\rfloor$ divides n.

We are often interested in how fast an algorithm is. One way the describe the efficiency of an algorithm is by computing its *complexity*. We say that the complexity of an algorithm is $\mathcal{O}(f(n))$ if for all $n \in \mathbb{N}$, the number of bit operations the algorithm performs on an input of length n is bounded by $f(n)$, up to some constant. For example, using schoolbook long division, we can divide two integers that are less than n in about $\log(n)^2$ bit operations, up to some constant. Hence, the complexity of division is $\mathcal{O}(\log(n)^2)$ and the complexity of the first trivial primality test is $\mathcal{O}(n\log(n)^2)$. The dominant factor in the last expression is the term n, since $\log(n)$ is a lot smaller. To make life easier, we sometimes write $\widetilde{\mathcal{O}}(n)$ instead. This means that if a term $g(n)$ appears in the complexity, then the $\log(g(n))$ terms and even slower growing terms are omitted. For instance, it turns out [41] that there is a faster way to divide and multiply n bit integers, namely in

$$\mathcal{O}(\log(n)\log\log(n)\log\log\log(n)),$$

or just $\widetilde{\mathcal{O}}(\log(n))$ for short.

Although the second primality test we saw was a lot faster than the first one, its running time is still *exponential* in the length of the input, i.e. the complexity of the algorithm is not of the form $\widetilde{\mathcal{O}}(\log(n)^k)$ for some fixed k. The complexity of the *Fermat test* however is polynomial, namely $\widetilde{\mathcal{O}}(\log(n)^2)$. This test depends on Fermat's little theorem: if p is prime, then for all integers $a \neq 0 \bmod p$ we have that

$$a^{p-1} = 1 \bmod p. \tag{1.1}$$

Given an integer n, we can test the converse: take an integer $a \neq 0 \bmod n$ and check if $a^{n-1} = 1 \bmod n$ holds or not. If not, then we know for sure that is composite! However, it turns out that there can be integers a such that $a^{n-1} = 1 \bmod n$ even though n is composite. It is even worse than that, there exist so-called *Carmichael numbers* which are integers n with the property that $a^{n-1} = 1 \bmod n$ for *all* integers a with $\gcd(a, n) = 1$. This means that detecting the compositeness of such n with this test is not faster than a random search for factors of n. The Fermat test is not a primality test, it is a *compositeness test* instead, since it can only be used to prove that an integer is *not* prime.

Fortunately, there is an improvement of the Fermat test called the *Miller-Rabin test*. This test also uses (1.1), but expands on it using the following remark. Write $p-1 = 2^r s$, where s is odd. Then given an integer $a \neq 0 \bmod p$, we first compute $b = a^s \bmod p$, and then we repeatedly square b afterwards. We know that we will eventually reach $1 \bmod p$ by Fermat's little theorem, but the Miller-Rabin test is also interested in the step before that happens, i.e. the number x such that $x^2 = 1 \bmod p$. Since $\mathbb{Z}/p\mathbb{Z}$ is a field, we know that $x = \pm 1 \bmod p$ are the only two solutions for this equation. However, if an integer n has more than one distinct prime factor, then $\mathbb{Z}/n\mathbb{Z}$ is not a field and the equation $x^2 = 1 \bmod n$ will have more than two solutions. Hence, the Miller-Rabin test also checks that the second to last number in the chain of squarings is $\pm 1 \bmod n$.

Rabin famously proved [37] that a composite integer n passes the Miller-Rabin test with probability at most $1/4$. This means that if an integer n passes the test multiple times with different random integers a, then n is *probably prime*. Because then the probability that n is composite is very small. However, this is still only a compositeness test, since it does not prove the primality of n, it only gives a very strong indication that n might be prime. The Miller-Rabin test already shows that there are a lot more sophisticated tests than the first obvious one we saw in the beginning of this introduction.

Thesis overview

Personally, I have only been fascinated by prime numbers for a few decades. Fortunately, this was still enough time to write two chapters about compositeness and primality tests for this thesis.

In the first chapter we consider a new compositeness test that takes place in matrix groups modulo n, where n is the number we want to test. We will show that that there is a way to define Carmichael numbers for this test that generalizes the Carmichael numbers from the Fermat test. We study the properties of these Carmichael numbers and prove that there are infinitely many of them using an explicit construction. We will exhibit a lot of experimental data that we got from searching for these Carmichael numbers. After that, we show that a lot of classical tests such as the Miller-Rabin test and the Lucas-Lehmer test can be performed in matrix groups in a very natural way.

In the second chapter we look at compositeness and primality tests in finite fields. The first major one we will discuss is Grantham's test. In practice, Grantham's test is most often performed in a quadratic extension of a prime field. We will consider

Grantham's test in larger extensions and discuss the similarities with the celebrated AKS test [2], which can decide if an integer is prime or not in polynomial time. After that, we will show some new ways to construct polynomials that are irreducible modulo a prime divisor of n, where n is the integer you want to test for compositeness/primality. We need these polynomials to create finite fields. As an application of these constructions, we show that they can be used in a new compositeness test that uses elliptic curves. Finally, we propose an algorithm that produces a polynomial which is irreducible modulo *all* divisors of n and study its use when combined with other compositeness/primality tests.

Acknowledgements

First of all, I would like to thank my supervisor Wieb Bosma, for his advice and contagious enthusiasm. I was not always sure if my thesis was heading in the right direction, but Wieb always managed to get me back on track.

I would also like to thank Wadim Zudilin for taking the time to be the second reader of this thesis.

Finally, I would like to thank Jeroen Winkel for asking me the question we will answer in Remark 2.17.

Chapter 2

Matrices

2.1 Matrix Carmichael numbers

In this chapter we will be looking at a generalization of the Carmichael numbers mentioned in the introduction. Generalizing Carmichael numbers is by no means a new idea, however it seems that our version of the matrix approach has not been done before. The main idea is the following. Let $n, d \in \mathbb{Z}_{\geq 1}$ and take $G = \mathrm{GL}(\mathbb{Z}/n\mathbb{Z}, d) = \mathrm{GL}(n, d)$. Then G is a group under matrix multiplication. This group is finite, so it has a certain order $x(n, d)$. By Lagrange's theorem, we know that for all $A \in G$ we have that

$$A^{x(n,d)} = I, \tag{2.1}$$

where I is the identity matrix of dimension n. We will see that there is a nice formula to compute $x(n, d)$ in the case that n is prime (later on we will look at the *group exponent* instead, but the idea is the same). So there is an obvious compositeness test: pretend that n is prime and compute $x(n, d)$, take a random matrix $A \in G$ and check if (2.1) holds or not. If it doesn't hold, then we know for sure that n is *not* prime. However, just like the classical Fermat test, the converse is not true. A natural question to ask is if there are any "Carmichael numbers" for matrix groups and if so, do they have interesting properties?

Proposition 2.1. *Let p be a prime number, then*

$$\#\mathrm{GL}(p, d) = \prod_{i=1}^{d} (p^d - p^{i-1})$$

Proof. A matrix is invertible if and only if all of its columns are independent. This gives us $n^d - 1$ choices for the first column (we don't want the 0-column). For the second column we have $n^d - n$ choices to prevent a dependency. More generally, we have $n^d - n^{i-1}$ choices for the ith column. □

Given n and d, the above product is easy to compute, thus we can apply it in our compositeness test. However, we can do better than this. We can use a stronger

condition than (2.1). One of the differences between $(\mathbb{Z}/p\mathbb{Z})^*$ and $\mathrm{GL}(p,d)$ for $d>1$ is that if p is prime, then $(\mathbb{Z}/p\mathbb{Z})^*$ is cyclic, i.e. there is an element whose order equals the order of the group. $\mathrm{GL}(p,d)$ for $d>1$ however is not cyclic, which means that the order of the elements of the group are all smaller than the order of the group. This gives rise to the following definition.

Definition 2.2. *Let G be a finite group with identity element e. The group exponent of G is the smallest integer $\lambda \geq 1$ such that $x^\lambda = e$ for all $x \in G$.*

The group exponent of $\mathrm{GL}(p,d)$ has been known since the forties [36].

Theorem 2.3. *Let p be prime and $d \in \mathbb{Z}_{\geq 1}$. Then the group exponent of $\mathrm{GL}(p,d)$ equals:*

$$p^{\lceil \log_p(d) \rceil} \operatorname{lcm}(p-1, p^2-1, \ldots, p^d-1).$$

We will only give a brief sketch of the proof, since the details are quite a lot of work. Let $A \in \mathrm{GL}(p,d)$ and let $f_A(x)$ be its characteristic polynomial. Then by the Cayley-Hamilton theorem, we know that $f_A(A) = 0$ (the zero-matrix). Let $<A>$ denote the subgroup of $\mathrm{GL}(p,d)$ that is generated by A by taking matrix sums and products. This gives an isomorphism of $\mathbb{F}_p$ algebras:

$$\begin{aligned} <A> &\cong \mathbb{F}_p[x]/(f_A) \\ A &\longleftrightarrow x \bmod f_A \end{aligned}$$

Now, if $f_A(x)$ is square-free, then $f_A(x)$ factors in $\mathbb{F}_p$ as $f_1^{e_1} \cdots f_d^{e_d}$, where f_i is irreducible and has degree i, and e_i is 0 or 1. Then

$$\mathbb{F}_p[x]/(f_A) \cong \mathbb{F}_p[x]/(f_1^{e_1}) \times \cdots \times \mathbb{F}_p[x]/(f_d^{e_d}) \cong (\mathbb{F}_p)^{e_1} \times \cdots \times (\mathbb{F}_{p^d})^{e_d}.$$

Then since $(\mathbb{F}_{p^i})^*$ is cyclic and $\#(\mathbb{F}_{p^i})^* = p^i - 1$, we see where the $\operatorname{lcm}(p-1, p^2-1, \ldots, p^d-1)$ term comes from. To explain the $p^{\lceil \log_p(d) \rceil}$ term, we can look at the case that f_A contains a square factor, so $e_i > 1$ for some i. For example, $A = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ has characteristic polynomial $(x-1)^2$ and order p, since $A^i = \left(\begin{smallmatrix} 1 & i \\ 0 & 1 \end{smallmatrix}\right)$ for all i. More generally, it can be shown that

$$B = \begin{pmatrix} 1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 1 & \ldots & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & \ldots & 1 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix}$$

has characteristic polynomial $(x-1)^d$ and order $p^{\lceil \log_p(d) \rceil}$. End of sketch.

Notation. *Let $n, d \in \mathbb{Z}_{\geq 1}$ where n can be composite. Then we we write:*

$$\begin{aligned} \mu(n,d) &:= n^{\lceil \log_n(d) \rceil} \operatorname{lcm}(n-1, n^2-1, \ldots, n^d-1), \\ \delta(n,d) &:= \operatorname{lcm}(n-1, n^2-1, \ldots, n^d-1). \end{aligned}$$

We can now give the main definition of this chapter.

Definition 2.4. *Let $n, d \in \mathbb{Z}_{\geq 1}$ with n composite. We say (n, d) is Carmichael if for all $A \in GL(n, d)$ we have that*

$$A^{\mu(n,d)} = I.$$

By plugging in $d = 1$ in the above definition, we get the definition of the usual Carmichael numbers. Hence this definition can be see as a generalization. A natural question to ask is whether any Carmichael numbers exist for $d > 1$. The answer is a very positive yes. We will prove later in this chapter that for all $d \geq 1$ there exists infinitely many Carmichael numbers (n, d). The first Carmichael number for $d = 2$ is $n = 4$. We will look at the number of Carmichael numbers below a certain bound later on in this chapter.

According to Definition 2.4, to see if (n, d) is Carmichael we have to check if a certain condition holds for all invertible matrices of dimension d modulo n. This is a finite process, but since there are a lot of those matrices (see Proposition 2.1), we would like to have an equivalent condition that can be checked more easily. For the usual Carmichael numbers this can be done using the well-known Korselt criterion:

Theorem 2.5. *Let $n \in \mathbb{Z}_{\geq 2}$ be composite. Then $(n, 1)$ is Carmichael if and only if n is square-free and $p - 1 \mid n - 1$ for all prime divisors p of n.*

We will prove that there is an analogous criterion for Carmichael numbers with $d > 1$. But first we need the following lemma, which is a generalization of Proposition 2.3. This lemma can be proven using the Chinese remainder theorem.

Lemma 2.6. *Let $n, d \in \mathbb{Z}_{\geq 1}$ and let $n = \prod_{i=1}^{r} p_i^{e_i}$ be the prime factorization of n. Then the group exponent $\lambda(n, d)$ of $GL(n, d)$ is:*

$$\lambda(n, d) := \operatorname{lcm}(p_1^{e_1-1}\mu(p_1, d), \ldots, p_r^{e_r-1}\mu(p_r, d)).$$

Note that (n, d) is Carmichael if and only if $\lambda(n, d) \mid \mu(n, d)$. This gives an easier way to check that $(4, 2)$ is indeed Carmichael: $\lambda(4, 2) = 2 \cdot 2 \cdot \operatorname{lcm}(1, 3) = 12$ and $\mu(4, 2) = 4 \cdot \operatorname{lcm}(3, 15) = 60$, and $12 \mid 60$. The next lemma does not only help us with formulating a criterion, it also shows that for every $d \geq 2$ there are infinitely many integers n such that (n, d) is *not* Carmichael.

Lemma 2.7. *Let $n, d \in \mathbb{Z}_{\geq 2}$. Suppose that $n \geq d$ and that n has a prime factor p with $p < d$. Then (n, d) is not Carmichael.*

Proof. Let m be number of factors of p in n. We will count the number of factors p in $\lambda(p^m, d)$ and in $\mu(n, d)$. We know that $\mu(p, d)$ contains at least 2 factors p, because $p < d$, so $\lceil \log_p(d) \rceil \geq 2$. Hence $\lambda(p^m, d) = p^{m-1}\mu(p, d)$ contains at least $m - 1 + 2 = m + 1$ factors p. Now we look at $\mu(n, d)$. First note that the lcm term in $\mu(n, d)$ does not contain any factors p, since $n^i - 1 \equiv -1 \not\equiv 0 \bmod p$. Now we use $n \geq d \geq 2$ to see that $\mu(n, d)$ has $m \cdot \lceil \log_n(d) \rceil = m$ factors p. Hence $\lambda(p^m, d) \nmid \mu(n, d)$, so $\lambda(n, d) \nmid \mu(n, d)$. Thus (n, d) is not Carmichael. □

We can now state and prove a Korselt-like criterion for matrix groups.

Theorem 2.8. *Let $n, d \in \mathbb{Z}_{\geq 2}$ with $n \geq d$ and n composite. Then (n, d) is Carmichael if and only if for all prime divisors p of n, we have that $\delta(p, d) \mid \mu(n, d)$ and $p \geq d$.*

Proof. Write $n = \prod_{i=1}^{r} p_i^{e_i}$. We first prove that for all i:

$$\lambda(p_i^{e_i}, d) \mid \mu(n, d) \iff \lambda(p_i, d) \mid \mu(n, d) \text{ and } p_i \geq d.$$

The implication from left to right is analogous to what we did in Lemma 2.7, namely if $p_i < d$ then there would be too many factors of p_i in $\lambda(p_i^{e_i}, d)$. For the converse, since $p_i \geq d$, we know that $\lambda(p_i^{e_i}, d)$ has $e_i - 1 + 1 = e_i$ factors p_i. On the other hand, $\mu(n, d)$ has at least e_i factors p_i. So there are no problems with the factors p_i, hence $\lambda(p_i^{e_i}, d) \mid \mu(n, d)$.
The proof of the theorem is now not very hard anymore:

$$\begin{aligned}
(n, d) \text{ is Carmichael} &\iff \lambda(n, d) \mid \mu(n, d) \\
&\iff \lambda(p_i^{e_i}, d) \mid \mu(n, d) \text{ for all } i \\
&\iff \lambda(p_i, d) \mid \mu(n, d) \text{ and } p_i \geq d \text{ for all } i \\
&\iff \delta(p_i, d) \mid \mu(n, d) \text{ and } p_i \geq d \text{ for all } i
\end{aligned}$$

The last equivalence can again be seen by counting the number of factors p_i. We know that $\mu(n, d)$ has e_i of them, $\lambda(p_i, d)$ has one of them and $\delta(p_i, d)$ has none. They are equivalent since $e_i \geq 1$. □

Using this theorem, we can use a computer to find to number of Carmichael pairs in a certain range. If we had to use Definition 2.4 directly, then we wouldn't be able to get nearly as far.

d	1	2	3	4	5	6	7	8
#CM (n, d)	43	4567	226	207	208	201	201	195

Table 2.1: Number of Carmichael numbers below 10^6

Note that there is no statement about square-freeness in the above theorem, like there is in Theorem 2.5. This has to do with the definition of $\mu(n, d)$. We see that if $d = 1$, then $\lceil \log_n(1) \rceil = 0$ for all n, so there is no factor n in $\mu(n, 1)$. That is why $(n, 1)$ can't be Carmichael if $n = p^2 m$, since then $\lambda(n, 1) \mid \mu(n, 1)$ is not possible, because $p \mid \lambda(n, 1)$ and $p \nmid \mu(n, 1)$. For $d > 1$ however, we have $\lceil \log_n(d) \rceil > 0$, so we do a get factor n, so then square factors are not a problem. That factor n also might explain why there are more Carmichael numbers for $d = 2$ than for $d = 1$. The condition that $(p^2 - 1) \mid n(n^2 - 1)$ is much weaker than $p^2 - 1 \mid n^2 - 1$. Suppose that $\mu(n, 1)$ also has a factor n. The number of $n < 10^6$ such that $(p-1) \mid n(n-1)$ for all $p \mid n$ is $10694 > 4567$, which shows the power of that factor n.

Theorem 2.8 is not only useful for computer computations, it can also be used to prove interesting corollaries quite easily.

Corollary 2.9. *Let $p, k \in \mathbb{Z}_{\geq 2}$ where p is prime. Then (p^k, d) is Carmichael for all $2 \leq d \leq p$.*

Proof. By Theorem 2.8, we only have to check that $\delta(p, d) \mid \mu(p^k, d)$. Note that

$$\mu(p^k, d) = p^k \operatorname{lcm}(p^k - 1, p^{2k} - 1, \ldots, p^{dk} - 1).$$

Furthermore, for all $1 \leq i \leq d$ we have that

$$p^{ik} - 1 = (p^i - 1)(p^{i(k-1)} + p^{i(k-2)} + \cdots + p^i + 1).$$

Thus $\delta(p, d) \mid \mu(p^k, d)$. □

This corollary already shows that there are infinitely many Carmichael numbers for all $d \geq 2$, but in a kind of trivial way. This raises the next question: are there for every $d \geq 2$ infinitely many Carmichael numbers (n, d), where n is not a power of a prime?

Definition 2.10. *Let $n, d \in \mathbb{Z}_{\geq 1}$ with n composite. We say (n, d) is a strong Carmichael number if (n, d) is Carmichael and n is not a prime power.*

d	1	2	3	4	5	6	7	8
#CM (n, d)	43	4331	8	0	0	0	0	0

Table 2.2: Number of strong Carmichael numbers below 10^6

We now see that all of the Carmichael numbers for we found in Table 2.1 for $d \geq 4$ were prime powers. Furthermore, Table 2.2 suggests that a Fermat test in a matrix group of dimension ≥ 3 might be a stronger than the classical Fermat test ($d = 1$).

We will now show that for $d = 2$, there exists an infinite family of strong Carmichael numbers.

Proposition 2.11. *Let $d = 2$, then $(2^i 3^j, d)$ is a strong Carmichael number for all $i \in \mathbb{Z}_{\geq 1}$ and all $j \in \mathbb{Z}_{\geq 3}$.*

Proof. Let $n = 2^i 3^j$ with i and j as in the proposition. By theorem 2.8, we only have to check that $\delta(2, 2) \mid \mu(n, 2)$ and $\delta(3, 2) \mid \mu(n, 2)$, since $2, 3 \geq d$.
Now, $\delta(2, 2) = \operatorname{lcm}(2 - 1, 2^2 - 1) = 3$ and $\delta(3, 2) = \operatorname{lcm}(3 - 1, 3^2 - 1) = 2^3$. Hence, it suffices to show that $2^3 \cdot 3 \mid \mu(n, 2)$. Now, $\mu(n, 2) = n \cdot \operatorname{lcm}(n - 1, n^2 - 1)$. By the construction of n, we already have that $2^3 \cdot 3 \mid n$, hence $2^3 \cdot 3 \mid \mu(n, 2)$. □

We found another infinite family for $d = 2$ that can be constructed using the factorial. The proof again shows that is it relatively easy for a number to be Carmichael for $d = 2$.

Proposition 2.12. *$(m!, 2)$ is a strong Carmichael number if and only if $m \geq 4$.*

Proof. First note that $2! = 2$ is prime, so $(2, 2)$ is not Carmichael by definition. So let $m \geq 3$ and write $n = m!$. By Theorem 2.8, we need that $(p-1)(p+1) \mid n(n-1)(n+1)$ for all primes $p \leq m$. We will show that $(p-1)(p+1) \mid n$ and hence $(p-1)(p+1) \mid n(n-1)(n+1)$. First consider $p = 2$. We have $(p-1)(p+1) = 3 \mid n$, since $m \geq 3$. So from now on, let $p > 2$. Suppose that $q^k \mid p+1$, with q prime and $k \in \mathbb{Z}$ as large as possible.

We first do the case that $q > 2$. If $q^k < m+1$, then $q^k \leq m$, hence $q^k \mid m! = n$. The case that $q^k = m+1$ happens only when $m = p$ is prime and $p+1 = m+1$ is a prime power. In that case, we see that $k \geq 2$, since $p+1$ is composite (we have $p \neq 2$). Then since $q > 2$, we have that $q^{k-1}, 2q^{k-1} \leq m$, so $q^{2k-2} \mid n$, with $2k-2 \geq k$, hence $q^k \mid n$.

We conclude that apart from the factors 2 in $p+1$, we have that $p+1 \mid n$. Likewise, we can show that $p-1 \mid n$, apart from the factors 2 for now. This would not yet imply that $(p-1)(p+1) \mid n$, since there can be common factors in $p-1$ and $p+1$. We see that $\gcd(p-1, p+1) = 2$, since p is odd. Hence, the only factors that can cause problems are the factors 2, which is why we had skipped them for now.

Suppose that 2^k is the largest power of 2 that divides $p-1$ or $p+1$, then $k \geq 2$ and 2^{k+1} is the largest power of 2 that divides $(p-1)(p+1)$. Hence, we will show that $2^{k+1} \mid n$, since $(n-1)(n+1)$ is odd. Now, if $2^k \leq m$, then $2^k, 2^{k-1} \leq m$, hence $2^{2k-1} \mid n$ and $2k-1 \geq k+1$, so $2^{k+1} \mid n$. For the final case, we have that $2^k > m$, i.e. $p+1 = m+1 = 2^k$, so m is a Mersenne prime. In this case we see that if $k = 2$, then we have that $2^{k+1} \nmid n$, since then $m = 3$ and $3!$ only contains 1 factor 2. This is why $(3!, 2)$ is not Carmichael. But, if we take $k \geq 3$, then $2^{k-1}, 2^{k-2}, 3 \cdot 2^{k-2} \leq m$, hence $2^{3k-5} \mid n$, with $3k-5 \geq k+1$. So, in that case, we do have that $2^{k+1} \mid n$. We conclude that if $m \geq 4$, then $(p^2-1) \mid n \mid n(n^2-1)$. Thus $(m!, 2)$ is Carmichael if and only if $m \geq 4$. □

We can also see that $(m!, 1)$ is never Carmichael since classical Carmichael numbers are square-free, and $(6, 1)$ isn't Carmichael. Furthermore, we can also easily see that if $d > 2$, then $(m!, d)$ is never Carmichael, since $2 \mid m!$ and $2 < d$. This gives a complete classification of Carmichael numbers of the form $(m!, d)$.

We now state without proof a few other infinite families of Carmichael numbers.

Proposition 2.13.

- $(3^i 5^j, 2)$ *is a strong Carmichael number for all* $i, j \geq 1$.
- $(37^{2i} 73^{9j}, 2)$ *is a strong Carmichael number for all* $i, j \geq 1$.
- $(5^{6i} 7^{5j}, 3)$ *is a strong Carmichael number for all* $i, j \geq 1$.
- $(3^{4i} 5^{2j} 7^{2k}, 3)$ *is a strong Carmichael number for all* $i, j, k \geq 1$.

Although those examples for a few values of d are quite nice, it is of course much better if we would have some general way of constructing an infinite family for a given $d \geq 2$. We will now describe a way to do that for numbers n of the form $p^a q^b$, this method also extends to more than two distinct primes.

Note that Theorem 2.8 says that $n = p^a q^b$ is Carmichael for d if and only if:

$$p, q \geq d \text{ and } \delta(p,d), \delta(q,d) \mid \mu(n,d).$$

Note that $\delta(p,d)$ and $\delta(q,d)$ are independent of the exponents a and b. This will help us tremendously, since now we can take a and b as large as we like.

Now, let r^c be a prime power dividing $p^i - 1$ for some $1 \leq i \leq d$. Then p has multiplicative order $\leq i \leq d$ when as seen as an element of $(\mathbb{Z}/r^c\mathbb{Z})^*$. This also implies that p^a has order $\leq i$ in $(\mathbb{Z}/r^c\mathbb{Z})^*$ for all $a \in \mathbb{Z}_{\geq 1}$. Now, let x_r be the order of q in $(\mathbb{Z}/r^c\mathbb{Z})^*$. Then q^{x_r} has order 1 in that group. Thus $n = p^a q^{x_r}$ has order $\leq i$ in $(\mathbb{Z}/r^c\mathbb{Z})^*$. This is great, because now:

$$r^c \mid n^j - 1 \mid \mu(n,d)$$

for some $j \leq i \leq d$. Do this for all prime powers r^c dividing $p^i - 1$ for some $i \leq d$ and let b be the least common multiple of all x_r's. This ensures that $\delta(p,d) \mid \mu(n,d)$. To also get $\delta(q,d) \mid \mu(n,d)$, we just do the same thing: look at the prime powers dividing $q^i - 1$ and get an exponent a for p from that process. There is however a slight problem with the definition of x_r. If $r^c \mid p^i - 1$, then it is possible that $r = q$. In that case, x_r does not exist. Luckily, we can make an exception for those primes, since $\mu(n,d)$ also has a factor $n = p^a q^b$. This means that for b big enough, the prime power $r^c = q^c$ fits inside the factor n, so then we don't need that x_r exists. If the b we found is not large enough, then we can take kb instead, with $k \in \mathbb{Z}$ and $kb \geq c$.

Example 2.14. Let's look at an example to get some more insight in how the construction works. Let's take $d = 3$, $p = 3$ and $q = 5$, which are the smallest primes possible for this value of d. We write $n = p^a q^b$ and we will try to find exponents a, b such that (n,d) is Carmichael. We compute:

$$\delta(p,d) = 104 = 2^3 \cdot 13 \text{ and } \delta(q,d) = 744 = 2^3 \cdot 3 \cdot 31.$$

If we run into factors 3 or 5, then we skip them for now. Let's first look at the factors of $\delta(p,d)$. Let's start with $r^c = 2^3$. The order of $q \bmod 8$ is 2, since $5^2 = 1 \bmod 8$. Now we use that $8 \mid p^i - 1$ for some $1 \leq i \leq 3$ to see that $n = p \cdot q^2$ has order $\leq 3 \bmod 8$. Thus $8 \mid \mu(n,3)$, which is what we want. We now do the same process for the factor 13. We get that the order of $q \bmod 13$ is 4. This gives our choice for b: we take $b = \text{lcm}(4,2) = 4$.
Now we look at the factors of $\delta(q,d)$. We see that p has order $2 \bmod 8$ and order $30 \bmod 31$. So we take $a = \text{lcm}(2,30) = 30$.
This gives us $n = 3^{30} \cdot 5^4 = 128681957559155625$, which is a fairly large number already, considering the sizes of d, p and q. We now have to check if there are any annoying factors 3 or 5. We see that $3 \mid \delta(q,d)$, but that is not a problem since n already has 30 copies of 3. Hence $\delta(p,d), \delta(q,d) \mid \mu(n,d)$, thus $(n,3)$ is Carmichael.
It follows that $n = 3^{30k} \cdot 5^{4l}$ is also Carmichael for $d = 3$ for all $k, l \in \mathbb{Z}_{\geq 1}$, since if p^a has order $\leq d$ modulo some prime power, then so does $p^{k \cdot a}$. It turns out that there is also a

"smaller" family of Carmichael numbers, namely $n = 3^{10k} \cdot 5^{4l}$. This is because we could have restricted the sizes of a and b a bit more in some stages of the algorithm. More precisely, we took $a = 30$ to ensure that $n = p^a q^b$ has small order mod 31, because then $31 \mid n^i - 1$ for some $1 \leq i \leq 3$. However, if $a = 10$, then n still has small enough order mod 31. But trying to do this optimally introduces a lot of complications.

As said before, this method extends to more than two distinct primes. For example, if we want $(p^a q^b r^c, d)$ to be Carmichael, where $p, q, r \geq d$. Then we look at the prime powers $s^e \mid p^i - 1$ with $i \leq d$, and we compute the order x_s of qr mod s^e. Let b' and c' both be the lcm of those x_s's. Because then $pq^{b'} r^{c'}$ will have small order mod s^e. Then we do the same for the prime powers $s^e \mid q^i - 1$, let x_s be the order of pr mod s^e. Then let a' be the lcm of those x_s's and let c be the lcm of c' together with those x_s's. Finally, we consider the prime powers $s^e \mid r^i - 1$, this time x_s is the order of pq mod s^e. Let a be the lcm of a' and those x_s's and let b be the lcm of b' and those x_s's. Then $n = p^a q^b r^c$ is Carmichael for d, since

$$p, q, r \geq d \text{ and } \delta(p, d), \delta(q, d), \delta(r, d) \mid \mu(n, d).$$

This of course extends to more than three distinct primes. So we have proven the following theorem.

Theorem 2.15. *Let $d, k \geq 2$ and let $p_1, \ldots, p_k$ be distinct primes $\geq d$. Then there exists exponents $e_1, \ldots, e_k$ such that $n = p_1^{m_1 e_1} \cdots p_k^{m_k e_k}$ is strong Carmichael for d for all $m_1, \ldots, m_k \in \mathbb{Z}_{\geq 1}$.*

The main theorem from this chapter now easily follows.

Theorem 2.16. *For every $d \geq 1$ there exist infinitely many strong Carmichael numbers (n, d).*

Proof. The case that $d = 1$ is the hardest. Fortunately for us, that case has already been done in the nineties by Alford, Granville and Pomerance [3]. The case that $d \geq 2$ is covered by Theorem 2.15. □

Until now, we have only seen strong Carmichael numbers for $d \leq 3$, so let's look at some for larger values of d. We have tried to find "small" Carmichael numbers for those values of d, nevertheless they are still quite big. The method described above is very good at finding infinite families of strong Carmichael numbers. However, it does not always give small Carmichael numbers.

Recall that in the method for finding infinite families, we took a prime power $r^c \mid p^i - 1$ and computed an exponent b such that q^b has small order mod r^c, because then $n = p^a q^b$ has small order mod r^c, hence $r^c \mid n^j - 1 \mid \mu(n, d)$. In that process, it might have been possible to take a smaller b and still have that $n = p^a q^b$ has small order mod r^c. For Table 2.3 we first computed those exponents a, b and then we looked if $p^i q^j$ was still Carmichael for d, where $i \mid a$ and $j \mid b$. We take divisors of a and b because then n will at least still have small order modulo some primes dividing $\delta(p, d)$ and $\delta(q, d)$.

This method also has it limits however, since the number of divisors of a and b get large when p, q or d get large. An optimization of the search process when using two primes is to write $a = a_s \cdot a_b$ and $b = b_s \cdot b_b$, where a_s consists of primes that are smaller than or equal to d and a_b consists of primes that are bigger than d, likewise for b_s and b_b. This is useful, because if $s > d$ is a prime dividing the exponent b. Then by construction, there is a $k \leq d$ such that there exist a prime power $r^e \mid p^k - 1$ such that the order of q mod r^e is a multiple of s. This is because then q^b has small order mod r^e. This means that checking if $p^i q^j$ is Carmichael for d with $s \nmid j$ is unnecessary, since then q^j has order $\geq s > d$ mod r^e, and p^i has order $\leq d$ mod r^e, so $n = p^i q^j$ has order $\geq s > d$ mod r^e. But then $r^e \nmid n^l - 1$ for all $1 \leq l \leq d$, hence $r^e \nmid \mu(n, d)$, so then n can't be Carmichael for d.

This method of splitting the exponents in a small part and a bigger part doesn't work as well when we have more than 2 distinct primes. E.g. if $n = p^a q^b r^c$ is Carmichael for d and $s \mid p^k - 1$ for some $k \leq d$, then we want that $q^b r^c$ has small order mod s. Then the construction says that we should compute the order x_s of qr mod s and then take b and c to be a multiple of x_s. However, if x_s is big, then that doesn't give rise to a small Carmichael number. It might be possible that $q^2 r$ has small order $y_s < x_s$ mod s, in that case, it might be better to let b be a multiple of $2y_s$ and r a multiple of y_s. However, this seems hard to predict. This is why we don't have a very fast way to construct small Carmichael numbers with 3 divisors or more.

d	Smallest known non prime power Carmichael number n for d	#digits of n
1	$3 \cdot 11 \cdot 17$	3
2	$3 \cdot 5$	2
3	$7^2 \cdot 11^2$	4
4	$11^4 \cdot 13^6$	11
5	$5^{240} \cdot 7^{42}$	183
6	$7^{210} \cdot 19^{560}$	894
7	$7^{53130} \cdot 19^{94640}$	165922
8	$11^{28768740} \cdot 19^{24454290}$	61230567

Table 2.3: Small strong Carmichael numbers

The "small" strong Carmichael numbers we found are not necessarily the smallest ones, since other choices of primes might result in smaller numbers. Also, the Carmichael numbers in Table 2.3 are not always part of an infinite family of the form $p_1^{m_1 e_1} \cdots p_k^{m_k e_k}$.

We also looked for Carmichael numbers that aren't perfect powers, i.e. the exponents have a gcd of 1. Up to now, we haven't really looked at non-perfect power Carmichael numbers. This is because our method for finding infinite families does not always produce non-perfect powers. That is why we can't prove something like that there are infinitely many of them. We only have been able to find them for the first few values of d. One way to get Carmichael numbers that are not a perfect power might be by considering numbers n that have a lot of distinct prime factors. Because this might increase the chance that the exponents have gcd 1. For example, see $d = 4$ in Table 2.4. However, as

said before, we don't really have a good method for finding small Carmichael numbers when they have more than two distinct prime divisors.

d	Smallest known non-perfect power Carmichael number n for d	#digits of n
1	$3 \cdot 11 \cdot 17$	3
2	$3 \cdot 5$	2
3	$3^3 \cdot 5 \cdot 13^2$	5
4	$5^{16} \cdot 7^7 \cdot 11 \cdot 13$	20
5	–	-
6	–	-
7	–	-
8	–	-

Table 2.4: Small non-perfect power Carmichael numbers

For completeness sake, we also list the number of non-perfect power Carmichael numbers below a million for the first values of d, like we did in Table 2.1 and Table 2.2.

d	1	2	3	4	5	6	7	8
#CM (n, d)	43	4097	3	0	0	0	0	0

Table 2.5: Number of non-perfect power Carmichael numbers below 10^6

Remark 2.17. A question one can ask is whether there exist integers n such that (n, d) is Carmichael for all $d \geq 1$. Suppose that we have an integer n with that property. Then n is composite. Let p be a prime divisor of n, then there exists an integer d such that $p < d < n$. Theorem 2.8 now implies that (n, d) can't be Carmichael, so those integers n don't exist.
Interestingly enough, given an integer $N > 1$, we can construct integers n such that (n, d) is a strong Carmichael number for all $d = 2, 3, \ldots, N$. Let $p, q > N$ be primes. We will now show that this can be done using the same construction as for the infinite families with these p and q. Let $s^e \mid p^i - 1$, where $1 \leq i \leq N$ and let x_s be the order of q mod s^e. Then pq^{x_s} has order $\leq i$, so by doing this for all $s^e \mid p^i - 1$ and all $s^e \mid q^i - 1$ and taking the lcm of the x_s's, we get exponents a, b such that

$$\delta(p, d), \delta(q, d) \mid \mu(n, d),$$

for all $2 \leq d \leq N$, where $n = p^a q^b$. Thus n will be Carmichael for all $2 \leq d \leq N$. For example,

$$5^{12600} \cdot 7^{420}$$

has 9162 digits and is Carmichael for all $2 \leq d \leq 5$, and

$$23^{43132354366466201647258379593011296785200408241962875382773635182726333287631477641281777096440 0} \times 29^{92644631755427239961966569007322791348225327664611609756935102609964621990152033354033341826080}$$

has roughly $7.23 \cdot 10^{95}$ digits and is Carmichael for all $2 \leq d \leq 20$. This number is so big that we can probably never write down its decimal expansion, since it is estimated that there are only 10^{80} atoms in the universe [35]. However, the construction of this number took less than a second, this is because we only had to factor $\delta(p, N)$ and $\delta(q, N)$, which are independent of the exponents a and b.

2.2 Miller-Rabin in matrix groups

We will now make a few remarks about using matrix groups in compositeness tests. A possible test was already mentioned in the introduction of this chapter. For a possibly composite number n we choose $d > 1$ and compute $\mu(n, d)$. We then take a random matrix $A \in \mathrm{GL}(n, d)$ and check whether

$$A^{\mu(n,d)} = I \tag{2.2}$$

holds or not. If it doesn't, then n is composite. If it is true, then n is *probably prime*. For $d = 2$ we can see in Table 2.2 that there are quite a lot of Carmichael numbers below a million, i.e. integers n which will never be detected as composite by this test. For $d > 2$ there seem to be fewer Carmichael numbers, hence the chance of proving compositeness might increase. However, computing (2.2) becomes more expensive when d grows. We will give two algorithms to compute that matrix power. The first one is the most natural one, but not that fast, the second one is slightly more sophisticated and also faster.

Proposition 2.18. *Suppose that multiplying two matrices in $GL(n, d)$ can be done in $\widetilde{\mathcal{O}}(\log(n)d^w)$. Then computing $A^{\mu(n,d)}$ in (2.2) can be done in*

$$\widetilde{\mathcal{O}}(\log(n)^2 d^{2+w}).$$

In particular, with $w = 3$, computing $A^{\mu(n,d)}$ in (2.2) can be done in

$$\widetilde{\mathcal{O}}(\log(n)^2 d^5).$$

Proof. Using binary exponentiation, we can compute A^m in $\mathcal{O}(\log(m))$ matrix multiplications mod n. Now,

$$\mu(n, d) \;<\; n^{\lceil \log_n(d) \rceil} \prod_{i=1}^{d} n^i \;=\; n^{\lceil \log_n(d) \rceil + \sum_{i=1}^{d} i} \;=\; n^{\lceil \log_n(d) \rceil + \frac{1}{2}d(d+1)}.$$

So, $\log(\mu(n, d)) < (\frac{1}{2}d(d+1) + \lceil \log_n(d) \rceil) \log(n)$. Then because $d \geq \log_n(d)$, we have that $\log(\mu(n, d)) \in \mathcal{O}(\log(n)d^2)$. Hence we can compute $A^{\mu(n,d)}$ in $\mathcal{O}(\log(n)d^2)$ matrix multiplications mod n. □

The above result in polynomial in $\log(n)$ and d, but the exponent of d is quite large. Fortunately, there is a more efficient way to compute $A^{\mu(n,d)}$. We use ideas from [12].

Proposition 2.19. *Suppose that multiplying two matrices in $GL(n,d)$ can be done in $\widetilde{\mathcal{O}}(\log(n)d^w)$ and that the characteristic polynomial of A equals its minimal polynomial. Then $A^{\mu(n,d)}$ can be computed in*

$$\widetilde{\mathcal{O}}(\log(n)^2 d^3 + \log(n)d^{w+1})$$

and we can check if $A^{\mu(n,d)} = I$ in

$$\widetilde{\mathcal{O}}(\log(n)^2 d^3).$$

Proof. The characteristic polynomial of A can be computed in $\mathcal{O}(d^w \log(n))$ [24]. We know from the Cayley-Hamilton theorem that A is a root of its own characteristic polynomial $f_A(x)$. This means we can first compute

$$g(x) = x^{\mu(n,d)} \bmod f_A(x),$$

and then compute $g(A)$ to get $A^{\mu(n,d)}$. This is great, since multiplying polynomials is not as expensive as multiplying matrices. Note that $\deg(f_A(x)) = d$. The product of two polynomials in $\mathbb{Z}/n\mathbb{Z}[x]$ of degree d can be computed in $\widetilde{\mathcal{O}}(d\log(n))$ using fast multiplication [6, Chapter 4]. We then have to reduce that product mod f_A, which can be done in time $\widetilde{\mathcal{O}}(d\log(n))$ [6, Chapter 17]. Then we can use binary exponentiation to compute $g(x)$ mod $f_A(x)$ in

$$\widetilde{\mathcal{O}}(\log(\mu(n,d)) \cdot d\log(n)) = \widetilde{\mathcal{O}}(\log(n)^2 d^3).$$

Finally, since g has degree at most $d-1$, we can retrieve $g(A) = \sum_{i=0}^{d-1} a_i A^i$ by first computing $A^2, A^3, \ldots, A^{d-1}$ in time $\widetilde{\mathcal{O}}(d \cdot d^w \log(n)) = \widetilde{\mathcal{O}}(d^{w+1}\log(n))$, and then since the sum of two matrices mod n can be computed in $\widetilde{\mathcal{O}}(\log(n)d^2)$, we can compute $g(A) = \sum_{i=0}^{d-1} a_i A^i$ in $\widetilde{\mathcal{O}}(d \cdot \log(n)d^2) = \widetilde{\mathcal{O}}(\log(n)d^3)$. In total, we get that we can compute $A^{\mu(n,d)}$ in

$$\widetilde{\mathcal{O}}(\log(n)^2 d^3 + \log(n)d^{w+1}).$$

Now note that $A^{\mu(n,d)} = I$ if and only if $g(x) = 1 \bmod f_A$, since f_A is the minimal polynomial of A. Hence, if we only want to see if $A^{\mu(n,d)} = I$ or not, then we don't have to compute $A^2, A^3, \ldots A^{d-1}$. By skipping that part, the total running time is

$$\widetilde{\mathcal{O}}(\log(n)^2 d^3). \qquad \square$$

We see that the method in Proposition 2.19 is a factor $\widetilde{\mathcal{O}}(d^{w-1})$ faster than the method in Proposition 2.18 in checking if $A^{\mu(n,d)} = I$. This is quite substantial, because there are d^2 elements in a matrix of dimension d, so $w \geq 2$. Currently, the best value of w is 2.3728639 [17].

Remark 2.20. Suppose that we do the compositeness test described above Proposition 2.18 with $d = 2$ and n a prime number and suppose that we have a random matrix

$A \in \mathrm{GL}(n,2)$. If A is diagonalizable, then we can write $A = PDP^{-1}$ for some diagonal matrix $D = \mathrm{diag}(a,b) \in \mathrm{GL}(n,2)$. But then

$$A^{n-1} = (PDP^{-1})^{n-1} = PD^{n-1}P^{-1} = PIP^{-1} = I.$$

Where we used that $a^{n-1} \equiv b^{n-1} \equiv 1 \bmod n$, since n is prime. This means that raising the matrix to the power $\mu(n,2)$ would be overkill in this situation.
Furthermore, if n is not prime and A is diagonalizable modulo all the divisors of n, then for $A^{\mu(n,2)} = I$ to happen, we only need that $\mu(n,2)$ is a multiple of $p-1$ for all $p \mid n$. But then it's better to do a usual Fermat test modulo n, because $\mu(n,2)$ has a lot more factors than the exponent $n-1$ occurring in the classic Fermat test. So then it's less effective in showing the compositeness of n than the classical Fermat test.
This shows that most of the time it's better to have a matrix that is not diagonalizable modulo at least one of its divisors. For the $d = 2$ case, there is a nice way to do this using the Legendre symbol. We know from linear algebra that if p is prime, then a matrix $A \in \mathrm{GL}(p,d)$ is diagonalizable if the characteristic polynomial splits in d distinct factors. Let δ be the discriminant of the characteristic polynomial of $A \in \mathrm{GL}(p,2)$. Then A is *not* diagonalizable if $\left(\frac{\delta}{p}\right) = -1$. If we switch back to the case that n is composite, then we see that $\left(\frac{\delta}{n}\right) = -1$ implies that $\left(\frac{\delta}{p}\right) = -1$ for some $p \mid n$. Hence the matrix won't be diagonalizable modulo at least one divisor of n. Given a polynomial f, we can construct a matrix with characteristic polynomial f by computing its *companion matrix* [8].
For general d it is still nice to use matrices whose characteristic polynomial does not split in factors of small degree, for the same reason as above. In the next chapter we will look at a lot of interesting ways to construct polynomials of large degree that are irreducible modulo at least one divisor of n.

Until now in this chapter, the only compositeness test we have looked at is a Fermat test for matrices. We will now show that another classical test in $\mathbb{Z}/n\mathbb{Z}$ can also be generalized to a test in $\mathrm{GL}(n,d)$, namely the Miller-Rabin test. As a test in $\mathbb{Z}/n\mathbb{Z}$, Miller-Rabin searches for "strange" square roots of $1 \bmod n$. This is because if n is prime, then the only two solutions to $x^2 \equiv 1 \bmod n$ are $x = 1$ and $x = -1$. This is because in that case $\mathbb{Z}/n\mathbb{Z}$ is a field. If n has more than one distinct prime divisor however, then there are more than two roots of $1 \bmod n$.

If we want to generalize this to matrix groups, then we have to know what the roots of the identity matrix $I \bmod n$ are when n is prime. We of course have the usual suspects I and $-I$, but what about $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 2 \end{smallmatrix}\right) \bmod 3$? It turns out that there are a lot of square roots of I, especially if n or d is large.

Proposition 2.21. *Let p be an odd prime, then for all $A \in GL(p,d)$ we have that*

$$A^2 = I \iff A \text{ is similar to } \mathrm{diag}(I_t, -I_{d-t}) \text{ for some } 1 \le t \le d.$$

Two matrices A and B are *similar* if there exists an invertible matrix C such that $A = CBC^{-1}$. The implication from right to left is not hard. Since then we can write

$$A = P \cdot \mathrm{diag}(I_t, -I_{d-t}) \cdot P^{-1}$$

for some $P \in \mathrm{GL}(p, d)$ and some $1 \leq t \leq d$. It follows that $A^2 = PIP^{-1} = I$. The converse however is a bit harder, we refer to [23].

Lemma 2.22. *Similar matrices A and B have the same characteristic polynomial $p_A(x)$ and $p_B(x)$.*

Proof. Suppose that matrices A and B are similar, say $B = PAP^{-1}$. Recall that the characteristic polynomial of A can be defined as $\det(xI - A)$. Now,

$$\begin{aligned} p_B(x) &= \det(xI - B) = \det(xI - PAP^{-1}) = \det(PxIP^{-1} - PAP^{-1}) \\ &= \det(P(xI - A)P^{-1}) = \det(P)\det(xI - A)\det(P^{-1}) = \det(xI - A) = p_A(x). \square \end{aligned}$$

Lemma 2.23. *$\mu(n, d)$ can be computed in $\widetilde{\mathcal{O}}(\log(n)d^2)$.*

Proof. We have to compute a lcm of a list, this can be done using the following rule:

$$\mathrm{lcm}(a, b, c) = \mathrm{lcm}(\mathrm{lcm}(a, b), c).$$

The computation of the lcm of two $\mathcal{O}(m)$ bit numbers can be done in $\widetilde{\mathcal{O}}(m)$ using a faster version of the Euclidean algorithm [40] and the fact that $\mathrm{lcm}(a, b) = \frac{ab}{\gcd(a,b)}$. Now, the list has length d and the bit size of the largest occurring number in the sequence of lcm's is $\mathcal{O}(\log(n)d^2)$, since $\log(\mu(n, d)) \in \mathcal{O}(\log(n)d^2)$. Hence we can compute $\mu(n, d)$ in $\widetilde{\mathcal{O}}(\log(n)d^2)$. $\square$

Using all those ingredients, we can now state a Miller-Rabin like test for matrices. In the algorithm, n is the number we want to test for compositeness, d is the dimension of the matrices used and k is the number of tries.

Algorithm 1 Miller-Rabin with matrices

1: **function** MR(n, d, k)
2: **if** n is a perfect power **then**
3: **return** n is composite
4: **end if**
5: $x = \mu(n, d)$
6: write $x = 2^s y$ where y is odd
7: precompute all polynomials of the form $(x-1)^t(x+1)^{d-t}$ with $0 \leq t \leq d$
8: $i = 0$
9: **while** $i < k$ **do**
10: **pick** random d by d matrix A
11: $f = \text{CharacteristicPolynomial}(A)$
12: $g = \gcd(f, f')$
13: **if** g does not exist **then**
14: **return** n is composite
15: **end if**
16: **if** $f(0) = 0$ **or** $g \neq 1$ **then**
17: **continue** while loop
18: **end if**
19: $i = i + 1$
20: $B = A^y$
21: **if** $B = I$ **then**
22: **continue** while loop
23: **end if**
24: $count = 0$
25: **while** $B^2 \neq I$ and $count < s$ **do**
26: $B = B^2$
27: $count = count + 1$
28: **end while**
29: $g = \text{CharacteristicPolynomial}(B)$
30: **if** g is not of the form $(x-1)^t(x+1)^{d-t}$ **then**
31: **return** n is composite
32: **end if**
33: **if** B can't be diagonalized to $\text{diag}(I_t, -I_{d-t})$ **then**
34: **return** n is composite
35: **end if**
36: **end while**
37: **return** n is probable prime
38: **end function**

Proposition 2.24. *Algorithm* 1 *is correct and runs in* $\widetilde{\mathcal{O}}(kd^3 \log(n)^2)$.

Proof. We first prove the correctness of the algorithm. If g in line 12 is equal to 1, then we know that f is square-free. This implies that the characteristic polynomial and

the minimal polynomial of A coincide, since there are no duplicate eigenvalues, see [15, Theorem 4.7]. The gcd is computed using the Euclidean algorithm, if n is not prime, then it is possible that one of the leading coefficients of a polynomial in that process is not invertible, showing that n is not prime. Note that $f(0) = \pm\det(A)$, so $f(0) \neq 0$ makes sure that A is an element of $\mathrm{GL}(n, d)$ in the case that n is prime. If n is not prime then it is possible that $\det(A) \neq 0$ but $A \notin \mathrm{GL}(n, d)$. This is certainly not a problem, since then $A^{\mu(n,d)} \neq I$, so we would prove that n is indeed composite. By Theorem 2.3, we know that B^2 in line 25 is eventually I if n is prime. In line 30 we assert that g is of the form $(x-1)^t(x+1)^{d-t}$, because that is the characteristic polynomial of $\mathrm{diag}(I_t, -I_{d-t})$ and Proposition 2.21 and Lemma 2.22 tell us that g has to be of that form. Now note that B is diagonalizable if and only if B is similar to $\mathrm{diag}(I_t, -I_{d-t})$, so in line 33 we explicitly check if B is similar to $\mathrm{diag}(I_t, -I_{d-t})$ or not.

Now we look at the running time. We begin by checking if n is a perfect power. This can be done by checking if one of $n^{1/2}, \ldots, n^{1/\lfloor \log(n) \rfloor}$ is an integer. A very efficient version of this algorithm by Bernstein [7] runs in $\mathcal{O}(\log(n)^2)$. In line 5 we compute $\mu(n, d)$, which can be done in $\widetilde{\mathcal{O}}(\log(n)d^2)$ by Lemma 2.23. In line 7 we compute a list of $d+1$ polynomials. Every polynomial is computed by a product of d polynomials of degree $\leq d$, so that can be done in $\widetilde{\mathcal{O}}(d^3 \log(n))$. In lines 11 and 29 we can compute the characteristic polynomial in $\widetilde{\mathcal{O}}(d^w \log(n))$ according to [24]. In line 12 we compute the gcd in $\widetilde{\mathcal{O}}(d^2 \log(n))$. In line 20 and in the while loop that starts in line 25, we compute in total $A^{\mu(n,d)}$, we know from Proposition 2.19 that this can be done in $\widetilde{\mathcal{O}}(\log(n)^2 d^3)$. If we reach line 33, then we know that the eigenvalues of B are 1 or -1. So we can try to diagonalize B by computing $\ker(B - I)$ and $\ker(B + I)$ using Gaussian elimination in $\widetilde{\mathcal{O}}(d^3 \log(n))$ [4]. Now note that B is diagonalizable if and only if we get d independent eigenvectors. We see that the running time of the while loop of line 9 is dominated by the computation time of $A^{\mu(n,d)}$. Since we repeat the process at most k times, the total running time is $\widetilde{\mathcal{O}}(kd^3 \log(n)^2)$. □

Example 2.25. We will use Algorithm 1 to see if $n = 377$ is prime or not, using $d = 2$. We compute $\mu(n, d) = 53582256 = 2^4 \cdot 3348891$. The first random matrix we pick is $A = \left(\begin{smallmatrix} 20 & 272 \\ 127 & 5 \end{smallmatrix}\right)$, with $\det(A) = 240$. Note that $\gcd(240, n) = 1$, so if n is not prime, then at least $A \in \mathrm{GL}(p, d)$ for all $p \mid n$. We continue by computing $B = A^{3348891} = \left(\begin{smallmatrix} 153 & 226 \\ 197 & 293 \end{smallmatrix}\right)$. We now repeatedly square this matrix and look if we get a normal square root of I. We have that $B^2 = \left(\begin{smallmatrix} 71 & 137 \\ 21 & 306 \end{smallmatrix}\right)$ and $B^4 = I$. We compute that the characteristic polynomial of B^2 is $x^2 - 1 = (x-1)(x+1)$, hence it is a "normal" square root of I. At this stage we might think that n is prime. However, the next random matrix we pick is $A = \left(\begin{smallmatrix} 360 & 239 \\ 122 & 39 \end{smallmatrix}\right)$ with $\det(A) = 339$ and $\gcd(339, n) = 1$. Then

$$B = A^{3348891} = \begin{pmatrix} 204 & 348 \\ 348 & 30 \end{pmatrix}, \quad B^2 = \begin{pmatrix} 233 & 0 \\ 0 & 233 \end{pmatrix}, \quad B^4 = I.$$

We compute that the characteristic polynomial of B^2 is $x^2 + 288x + 1$, which is not of the form $(x-1)^i(x+1)^{2-i}$ for some $0 \leq i \leq 2$. Hence we found an unexpected root of I, so n is not prime. Furthermore, we can find a factor of n is this situation, since we

know that the characteristic polynomial of B^2 is of the form $(x-1)^i(x+1)^{2-i}$ modulo the prime divisors of n, where i can vary for the different factors of n. If we look at $x^2+288x+1-(x-1)^2 = 290x$ then we can compute $\gcd(290, n) = 29$. So we can factor $n = 13 \cdot 29$. However, this is not a very good factorization algorithm in general, since if n does not have small prime factors, then most of the time we have that $A^{\mu(n,d)} \neq I$, so we don't get into that situation.

Although we have shown that Algorithm 1 is correct, it would of course be a lot better if we could also say something about the probability that this test detects a composite number n. It is natural to look at the proof of the classical Miller Rabin test and look if we can do something similar here. Those proofs (like [14]) often look at sets like $H = \{a \in \mathbb{Z}/n\mathbb{Z} \mid a^{n-1} = 1\}$, it is not hard to see that this is a subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$, since if $a^{n-1} = b^{n-1} = 1 \bmod n$, then also $(ab)^{n-1} = 1 \bmod n$. This is very useful, because if n is not a Carmichael number, then H is a subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ of index at least 2, which gives a probability of at least one half that $a^{n-1} \neq 1 \bmod n$. For the case that n is a Carmichael number, another subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ is considered. The problem for us is that $\mathrm{GL}(n, d)$ is not commutative for $d > 1$, so we can't construct such subgroups.

Our test consists of two major steps. Checking if $A^{\mu(n,d)} = I$ and checking if we don't encounter any unexpected square roots of I. Using the second property, we have been able to prove that a composite number n is detected with probability at least one half using Algorithm 1. This is not bad, because it means that if we run the test k times, then we either proved that n is composite or we know that the probability that n is composite is less than 2^{-k}, which gets small very quickly. However, it still not very useful in practice, since the classical Miller-Rabin test for example is more efficient.

Lemma 2.26. *Let G be a cyclic group with order $2^s y$, where y is odd. If g is a random element of G, then the probability that $2^s \mid \mathrm{order}(g)$ is $1/2$.*

Proof. Since G is cyclic, we know that if $m \mid \#G$, then there are $\phi(m)$ elements of order m in G, where ϕ is the Euler phi function. This means that the number of elements in G whose order is divisible by 2^s is

$$\sum_{a|y} \phi(2^s a) = \phi(2^s) \sum_{a|y} \phi(a) = 2^{s-1} y = \frac{1}{2} \#G. \qquad \square$$

Proposition 2.27. *Let $n, d \in \mathbb{Z}_{\geq 1}$ and suppose that n is composite. Then (n, d) passes one run of Algorithm 1 with probability at most $1/2$.*

Proof. If $A^{\mu(n,d)} \neq I$, then it shows that n is composite, so suppose $A^{\mu(n,d)} = I$. We know that n is not a perfect power, so n has at least two distinct prime divisors p and q. We follow the notation from Algorithm 1, so let A be the random matrix we pick. Now, $\gcd(f, f') = 1$ implies that f is square-free modulo all divisors of n, where f is the characteristic polynomial of A. This means that A is diagonalizable in large enough extensions of $\mathbb{F}_p$ and $\mathbb{F}_q$. Say $A = PDP^{-1}$ in $\mathbb{F}_{p^r}$ and $A = QEQ^{-1}$ in $\mathbb{F}_{q^s}$, where D

and E are diagonal matrices.
In the algorithm we compute $B = A^y$, where y is the odd part of $\mu(n,d)$. Then since $A^{\mu(n,d)} = I$, we know that the order of B is a power of 2 modulo all divisors of n. Hence the order of $D^y \bmod p$ and $E^y \bmod q$ are both a power of 2. Now, suppose that f factors into irreducibles as $f_1 \cdots f_r$ in $\mathbb{F}_p$ and as $g_1 \cdots g_s$ in $\mathbb{F}_q$. Let $F_i = \mathbb{F}_p[x]/(f_i)$ and $G_i = \mathbb{F}_q[x]/(g_i)$ for all i. If α_{ij} for $j = 1, \ldots, \deg(f_i)$ are the roots of f_i in F_i and β_{ij} the roots of g_i in G_i for all i. Then the elements on the diagonal of D are precisely α_{ij}^y with $1 \leq i \leq r$ and $1 \leq j \leq \deg(f_i)$, likewise for the matrix E. We see that if p_{max} is the largest order of an element in $D^y \bmod p$ and if q_{max} is the largest order of an element in $E^y \bmod q$. Then we get an expected root of I if and only if $p_{max} = q_{max}$ and the number of α_{ij}^y that have maximal order p_{max} equals the number of β_{ij}^y that have order q_{max}. We will show this happens with probability at most $1/2$.
Let a be an index such that $\#F_a^*$ has the most factors of 2 of all F_i, likewise an index b for the G_i. There might be multiple copies of F_a or G_b, since f can have multiple irreducible factors of the same degree. So say that there are v copies of F_a and w copies of G_b. We will now use that all of the groups F_i^* and G_i^* are cyclic. Note that if i is fixed, then the multiplicative order of all the α_{ij} is the same, likewise for the β_{ij}.
First suppose that $p_{max} \neq q_{max}$, say $p_{max} > q_{max}$. Then by Lemma 2.26, we know that the α_{aj}^y for $j = 1, \ldots, \deg(f_a)$ have order $2^{p_{max}}$ with probability $1/2$. There can't be any elements β_{ij}^y with such large order, hence we get an unexpected root of I in that case.
Now suppose that $p_{max} = q_{max}$. By Lemma 2.26, we expect $\frac{v \deg(f_a)}{2}$ elements α_{ij}^y of order p_{max} and $\frac{w \deg(g_b)}{2}$ elements β_{ij}^y of order q_{max}. The probability that those fractions are equal is maximal when $v = w$ and $\deg(f_a) = \deg(g_b)$. The probability that $k \deg(f_u)$ elements α_{ij}^y have order p_{max} is $x = \binom{v}{k}(1/2)^v$, since it is binomially distributed with probability $1/2$. Hence the the probability that the number of α_{ij}^y that have maximal order p_{max} equals the number of β_{ij}^y that have order p_{max} is:

$$\sum_{k=0}^{v} \left(\binom{v}{k} \left(\frac{1}{2}\right)^v \right)^2 = \left(\frac{1}{4}\right)^v \sum_{k=0}^{v} \binom{v}{k}^2 = \left(\frac{1}{4}\right)^v \binom{2v}{v} \leq \left(\frac{1}{4}\right)^v 2^{2v-1} = \frac{1}{2}. \qquad \square$$

It would be nice if we could prove that the probability that Algorithm 1 detects composite numbers would increase if we increase d. Experimental testing does seem to support this idea. For all of the composite non-perfect power numbers n below 10^4 and for all $1 \leq d \leq 5$, we computed an approximation of the probability that Algorithm 1 detects that n is composite in one run. We did this by running the algorithm on 10^4 random d by d matrices for all of those n and for all $1 \leq d \leq 5$.

	$n \leq 10^2$		$n \leq 10^3$		$n \leq 10^4$	
d	average	worst	average	worst	average	worst
1	0.957	0.792	0.989	0.772	0.997	0.764
2	0.983	0.885	0.994	0.883	0.998	0.831
3	0.995	0.947	0.998	0.939	0.9994	0.921
4	0.995	0.969	0.998	0.966	0.9996	0.960
5	0.998	0.984	0.9995	0.981	0.9999	0.981

Table 2.6: Approximation of the probability that Algorithm 1 detects composite numbers in one run in certain intervals.

Remark 2.28. You might be wondering if a Miller-Rabin like test can be used in some other groups that can be defined modulo n. Here is a rough list of things you need to be able to do in the case that n is prime:

- Efficiently compute the order of the group.
- Efficiently pick a random element of your group.
- Know what the elements of order 2 are.

An example would be elliptic curves. Since if n is prime, then we can compute the order of an elliptic curve over $\mathbb{F}_n$ in polynomial time [42], we can pick random elements by computing square roots mod n, and the elements $P = (x, y)$ of order 2 are of the form $(x, 0)$. If n is not prime, then you might fail to accomplish one of those requirements, but that would already show that n is not prime. One drawback of this method is that the computation of the order of the elliptic curve is not very fast. There are some exceptions however, for example when you look at *supersingular curves*, see [20]. Another interesting instance of a Miller-Rabin like test can be done in quadratic unique factorization domains [50].

2.3 Lucas-Lehmer in matrix groups

Another famous primality test is the Lucas-Lehmer test for Mersenne numbers. It states that if p is an odd prime, then $q = 2^p - 1$ is prime if and only if $s_{p-2} = 0 \bmod q$, where

$$s_i = \begin{cases} 4 & \text{if } i = 0; \\ s_{i-1}^2 - 2 & \text{otherwise.} \end{cases} \tag{2.3}$$

This gives a very efficient test to check if a Mersenne number is prime or not. Currently [19], the largest known prime number is a Mersenne prime, namely

$$2^{82589933} - 1,$$

which has more than 24 million decimal digits. Since this chapter is about matrices, we will give a matrix approach to this test. We follow in essence the same proof as in [10]

and [38], but we have translated it to the world of matrices. In Theorem 2.31 we will see that there is a very nice way to go from the matrix world back to the setting of (2.3).

Proposition 2.29. *Let p be an odd prime and let $q = 2^p - 1$. Let A be a matrix in* $\mathrm{GL}(q, 2)$ *with characteristic polynomial*

$$p_A(x) = x^2 - 4x + 1.$$

Then q is prime if and only if

$$A^{2^{p-1}} = -I.$$

Proof. First suppose that $A^{2^{p-1}} = -I$. This implies that A has order 2^p modulo all divisors of q. If q is not prime, then q has a prime divisor r with $r \leq \sqrt{q}$. We know that, as an element of $\mathrm{GL}(r, 2)$, the order of A divides $r(r^2 - 1)$. Now, q is odd, hence so is r. This means that the factors 2 have to lie in the r^2-1 part. Now, $r^2-1 \leq q-1 < q+1 = 2^p$, but A has order 2^p mod r. This gives a contradiction, thus q is prime.
Now suppose that q is prime. Note that the discriminant of $p_A(x)$ is $16-4 = 12 = 2^2 \cdot 3$. Now,

$$\left(\frac{12}{q}\right) = \left(\frac{3}{q}\right) = -\left(\frac{q}{3}\right),$$

where in the last step we used quadratic reciprocity and the fact that $q = 3 \bmod 4$. Now, since p is odd, we know that $2^p = 2 \bmod 3$. Hence

$$\left(\frac{q}{3}\right) = \left(\frac{2^p - 1}{3}\right) = \left(\frac{1}{3}\right) = 1.$$

Thus $\left(\frac{12}{q}\right) = -1$, which means that $p_A(x)$ is irreducible over $\mathbb{F}_q$. We will now use the correspondence between A and $x \bmod p_A(x)$ as described below Theorem 2.3. Note that $p_A(x)$ factors in $\mathbb{F}_{q^2}$ as $(x - (2 + \sqrt{3}))(x - (2 - \sqrt{3}))$. Let $\omega = 2 + \sqrt{3}$ and $\overline{w} = 2 - \sqrt{3}$ in $\mathbb{F}_{q^2}$. If we can show that $w^{2^{p-1}} = -1$ and $\overline{w}^{2^{p-1}} = -1$, then $x^{2^{p-1}} = -1 \bmod p_A(x)$ and hence $A^{2^{p-1}} = -I$. Note that

$$\frac{(3 + \sqrt{3})^2}{6} = \frac{9 + 6\sqrt{3} + 3}{6} = \omega.$$

Hence

$$(6\omega)^{2^{p-1}} = (3 + \sqrt{3})^{2^p} = (3 + \sqrt{3})^q (3 + \sqrt{3}) = (3^q + \sqrt{3}^q)(3 + \sqrt{3}),$$

by the freshman's dream. Now note that since $\left(\frac{3}{q}\right) = -1$, we know that $3^{\frac{q-1}{2}} = -1$, hence

$$\sqrt{3}^q = \sqrt{3} \cdot 3^{\frac{q-1}{2}} = -\sqrt{3}.$$

We conclude that

$$(6\omega)^{2^{p-1}} = (3 - \sqrt{3})(3 + \sqrt{3}) = 9 - 3 = 6.$$

Now, since $q \equiv 7 \bmod 8$, we know that $\left(\frac{2}{q}\right) = 1$, so $\left(\frac{6}{q}\right) = -1$. Thus

$$\omega^{2^{p-1}} = \frac{6}{6^{2^{p-1}}} = \frac{1}{6^{2^{p-1}-1}} = \frac{1}{6^{\frac{q-1}{2}}} = \frac{1}{\left(\frac{6}{q}\right)} = -1.$$

We can do a similar calculation for $\overline{\omega}$, or we can note that

$$\overline{\omega}^{2^{p-1}} = \omega^{q \cdot 2^{p-1}} = (-1)^q = -1. \qquad \square$$

Proposition 2.29 already gives us a way to detect Mersenne primes using matrices, take $A = \left(\begin{smallmatrix} 2 & 1 \\ 3 & 2 \end{smallmatrix}\right)$ for example. Then we can check if q is prime in $\mathcal{O}(p)$ matrix multiplications mod q. This algorithm has the same asymptotic complexity as the classical Lucas-Lehmer test, since there you have to do $\mathcal{O}(p)$ integer multiplications mod q. However, this matrix approach is a constant factor slower, because multiplying 2 by 2 matrices consists of multiple integer multiplications instead of just 1.

To complete the discussion on the Lucas-Lehmer test, we will prove the correctness of the test in the famous form as in (2.3). We do this by using the trace of a matrix. Recall that the characteristic polynomial of a 2 by 2 matrix A is $x^2 - \operatorname{tr}(A)x + \det(A)$.

Lemma 2.30. *Let p, q and A as in Proposition 2.29. Then*

$$A^{2^{p-1}} = -I \text{ if and only if } \operatorname{tr}(A^{2^{p-2}}) = 0.$$

Proof. Write $B = A^{2^{p-2}}$. If $A^{2^{p-1}} = -I$ then $B^2 + I = 0$. So the characteristic polynomial of B is $x^2 + 1$, hence $\operatorname{tr}(B) = 0$. Now suppose that $\operatorname{tr}(A^{2^{p-2}}) = 0$, then the characteristic polynomial of B is $x^2 + \det(B)$. We see that

$$\det(B) = \det(A)^{2^{p-2}} = 1^{2^{p-2}} = 1.$$

Hence $B^2 + I = 0$, so $B^2 = -I$. $\square$

We can now prove the correctness of the Lucas-Lehmer test via our matrix approach.

Theorem 2.31. *Let p be an odd prime and let $q = 2^p - 1$. Then q is prime if and only if $s_{p-2} = 0 \bmod q$.*

Proof. Let A be a matrix in $\mathrm{GL}(q, 2)$ with characteristic polynomial $p_A(x) = x^2 - 4x + 1$. We will first prove by induction that $\operatorname{tr}(A^{2^i}) = s_i$ for all $i \geq 0$. For $i = 0$ we have that

$$\operatorname{tr}(A^{2^0}) = \operatorname{tr}(A) = 4 = s_0.$$

Now let $m \geq 1$ and set $B = A^{2^{m-1}}$. Our induction hypothesis is that $\operatorname{tr}(B) = s_{m-1}$. We see that B again has determinant one, so $B^2 - \operatorname{tr}(B) \cdot B + I = 0$. Then since the trace function is linear, we get that

$$\operatorname{tr}(B^2) = \operatorname{tr}(B) \cdot \operatorname{tr}(B) - \operatorname{tr}(I).$$

Thus

$$\operatorname{tr}(A^{2^m}) = \operatorname{tr}(B)^2 - 2 = s_{m-1}^2 - 2 = s_m.$$

Now we use Proposition 2.29 and Lemma 2.30 to conclude that:

$$q \text{ is prime } \iff A^{2^{p-1}} = -I \iff \operatorname{tr}(A^{2^{p-2}}) = 0 \iff s_{p-2} \equiv 0 \bmod q. \qquad \square$$

Chapter 3

Finite Field Extensions

A lot of compositeness tests take in place in the ring $\mathbb{Z}/n\mathbb{Z}$, where n is the number we want to test. If n is prime then this ring is the finite field of n elements $\mathbb{F}_n$. Not all finite fields have prime order, we can also create finite fields which are extensions of a finite field of prime order. This is done by adjoining a root of an irreducible polynomial in $\mathbb{F}_n[x]$ to $\mathbb{F}_n$. The orders of those extension fields are powers of the order of the prime field. More precisely, if $f \in \mathbb{F}_n[x]$ is irreducible of degree d and has a root α, then

$$\mathbb{F}_n(\alpha) \cong \mathbb{F}_n[x]/(f) \cong \mathbb{F}_{n^d}$$

is a finite field of order n^d. We can do computations in those finite fields using the polynomial representation $\mathbb{F}_n[x]/(f)$. E.g., given $\beta, \gamma \in \mathbb{F}_{n^d}$, we can get polynomial representations $h, g \in \mathbb{F}_n[x]/(f)$, respectively. Then say, computing $\beta \cdot \gamma$ can be done using polynomial multiplication: $h(x)g(x) \bmod (f, n)$. If it is clear that we are working modulo n, then we will just write $h(x)g(x) \bmod f$ instead.

3.1 Grantham's test

A natural question to ask is if there are any interesting primality or compositeness tests that take place in extensions of finite fields. We have just seen that if n is prime then we can create those fields, as long as we can find irreducible polynomials. However, if n is not prime, then $\mathbb{Z}/n\mathbb{Z}$ isn't even a field, let alone extensions of $\mathbb{Z}/n\mathbb{Z}$. This gives the following general setting for a compositeness test that uses finite fields. If n is prime, then we can exploit properties of those finite fields, and if n is not prime, then we can hope that those properties don't hold, showing that n is indeed composite. We will now state some of those properties that we will use throughout this chapter. The first one is a well-known theorem from finite field theory [25, Theorem VIII.2.1].

Proposition 3.1. *Let p be a prime and let $d \in \mathbb{Z}_{\geq 1}$. Then $x^{p^d} - x$ is equal to the product of all monic irreducible polynomials in $\mathbb{F}_p[x]$ whose degree divides d.*

Another property of finite fields comes from Galois theory and is strongly related to the one stated above [25, Theorem VIII.2.4].

Proposition 3.2. *Let p be a prime and let $f \in \mathbb{F}_p[x]$ be irreducible and of degree d. If α is a root of f in some extension of $\mathbb{F}_p$, then the remaining $d-1$ roots of f are*

$$\alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{d-1}}.$$

Equivalently, the automorphism group of $\mathbb{F}_{p^d}$ over $\mathbb{F}_p$ is generated by the the *Frobenius homomorphism* σ that sends β to β^p for all $\beta \in \mathbb{F}_{p^d}$.

The next property is less well-known, so this time we will provide a proof. We follow the proof of [49, Theorem 1], but we only do the case that f is irreducible. Let $\Delta(f)$ denote the discriminant of f. Recall that if $\alpha_1, \ldots, \alpha_d$ are the roots of f in a splitting field of f, then

$$\Delta(f) = \prod_{1 \le i < j \le d} (\alpha_i - \alpha_j)^2.$$

Proposition 3.3. *Let p be prime and let f be irreducible over $\mathbb{F}_p$ of degree d. Then $\left(\frac{\Delta(f)}{p}\right) = 1$ if and only if d is odd.*

Proof. Let α be a root of f in some extension of $\mathbb{F}_p$. Then by Proposition 3.2, the roots of f are $\sigma^i(\alpha)$ for $0 \le i \le d-1$. Then by the definition of the discriminant, we have that $\Delta(f) = \delta(f)^2$, where

$$\delta(f) = \prod_{0 \le i < j \le d-1} (\sigma^i(\alpha) - \sigma^j(\alpha)).$$

We will now use the fact that for all a in some extension of $\mathbb{F}_p$, we have that $\sigma(a) = a$ if and only if $a \in \mathbb{F}_p$. We compute

$$\begin{aligned}
\sigma(\delta(f)) &= \prod_{0 \le i < j \le d-1} (\sigma^{i+1}(\alpha) - \sigma^{j+1}(\alpha)) = \prod_{1 \le i < j \le d-1} (\sigma^i(\alpha) - \sigma^j(\alpha)) \cdot \prod_{i=1}^{d-1} (\sigma^i(\alpha) - \alpha) \\
&= (-1)^{d-1} \prod_{1 \le i < j \le d-1} (\sigma^i(\alpha) - \sigma^j(\alpha)) \cdot \prod_{i=1}^{d-1} (\sigma^0(\alpha) - \sigma^i(\alpha)) \\
&= (-1)^{d-1} \prod_{0 \le i < j \le d-1} (\sigma^i(\alpha) - \sigma^j(\alpha)) = (-1)^{d-1} \delta(f).
\end{aligned}$$

Hence, we see that $\delta(f) = \sigma(\delta(f))$ if and only if d is odd. Thus, $\Delta(f)$ is a square mod p if and only if d is odd. □

The definition of the discriminant above Proposition 3.3 is not the usual way how the discriminant is computed in practice. Instead, it can be derived purely from the coefficients of the polynomial.

The final statement we need is a multiplicative rule for the discriminant of a polynomial.

Lemma 3.4. *Let p be prime and let $f, g \in \mathbb{F}_p[x]$, possibly reducible. Then*

$$\Delta(fg) = \Delta(f)\Delta(g)\beta^2$$

for some $\beta \in \mathbb{F}_p$.

Proof. Let $\alpha_1, \ldots, \alpha_k$ and $\alpha_{k+1}, \ldots, \alpha_l$ be the roots of f and g respectively. Then $\alpha_1, \ldots, \alpha_l$ are the roots of fg, so

$$\Delta(fg) = \prod_{1 \le i < j \le l} (\alpha_i - \alpha_j)^2 = \Delta(f)\Delta(g) \prod_{1 \le i \le k < j \le l} (\alpha_i - \alpha_j)^2.$$

Let $\beta = \prod_{1 \le i \le k < j \le l}(\alpha_i - \alpha_j)$, we will show that $\beta \in \mathbb{F}_p$ by showing that $\sigma(\beta) = \beta$. If $1 \le i \le k$, then $\sigma(\alpha_i) = \alpha_{\tau(i)}$, where $1 \le \tau(i) \le k$, since it is a root of f again, even if f is reducible. Likewise for the roots α_j with $k+1 \le j \le l$. So,

$$\sigma(\beta) = \prod_{1 \le i \le k < j \le l} (\alpha_{\tau(i)} - \alpha_{\tau(j)}) = \prod_{1 \le i \le k < j \le l} (\alpha_i - \alpha_j) = \beta,$$

since σ is injective, so we get a permutation of the terms of β. Thus $\beta \in \mathbb{F}_p$. □

We can now state and prove the correctness of the compositeness test of Grantham. The test takes as input an odd integer $n > 1$ and a polynomial $f \in \mathbb{Z}/n\mathbb{Z}[x]$ of degree d such that $\gcd(n, f(0)\Delta(f)) = 1$. We call n a Frobenius probable prime with respect to f if it passes Algorithm 2 with parameters n and f.

Algorithm 2 Grantham's test

```
1:  function GRANTHAM(n, f)
2:      f_0(x) = f
3:      for 1 ≤ i ≤ d do                                  ▷ Factorization step
4:          F_i(x) = gcd(x^{n^i} − x, f_{i−1}(x))
5:          f_i(x) = f_{i−1}(x)/F_i(x)
6:      end for
7:      if one of the gcds does not exist or f_d(x) ≠ 1 then
8:          return n is composite
9:      end if
10:     for 2 ≤ i ≤ d do                                  ▷ Frobenius step
11:         if F_i(x^n) ≠ 0 mod F_i(x) then
12:             return n is composite
13:         end if
14:     end for
15:     S = Σ_{2|i}^{d} deg(F_i(x))/i                     ▷ Jacobi step
16:     if (−1)^S ≠ (Δ(f)/n) then
17:         return n is composite
18:     end if
19:     return did not prove compositeness
20: end function
```

Proposition 3.5. *Algorithm 2 is correct and runs in $\widetilde{\mathcal{O}}(\log(n)^2 d^2)$, where $d = \deg(f)$.*

Proof. We first prove the correctness of the algorithm. Suppose that n is prime. Using Proposition 3.1, we see that in the Factorization step we have for all $1 \leq i \leq d$ that $F_i(x)$ is the product of the irreducible factors of f of degree i. Then since f has degree d, we expect that $f_d(x) = 1$. It is possible that $f_j(x) = 1$ for $j < d$, namely if f is reducible over $\mathbb{F}_n$. If n is not prime, then it is possible that one of the gcds does not exist. This happens if one of the leading coefficients of a polynomial in the process of the Euclidean algorithm is not invertible mod n.
Next up is the Frobenius step. Suppose again that n is prime. Proposition 3.2 implies that $x^n \bmod F_i(x)$ is again a root of $F_i(x)$.
Finally, we arrive at the Jacobi step. Again, we suppose that n is prime. Let $a_i = \deg(F_i(x))/i$, then a_i is the number of irreducible factors of degree i of $F_i(x)$. Write $F_i(x) = \prod_{j=1}^{a_i} G_{i,j}(x)$ for all i, where the $G_{i,j}$ are irreducible over $\mathbb{F}_n$. Then by first using Lemma 3.4 twice and then Proposition 3.3, we see that since the Jacobi symbol is multiplicative:

$$\begin{aligned}\left(\frac{\Delta(f)}{n}\right) &= \prod_{i=1}^{d}\left(\frac{\Delta(F_i)}{n}\right) = \prod_{i=1}^{d}\prod_{j=1}^{a_i}\left(\frac{\Delta(G_{i,j})}{n}\right) = \prod_{\substack{i=1\\ i \text{ odd}}}^{d} 1^{a_i} \cdot \prod_{\substack{i=2\\ i \text{ even}}}^{d} (-1)^{a_i} \\ &= (-1)^{\sum_{2|i}^{d} a_i} = (-1)^S.\end{aligned}$$

Now we determine the running time of the algorithm. In the Factorization step we can first compute $x^{n^i} \bmod f_{i-1}(x)$ and then $\gcd(x^{n^i} - x, f_{i-1}(x))$. We thus have to compute $x^n, x^{n^2}, \ldots, x^{n^d} \bmod g$, where g has degree $\leq d$. This can be done in $\widetilde{\mathcal{O}}(\log(n)^2 d^2)$ using binary exponentiation and fast multiplication [6, Chapter 4]. The d quotients can be computed in at most $\widetilde{\mathcal{O}}(\log(n) d^2)$ [6, Chapter 17].
In the Frobenius step we can use the computation of $x^n \bmod f$ from before by reducing it modulo $F_i(x)$ for all i. Given three polynomials $g_1, g_2, g_3 \in \mathbb{Z}/n\mathbb{Z}[x]$ of degree $\leq d$, we can compute $g_1(g_2(x)) \bmod g_3(x)$ in $\widetilde{\mathcal{O}}(\log(n) d^2)$, by computing $g_2(x)^2, \ldots, g_2(x)^d \bmod g_3(x)$ and substituting that in $g_1(x)$. So, we can compute $F_i(x^n)$ for $i = 2, \ldots, d$ in

$$\widetilde{\mathcal{O}}(\sum_{i=2}^{d} (\deg(F_i)^2 \log(n)).$$

Now, we know that $\sum_{i=1}^{d} \deg(F_i) = d$, since (n, f) passed the Factorization step. So, $\sum_{i=1}^{d} \deg(F_i)^2 \leq d^2$, which implies that the running time of the Frobenius step is $\widetilde{\mathcal{O}}(\log(n) d^2)$.
Finally, we consider the Jacobi step. We only have to do a simple Jacobi symbol computation, which can be done in $\widetilde{\mathcal{O}}(\log(n)^2)$, since the computation is analogous to the Euclidean algorithm.
We see that the running time is dominated by the Factorization step, which takes $\widetilde{\mathcal{O}}(\log(n)^2 d^2)$. □

Example 3.6. Let's take $n = 1159$ and $f = x^2 + 389x + 596$. Before we do Grantham's test, we first we have to check that $\gcd(n, f(0)\Delta(f)) = 1$. Now, $\Delta(f) = 389^2 - 4 \cdot 596 = 585 \bmod n$, then using the Euclidean algorithm, we can check that $\gcd(1159, 596 \cdot 585) = 1$. So we can begin the test! We start with the Factorization step. We compute $x^n = 1158x + 770 \bmod f$ and $x^{n^2} = x \bmod f$. Furthermore, $F_1 = \gcd(1157x + 770, f) = 1$, so $f_1 = f$ and $F_2 = \gcd(0, f) = f$, so $f_2 = 1$. Hence, (n, f) passes the Factorization step. Next up is the Frobenius step. Since $F_2 = f$, we can use our previous results to compute

$$F_2(x^n) = f(1158x + 770) = x^2 + 389x + 596 = f(x) = 0 \bmod F_2(x).$$

Hence, (n, f) also passes the Frobenius step. Finally, we use the Jacobi symbol to check that $\left(\frac{585}{n}\right) = (-1)^1$. So (n, f) passes all steps of the test, hence n is a Frobenius probable prime with respect to f.
This might lead us to believe that n is prime, but lo and behold, if we take another polynomial such as $g = x^2 + 574x + 795$, then we get a different result. We still have that $\gcd(n, g(0)\Delta(g)) = \gcd(1159, 575 \cdot 617) = 1$. But this time

$$\gcd(x^n - x, g) = \gcd(1074x + 152, g) = 1 \quad \text{and} \quad \gcd(x^{n^2} - x, g) = \gcd(785x + 19, g) = 1.$$

Hence $f_2 \neq 1$, so (n, g) does not pass the Factorization step, thus n is composite. This test usually does not give a factorization of n, even when it shows that n is composite. Nevertheless, it might be pleasing to know that $n = 19 \cdot 61$.

Now that we know how the test works, we can start to look at its interesting properties. Grantham not only showed that his test is correct, he also showed that using a variant of his test, a composite number passes the test with probability less than $1/7710$ [22]. In this variant Grantham only uses polynomials of degree 2 of which the discriminant is not a square mod n and he incorporates a Miller-Rabin step in his test. He then carefully computed an upper bound for the number of such quadratic polynomials f such that (n, f) passes his test to arrive at that probability.

Furthermore, Grantham also showed that his quadratic test is efficient, he shows that it runs in $\widetilde{\mathcal{O}}(\log(n)^2)$. The Miller-Rabin test has the same complexity, but Rabin [37] only showed that a composite number passes the Miller-Rabin test with probability less than $1/4$. Hence it might be natural to think that it is better to use Grantham's test, since $1/7710 < 1/4$. However, it is possible that the coefficients of the complexity function of Grantham's test are much larger. That's why Grantham computed the explicit number of bit operations his quadratic test takes and showed that it is fewer than 3 Miller-Rabin tests. Now, $(1/4)^3 = 1/64 > 1/7710$, hence Grantham's test is faster in proving strong probabilities.

There have been a lot of optimizations of Grantham's test such as [43, 16, 33, 34]. But they all work in extensions of degree at most 2. This is perhaps because computations in larger extensions take more time. Nevertheless, we will consider those larger extensions later on in this chapter and we will discuss the problems arising when you try to construct those finite fields when you don't know if your input n is prime or not.

Grantham's test is not only useful in a practical setting, it also has an important place in the theory of primality and compositeness tests. Grantham showed that his test can be seen as a generalization of several other tests. An interesting example of this is a classical test that uses Lucas sequences. A Lucas sequence with parameters $P, Q \in \mathbb{Z}$ is of the form:

$$U_0 = 0,\ U_1 = 1,\ U_m = PU_{m-1} - QU_{m-2} \text{ (for } m \geq 2).$$

We first prove some basic properties of Lucas sequences.

Lemma 3.7. *Let p be prime and set $f(x) = x^2 - Px + Q$. If α and β are the roots of f in $\mathbb{F}_{p^2}$, then*

$$U_m = \frac{\alpha^m - \beta^m}{\alpha - \beta} \text{ for all } m \geq 0.$$

Proof. We will prove it by using strong induction on m. The base cases $m = 0$ and $m = 1$ can easily be checked. Now suppose the formula holds for all $0 \leq k \leq m$. Then

$$\begin{aligned} U_{m+1} = PU_m - QU_{m-1} &= P\frac{\alpha^m - \beta^m}{\alpha - \beta} - Q\frac{\alpha^{m-1} - \beta^{m-1}}{\alpha - \beta} \\ &= \frac{(P\alpha - Q)\alpha^{m-1} - (P\beta - Q)\beta^{m-1}}{\alpha - \beta} = \frac{\alpha^2\alpha^{m-1} - \beta^2\beta^{m-1}}{\alpha - \beta} = \frac{\alpha^{m+1} - \beta^{m+1}}{\alpha - \beta}. \end{aligned}$$

Here we used in the second to last equality that α and β are roots of f. □

Proposition 3.8. *Given $P, Q \in \mathbb{Z}$, write $U_m = U_m(P, Q)$ and $\Delta = P^2 - 4Q$. If p is prime and $p \nmid 2Q\Delta$, then $U_{p-\left(\frac{\Delta}{p}\right)} = 0 \bmod p$.*

Proof. Again let $f(x) = x^2 - Px + Q$ with roots α and β in $\mathbb{F}_{p^2}$. Then since $p \nmid \Delta = \Delta(f)$, we know that $\alpha \neq \beta$. First suppose that $\left(\frac{\Delta}{p}\right) = 1$. Then $\alpha, \beta \in \mathbb{F}_p$, so $\alpha^{p-1} = \beta^{p-1} = 1 \bmod p$. Hence by Lemma 3.7,

$$U_{p-1} = \frac{1 - 1}{\alpha - \beta} = 0 \bmod p.$$

Now suppose that $\left(\frac{\Delta}{p}\right) = -1$, then f is irreducible over $\mathbb{F}_p$. Then by Proposition 3.2, we know that $\alpha^p = \beta$ and $\beta^p = \alpha$. Thus

$$U_{p+1} = \frac{\beta\alpha - \alpha\beta}{\alpha - \beta} = 0 \bmod p.$$ □

Proposition 3.8 naturally gives rise to a compositeness test. Given a positive integer n, we can randomly choose $0 \leq P, Q \leq n - 1$ and compute $U_{n-\left(\frac{\Delta}{n}\right)} \bmod n$ using the Jacobi symbol and a fast method to compute a term in a recurrence sequence [30]. If $\gcd(n, 2Q\Delta) = 1$ and $U_{n-\left(\frac{\Delta}{n}\right)} \neq 0 \bmod n$, then we know for sure that n is composite. But as usual, the converse is not always true. We say n is a *Lucas pseudoprime* with respect to P and Q if $\gcd(n, 2Q\Delta) = 1$ and $U_{n-\left(\frac{\Delta}{n}\right)} = 0 \bmod n$.

The proof of Proposition 3.8 already hints at a possible connection between Lucas pseudoprimes and Frobenius pseudoprimes. We will make this explicit in the next proposition. We follow the proof from Grantham [21].

Proposition 3.9. *If n is a Frobenius pseudoprime with respect to $x^2 - Px + Q$, then n is a Lucas pseudoprime with respect to P and Q.*

Proof. Let $f = x^2 - Px + Q$. We will try to use Lemma 3.8 again. First suppose that $\left(\frac{\Delta}{n}\right) = 1$. Then by the Jacobi step, we know that $S = 0$, which means that $x^n = x \bmod f$. Now, $x(x - P) = -Q \bmod f$, so $x \cdot (\frac{x-P}{-Q}) = 1 \bmod f$, which means that x is invertible mod f. Hence $x^{n-1} = 1 \bmod f$. Explicitly, that means that $x^{n-1} = 1 + f(x)g(x) \bmod n$ for some $g(x) \in \mathbb{Z}/n\mathbb{Z}[x]$. Now note that

$$f(P - x) = P^2 - 2Px + x^2 - P^2 + Px + Q = x^2 - Px + Q = f(x) = 0 \bmod f(x),$$

which means that $P - x$ is a root of $f(x)$. Now we use composition of polynomials to see that

$$(P-x)^{n-1} = x^{n-1} \circ (P-x) = (1+f(x)g(x)) \circ (P-x) = 1+f(P-x)g(P-x) = 1 \bmod f.$$

Also note that

$$(2x - P)^2 = 4x^2 - 4xP + P^2 = 4Px - 4Q - 4xP + P^2 = P^2 - 4Q = \Delta(f) \bmod f.$$

Then since $\Delta(f)$ is invertible mod n, we see that $2x - P$ is invertible mod f, since $(2x - P) \cdot \frac{2x-P}{\Delta(f)} = 1 \bmod f$. Thus, we can now use Lemma 3.8 to see that

$$U_{n-1} = \frac{x^{n-1} - (P - x)^{n-1}}{x - (P - x)} = \frac{1 - 1}{2x - P} = 0 \bmod f.$$

Now suppose that $\left(\frac{\Delta}{n}\right) = -1$. This time the Jacobi step tells us that $S = 1$. So, by the Factorization step, we know that $\gcd(x^n - x, f) = 1$. This implies that $x^n \neq x \bmod (f, p^k)$ for all prime powers $p^k \mid n$. Now, if $p \mid n$, then $f(x)$ has precisely two roots mod p, since $\gcd(p, \Delta(f)) = 1$. Then by Hensel's lemma, we know that $f(x)$ also has precisely two roots mod p^k. We already saw the other root, namely $P - x$. Hence $x^n = P - x \bmod (f, p^k)$. Then by the Chinese remainder theorem, we know that $x^n = P - x \bmod (f, n)$. Finally, we know that

$$x = x^{n^2} = (P - x)^n \bmod (f, n),$$

so we can use Lemma 3.8 to see that

$$U_{n+1} = \frac{x^{n+1} - (P - x)^{n+1}}{x - (P - x)} = \frac{x(P - x) - (P - x)x}{2x - P} = 0 \bmod f.$$

Thus, n is a Lucas pseudoprime with respect to P and Q. □

We gave the proof of 3.9 since it uses all of the steps of Grantham's test and also because it uses composition of polynomials. Later on in this chapter we will see some more uses of polynomial composition. Having said that, we could have showcased several

other types of pseudoprimes that have a strong relation with Frobenius pseudoprimes. Grantham does this in Chapters 4 and 5 of his paper [21].

We already said if n is not prime, then $\mathbb{Z}/n\mathbb{Z}$ isn't a field. Fortunately for us, it will still have a prime divisor p which we can work with. Given $f \in \mathbb{Z}/n\mathbb{Z}[x]$ of degree d, we can view it modulo p. Suppose that f is square-free modulo p, which we can check by computing $\gcd(f, f')$. Say f factors as $f_1 \cdots f_r$ over $\mathbb{F}_p$, where the f_i are irreducible over $\mathbb{F}_p$. Then we get the following ring homomorphisms:

$$\mathbb{Z}/n\mathbb{Z}[x]/(f) \to \mathbb{Z}/p\mathbb{Z}[x]/(f) \to \mathbb{F}_p[x]/(f_i) \cong \mathbb{F}_{p^{\deg(f_i)}}. \tag{3.1}$$

This means that even if n is not prime, we can still work in a finite field that is strongly related to n. This is of course a bit vague, so let's look at an example. We have already seen an instance where this idea is used, namely in the proof of the Lucas-Lehmer test in paragraph 2.3, so we will only sketch the test here. In that test we have a prime p and the integer $q = 2^p - 1$ that we want to test for primality. The test is usually formulated in terms of a recurrence relation, but for this example it is better to describe it in terms of finite field extensions. The test uses the fact that 3 is always a non-square mod q, so if q is prime, then we can create the finite field $\mathbb{F}_q(\sqrt{3}) = \mathbb{F}_q[x]/(x^2 - 3)$ of q^2 elements. Furthermore, if q is not prime, then it has a prime divisor r with $r \leq \sqrt{q}$. We can then use the homomorphism from $\mathbb{Z}/n\mathbb{Z}[x]/(x^2 - 3)$ to $G = \mathbb{F}_r[x]/(x^2 - 3)$ to show that G^* contains an element of order $q + 1 > r^2 - 1 \geq \#G^*$. This gives a contradiction, proving that q is prime. For a rigorous proof of this test we refer back to paragraph 2.3.

Grantham [22] also used the map (3.1) to show that a composite number is unlikely to pass his test. Like we said before, he did this is the case that f is of degree 2. Furthermore, he chose the coefficients of $f = x^2 + ax + b$ such that $\left(\frac{\Delta(f)}{n}\right) = -1$. He does this because then n has at least one prime divisor p such that $\left(\frac{\Delta(f)}{p}\right) = -1$. Hence, if we reduce modulo that prime divisor, then $\mathbb{F}_p[x]/(f)$ is a finite field of p^2 elements. Then if (n, f) passes his test, then we know that $x^n = x^p \bmod (f, p)$, since x^n is again a root of f and it is not equal to $x \bmod (f, p)$. This implies that $x^{n-p} = 1 \bmod (f, p)$. Now, $\mathbb{F}_{p^2}^*$ is cyclic so there are $\gcd(n - p, p^2 - 1)$ elements $y \in \mathbb{F}_{p^2}$ such that $y^{n-p} = 1 \bmod (f, p)$. Grantham used this to compute an upper bound for the probability that you choose coefficients a, b such that $x^n = x^p \bmod (f, p)$.

We will now explore the option of using polynomials f of higher degree to see if they bring something new to the table. We will also use the fact that $\mathbb{F}_{p^d}^*$ is cyclic, but in a slightly different way.

Lemma 3.10. *Suppose that p is a prime divisor of n and $f \in \mathbb{Z}/n\mathbb{Z}[x]$ is irreducible over $\mathbb{F}_p$ and of degree d. If (n, f) passes the Frobenius Step of Grantham's test, then*

$$x^n = x^{p^m} \bmod (f, p) \text{ for some } 1 \leq m \leq d - 1.$$

Proof. We know that $x^n \bmod (f, p)$ is again a root of f and different from x. The claim now follows from Proposition 3.2. □

Now, if a is the order of $x \bmod (f,p)$ in the group $\mathbb{F}_{p^d}^* = (\mathbb{F}_p[x]/(f))^*$, then $x^n = x^{p^m} \bmod (f,p)$ implies that $n = p^m \bmod a$. The idea will be that increasing d will on average increase a, which means that $x^n = x^{p^m} \bmod (f,p)$ will be less likely. Using this idea, we can formulate a sufficient condition for primality.

Theorem 3.11. *Suppose that all conditions of Lemma* 3.10 *hold. Suppose furthermore that f is primitive, i.e., $x \bmod (f,p)$ has maximal order $p^d - 1$. Finally, suppose that $d > \log_2(n)$ and that n is not a perfect power. Then n is prime.*

Proof. From Lemma 3.10 we know that $n = p^m \bmod p^d - 1$ for some $1 \leq m \leq d-1$. Now, $m < d$, so $p^d - 1 > p^m$. Furthermore, $d > \log_2(n)$ and $p \geq 2$, hence $p^d - 1 > n - 1$, thus $p^d - 1 \geq n$. This means that $n = p^m$ is also true as an equality in $\mathbb{Z}$. Then since n is not a perfect power, we know that $m = 1$, hence n is prime. □

We saw in the proof of Proposition 3.5 that we can compute $x^n \bmod f$ in $\widetilde{\mathcal{O}}(\log(n)^2 d)$ and that the rest of the Frobenius step can then be performed in $\widetilde{\mathcal{O}}(\log(n) d^2)$. Hence, if f has degree $\mathcal{O}(\log(n))$, then we can check that (n, f) passes the Frobenius step in $\widetilde{\mathcal{O}}(\log(n)^3)$.

Since Theorem 3.11 has so many assumptions, it is important to check that they are sensible ones. For example, we need a polynomial f that is irreducible over $\mathbb{F}_p$ and of large degree, where p is some unknown prime divisor of n. Using the Legendre symbol trick, we could do it for degree 2 polynomials, but we have not yet seen any methods to go beyond that. Later on in this chapter we will look at several ways to construct such polynomials, so we will see that that assumption is fine. All the other assumptions are also fine, except for the one that says that $x \bmod (f,p)$ has to be a primitive root. Since $\mathbb{F}_{p^d}^*$ is cyclic, we know that $\phi(\#\mathbb{F}_{p^d}^*) = \phi(p^d - 1)$ elements in $\mathbb{F}_{p^d}^*$ are generators for that group, where ϕ is the Euler phi function. There is a classical lower bound for this function [39, Theorem 15], that says that for all integers $m > 2$,

$$\phi(m) > \frac{m}{e^\gamma \log\log m + \frac{3}{\log\log m}}.$$

This means that after randomly choosing $\mathcal{O}(\log(\log(p^d - 1)))$ elements in $\mathbb{F}_{p^d}^*$, we expect to get one primitive root. So by trying multiple polynomials f, Theorem 3.11 can be turned into a probabilistic primality test. However, the only known way to check if an element generates the group requires the factorization of the order of the group. But, we can't expect to know the complete factorization of $n^d - 1$, let alone $p^d - 1$. This is why we will improve on Theorem 3.11. The next proposition relaxes the conditions of Theorem 3.11 in two ways. We don't have to restrict ourselves to the element $x \bmod (f,p)$ and we don't need a generator for $\mathbb{F}_{p^d}^*$, just an element of fairly large order.

Proposition 3.12. *Suppose that all conditions of Lemma* 3.10 *hold. Suppose that we have an element $g(x) \bmod (f,p)$ that has order at least p^{d-2} and satisfies $g(x^n) = g(x)^n \bmod (f,p)$. If $d > \log_2(n) + 2$ and if n is not a perfect power, then n is prime.*

Proof. From Lemma 3.10 we again know that $x^n = x^{p^m} \bmod (f,p)$ for some $1 \leq m \leq d-1$. Note that

$$g(x)^n = g(x^n) = g(x^{p^m}) = g(x)^{p^m} \bmod (f,p),$$

because the freshman's dream implies that $g(x^p) = g(x)^p \bmod (f,p)$, the equality then follows from induction. Let a be the order of $g(x) \bmod (f,p)$, then $n = p^m \bmod a$. So we get the same congruence as in the proof of Theorem 3.11, apart from the fact that a is now somewhat smaller than $p^d - 1$. We will now show that a is still large enough. Note again that $p \geq 2$, hence $p^{d-2} > n$. If $m = d-1$, then $p^m > a$, which is a bit worrying. But since n and p^m are both integers that are divisible by p, we see that $\frac{n}{p} = p^{m-1} \bmod a$, where $a > \frac{n}{p}, p^{m-1}$, hence $\frac{n}{p} = p^{m-1}$. So, we see that n is a power of a prime, thus by the assumption that n is not a perfect power, n has to be prime. □

The question now becomes how many elements in $\mathbb{F}^*_{p^d}$ have order at least p^{d-2}. Using the fact again that $\mathbb{F}^*_{p^d}$ is cyclic, we see that there are

$$\sum_{\substack{a \mid p^d-1 \\ a \geq p^{d-2}}} \phi(a)$$

such elements in $\mathbb{F}^*_{p^d}$. Experimentally, this seems to be around $p^d - \mathcal{O}(p^{d-1})$, but we have no proof of this. If we could prove it, then we would know that we have a probability of about

$$1 - \frac{p^d - p^{d-1}}{p^d} = 1 - \frac{p-1}{p} = \frac{1}{p}$$

that we choose an element that does *not* have an order high enough for our purposes, so that would be a an enormous improvement.

It should be noted that we don't actually know what p is, so these probabilities might seem a bit meaningless. However, if we check that n is not divisible by all primes below a certain bound B, then we know that $p > B$ for all primes dividing n, which makes the probability $1/p$ very small. Doing this also allows us to choose d somewhat smaller than in Proposition 3.12. There we stated that $d > \log_n(n) + 2$, since $p \geq 2$. But if we know that $p \geq B$, then $d > \log_B(n) + 2$ suffices. This is very helpful in practice, since it greatly increases the speed of the finite field arithmetic. However, it does not improve the complexity of the algorithm, since $\log_B(n) = \frac{\log_2(n)}{\log_2(B)}$ and B has to be polynomial in $\log(n)$.

There are still more improvements to be discussed, the next one uses the group structure of $\mathbb{F}^*_{p^d}$.

Lemma 3.13. *The elements $g(x) \bmod (f,p)$ such that $g(x^n) = g(x)^n \bmod (f,p)$ form a group under multiplication.*

Proof. First note that $1 = 1^n \bmod (f,p)$. Now suppose that $g(x^n) = g(x)^n \bmod (f,p)$ and $h(x^n) = h(x)^n \bmod (f,p)$. Then

$$(gh)(x^n) = g(x^n)h(x^n) = g(x)^n h(x)^n = (g(x)h(x))^n \bmod (f,p). \qquad \square$$

This means that we don't necessarily need one element that has big order, we just need a set of elements $g_1(x), \ldots, g_m(x) \bmod (f, p)$ such that

$$\text{lcm}(\text{order}(g_1(x)), \ldots, \text{order}(g_r(x)))$$

is at least p^{d-2}. Unfortunately, there is no unconditional polynomial time algorithm known that produces such a set. But, we can look how far we can get by assuming the Riemann hypothesis, or some generalization of it. In the case that $d = 1$, it is known that under assumption of the extended Riemann hypothesis (ERH), the elements $1, 2, \ldots, 2\log(p)^2$ generate $\mathbb{F}_p^*$ [5]. Furthermore, in the case that $d = 2$, it is known that if we assume ERH, then the set

$$\{a_1 + a_2 x \mid a_i \in \mathcal{O}(\log(p)^B)\}$$

generates $\mathbb{F}_{p^2}^*$ for some constant B [44]. However, a similar statement is not known for $d > 2$. But, there is a deterministic polynomial time algorithm assuming ERH that given a prime p, produces a model for $\mathbb{F}_{p^d}$ with basis $\theta_1, \ldots, \theta_d$ over $\mathbb{F}_p$ such that

$$\{\sum a_i\theta_i : |a_i| \leq Cd^{Dd}\log(p)^{\max(d-1,2)}\}$$

generates $\mathbb{F}_{p^d}^*$ for certain constants C and D [11]. But, we can't use this construction, since we don't know if our number n that we want to test is prime. Hence the currently known results are not strong enough to make Theorem 3.11 into a primality proving algorithm.

However, there is a test that uses some of same ideas that actually is fully deterministic and can prove primality in polynomial time. This algorithm is the celebrated AKS test from 2002, by Agrawal, Kayal and Saxena. In their original paper [2] they take $f = x^r - 1$ where r is of size $\mathcal{O}(\log(n)^5)$ and show that to prove that n is prime, it is enough to check that

$$(x + a)^n = x^n + a \bmod f, \text{ for all } 1 \leq a \leq l,$$

where l is also polynomially bounded by $\log(n)$. They do this by constructing a large set of elements $g(x)$ such that $g(x^n) = g(x)^n \bmod f$ and show that this is only possible when n is a power of a prime. The full algorithm runs in $\widetilde{\mathcal{O}}(\log(n)^{10.5})$. We tried similar constructions, Lemma 3.13 for example is Lemma 4.6 in [2]. But we were not as successful, since the AKS test is fully deterministic. However, the running time of our test is also polynomial in $\log(n)$ and is also quite a lot faster. This is because the degree of our f is only $\mathcal{O}(\log(n))$. But this is comparing apples to oranges, since our algorithm isn't deterministic at all. But we certainly can compare the original AKS test to its improvements. Lenstra and Pomerance [26] showed that the AKS test can be adapted to an algorithm that runs in $\widetilde{\mathcal{O}}(\log(n)^6)$, which is a lot better. They did this by showing that you only need a polynomial f of degree $\mathcal{O}(\log(n)^2)$ (with certain properties) and by lowering the bound on l.

3.2 Irreducible polynomials modulo divisors of n

The polynomials f used in the AKS test are not necessarily irreducible modulo p for some $p \mid n$. Instead, they are constructed such that they have an irreducible factor of large enough degree modulo some $p \mid n$. We will now look at various techniques to construct irreducible polynomials modulo a prime divisor of n, some of which will be new. We have seen that constructing such a polynomial of degree 2 isn't very difficult. If we have some a with $\left(\frac{a}{n}\right) = -1$, then $\left(\frac{a}{p}\right) = -1$ for some $p \mid n$, hence $x^2 - a$ is irreducible over $\mathbb{F}_p$. We can extend this idea in multiple ways; the first one produces polynomials with degree a power of 2.

Proposition 3.14. *Let $p = 1 \bmod 4$ be a prime and suppose we have some a with $\left(\frac{a}{p}\right) = -1$. Then $f = x^{2^m} - a$ is irreducible over $\mathbb{F}_p$ for all $m \in \mathbb{Z}_{\geq 0}$.*

Proof. Note that by Proposition 3.1, if we can show that $\gcd(x^{p^{2^{m-1}}} - x, f) = 1$ and $f \mid x^{p^{2^m}} - x$, then f is irreducible over $\mathbb{F}_p$, because then the degree of all irreducible factors of f divide 2^m, but not 2^{m-1}. Now, p is odd and $\phi(2^m) = 2^{m-1}$, hence $p^{2^{m-1}} = 1 \bmod 2^m$. This means that

$$x^{p^{2^{m-1}}} = x a^{\frac{p^{2^{m-1}}-1}{2^m}} \bmod f,$$

since $x^{2^m} = a \bmod f$. Note that

$$p^{2^{m-1}} - 1 = (p-1)(p+1)(p^2+1)(p^4+1)\cdots(p^{2^{m-2}}+1).$$

Now we use that $p = 1 \bmod 4$ to see that $p^{2^k} + 1 = 2 \bmod 4$ for all $k \in \mathbb{Z}_{\geq 0}$. This implies that

$$(p+1)(p^2+1)(p^4+1)\cdots(p^{2^{m-2}}+1)$$

contains $m-1$ factors of 2. Hence

$$\frac{p^{2^{m-1}} - 1}{2^m} = \frac{p-1}{2} \cdot r,$$

where r is an odd integer. Now we use the assumption that $\left(\frac{a}{p}\right) = -1$ to get that $a^{\frac{p-1}{2}} = -1 \bmod p$. Thus

$$x^{p^{2^{m-1}}} = x \cdot (-1)^r = -x \bmod f.$$

This implies that

$$\gcd(x^{p^{2^{m-1}}} - x, f) = \gcd(-2x, f) = 1.$$

Finally, we check that $f \mid x^{p^{2^m}} - x$. Note that

$$\frac{p^{2^m} - 1}{2^m} = (p-1) \cdot s,$$

where s is odd, since we now have an extra factor of 2. Then

$$x^{p^{2^m}} = x \cdot 1^s = x \bmod f.$$

Hence $f \mid x^{p^{2^m}} - x$, so f is irreducible over $\mathbb{F}_p$. □

This proposition only helps us in the case that $p = 1 \bmod 4$. Fortunately, a similar polynomial can be used in the case that $p = 3 \bmod 4$. For a proof we refer to [26, Lemma 4.2].

Proposition 3.15. *Let $p = 3 \bmod 4$ be a prime and suppose we have some a with $\left(\frac{a^2+4}{p}\right) = -1$. Then $f = x^{2^{m+1}} - ax^{2^m} - 1$ is irreducible over $\mathbb{F}_p$ for all $m \in \mathbb{Z}_{\geq 0}$.*

It should be noted that there currently is no deterministic polynomial time algorithm that can find non-residues modulo a prime p, let alone modulo n. However, since half of the integers mod p are non-residues, we can simply pick random numbers a and hope that $\left(\frac{a}{p}\right) = -1$. This works well in practice, since you expect to find one after only two tries. Then since the Legendre symbol is (also) multiplicative in the second argument, we can also expect to find numbers a with $\left(\frac{a}{n}\right) = -1$ very quickly, assuming that n is not a perfect square.

Given Proposition 3.14 and 3.15, we are very close to constructing irreducible polynomials modulo a prime divisor of a given odd integer n. The problem is that if $n = 1 \bmod 4$ for example, then the divisors of n can still be both 1 mod 4 or 3 mod 4. This makes it harder to use one of the propositions, since we know that $\left(\frac{a}{p}\right) = -1$ for some $p \mid n$, but we don't know what $p \bmod 4$ is. You could maybe come up with some argument that shows that at least one of $x^{2^m} - a$ and $x^{2^{m+1}} - ax^{2^m} - 1$ irreducible modulo some $p \mid n$, but it is much more satisfying to just have one. We will show that this can be achieved in a compositeness/primality test setting.

Theorem 3.16. *Given positive integers n and m with n odd, there is an algorithm that either proves that n is composite, or returns a polynomial of degree 2^m that is irreducible modulo some prime divisor p of n.*

Proof. We will use that if n is prime, then we can compute square roots modulo n given some quadratic non-residue. This can be done using the Tonelli-Shanks algorithm for example [1]. Furthermore, we will also use that if p is prime, then -1 is a square mod p if and only if $p = 1 \bmod 4$.
First suppose that $n = 1 \bmod 4$. Try to compute a square root of $-1 \bmod n$. If the process fails, then we know that n is composite. If the process doesn't fail, then we know that all prime divisors of n are 1 mod 4, since -1 is a square modulo all of those divisors. Now find an integer a with $\left(\frac{a}{n}\right) = -1$, then the claim follows from Proposition 3.14.
Now suppose that $n = 3 \bmod 4$. Then we know that n has a prime divisor p that is also 3 mod 4, hence $\left(\frac{-1}{p}\right) = -1$. Find an integer a with $\left(\frac{a^2+4}{n}\right) = -1$, then $\left(\frac{a^2+4}{q}\right) = -1$ for some $q \bmod n$. We can circumvent the problem that q might be unequal to p by using

the following neat trick. Note that $\left(\frac{-(a^2+4)}{n}\right) = 1$. Try to compute the square root of $-(a^2+4) \bmod n$ using the square root algorithm. If it fails, then n is composite. If it doesn't, then $\left(\frac{-(a^2+4)}{r}\right) = 1$ for all prime divisors r of n. This means that

$$\left(\frac{-1}{r}\right) = \left(\frac{a^2+4}{r}\right) \text{ for all prime divisors } r \text{ of } n.$$

Hence $\left(\frac{a^2+4}{p}\right) = -1$, where p is that prime divisor of n which is 3 mod 4. The claim now follows from Proposition 3.15. □

Theorem 3.16 is very useful for Theorem 3.11. Because now we can construct a polynomial that satisfies the assumption. It's also efficient, since the Tonelli-Shanks algorithm is efficient. We could stop here since we got what we needed, but there are a lot more interesting ways to construct these kinds of polynomials, so we will consider some of them. The first one is a generalization of the one we just saw. A proof of the following proposition can be found in [27, Theorem 3.75].

Proposition 3.17. *Let $p = 1 \bmod 4$ be prime and let r also be prime and suppose that a is not an rth power mod p. Then $x^{r^m} - a$ is irreducible over $\mathbb{F}_p$ for all $m \in \mathbb{Z}_{\geq 0}$.*

Note that if n is prime, then we can find such a if $r \mid (\mathbb{Z}/n\mathbb{Z})^*$. So, Proposition 3.17 can be used in a compositeness/primality test setting in the following way. Suppose that $n = 1 \bmod 4$, then we can check that -1 is a square mod n like in the proof of Theorem 3.16 to show that all primes $p \mid n$ are 1 mod 4. Now write $n - 1 = r^k s$, where $r \nmid s$. Then find an element a of order r^k modulo n. To make sure that a also has order r^k modulo its prime divisors, we check that $\gcd(a^{r^{k-1}} - 1, n) = 1$. If it is not 1, then n is composite, else a has order r^k modulo all $p \mid n$. Now, $r^{k+1} \nmid n - 1$, which implies that n has a prime divisor p such that $r^{k+1} \nmid p - 1$, because otherwise $n = 1 \bmod r^{k+1}$, which gives a contradiction. Thus we can apply Proposition 3.17 to get that $x^{r^m} - a$ is irreducible mod p for all $m \in \mathbb{Z}_{\geq 0}$. The bottleneck of this method is that you have to (partially) factor $n - 1$.

We have already seen that we can make irreducible polynomials mod $p \mid n$ of degree 2^m. We found a different way to construct such polynomials using so-called self-reciprocal polynomials.

Definition 3.18. *Given a polynomial f of degree m, the reciprocal polynomial f^* of f is given by*

$$f^*(x) = x^n f(1/x).$$

Furthermore, f is called self-reciprocal if $f(x) = f^*(x)$.

The coefficients of $f^*(x)$ are the same as f, but in reverse. For example, if $f(x) = 5x^4 + x^2 + 2x + 3$, then $f^*(x) = 3x^4 + 2x^3 + x^2 + 5$. Self-reciprocal polynomials have a lot of interesting properties. The next proposition is one of them, the similarity with Proposition 3.1 is striking. We only prove the first part, the second part can by found in [29, Theorem 1].

Proposition 3.19. *Let p be an odd prime. Then each self-reciprocal irreducible monic polynomial $f \in \mathbb{F}_p[x]$ of degree $2m$ with $m \geq 1$ is a factor of*

$$H_{p,m}(x) := x^{p^m+1} - 1.$$

Furthermore, each irreducible factor of degree > 2 of $H_{p,m}(x)$ in $\mathbb{F}_p[x]$ is self-reciprocal of degree $2d$, where $d \mid m$ and m/d is odd.

Proof. First note that if α is a root of f, then so is $1/\alpha$, since

$$f(1/\alpha) = f(\alpha)\alpha^n = 0.$$

So, by Proposition 3.2, we know that $\alpha^{-1} = \alpha^{p^j}$ for some $0 \leq j \leq 2m-1$. Hence, $\alpha^{p^{2j}} = (\alpha^{-1})^{p^j} = \alpha$. We know that the minimal k such that $\alpha^{p^k} = \alpha$ is $k = 2m$, since f is irreducible. This implies that $2m \mid 2j$, hence $m \mid j$, which means that $j = 0$ or $j = m$. If $j = 0$, then $\alpha^2 = 1$ for all roots α of f, which means that $f(x) \mid x^2 - 1$. Now note that $x^2 - 1 \mid H_{p,m}(x)$, because $2 \mid p^m + 1$. Finally, if $j = m$, then $\alpha^{p^{m+1}} = 1$ for all roots α of f, hence $f(x) \mid H_{p,m}(x)$. □

Meyn [29] showed that you can construct irreducible polynomials over $\mathbb{F}_2$ using self-reciprocal ones. Cohen [13] generalized this method to finite fields of odd order. He defined an operator R such that given a polynomial f of degree m,

$$f^R(x) = (2x)^m f(1/2(x + 1/x)).$$

It can be shown that $f^R(x)$ is a self-reciprocal polynomial of degree $2m$. Cohen then proved the following theorem [13, Theorem 2].

Theorem 3.20. *Let p be an odd prime and let $f_0(x)$ be a monic irreducible polynomial mod p of degree $d \geq 1$, where d is even if $p = 3 \bmod 4$. Suppose that $\left(\frac{f_0(1)f_0(-1)}{p}\right) = -1$. For each integer $m \geq 1$ define*

$$f_m(x) = f_{m-1}^R(x).$$

Then for all $m \geq 0$, $f_m(x)$ is irreducible over $\mathbb{F}_p$ and has degree $d2^m$.

Example 3.21. If we take $p = 13$ and $f_0 = x + 3$, then $\left(\frac{f_0(1)f_0(-1)}{p}\right) = \left(\frac{8}{13}\right) = -1$. So Theorem 3.20 says that

$$x + 3, \quad x^2 + 6x + 1, \quad x^4 + 12x^3 + 6x^2 + 12x + 1,$$
$$x^8 + 11x^7 + 2x^6 + 12x^5 + 5x^4 + 12x^3 + 2x^2 + 11x + 1$$

are all irreducible mod 13.

Using Theorem 3.20, we will now show that there is another way to construct irreducible polynomials of degree $2^m \bmod p \mid n$.

Alternative proof of Theorem 3.16. We will again use that if n is prime, then we can compute square roots modulo n given some quadratic non-residue.
First suppose that $n = 1 \bmod 4$. First check that -1 has a square root mod n, so we know that all prime divisors of n are 1 mod 4. Now find an integer a with $\left(\frac{a^2-1}{n}\right) = -1$. Take $f_0(x) = x+a$, then Theorem 3.20 constructs polynomials of degree 2^m for all $m \geq 0$ that are irreducible modulo some $p \mid n$.
The case that $n = 3 \bmod 4$ is a bit harder, because the degree of $f_0(x)$ now has to be at least 2. Find an a with $\left(\frac{a}{n}\right) = -1$ and find a polynomial $f_0(x) = x^2 + bx + c$ with the following properties: $\left(\frac{\Delta(f_0)}{n}\right) = -1$ and $\left(\frac{f_0(1)f_0(-1)}{n}\right) = -1$. Cohen proves in Lemma 4 of his paper [13] that those polynomials exist. We know that n has a prime divisor p that is 3 mod 4. If we want to apply Theorem 3.20, then we need that all the equalities:

$$\left(\frac{-1}{p}\right) = \left(\frac{\Delta(f_0)}{p}\right) = \left(\frac{f_0(1)f_0(-1)}{p}\right) = -1$$

are true for that prime divisor p. Now, similarly as in the previous proof of this proposition, check that $-\Delta(f)$ and $-f_0(1)f_0(-1)$ are indeed both a square mod n using the square root algorithm. If not, then n is composite, otherwise we know that

$$\left(\frac{-1}{q}\right) = \left(\frac{\Delta(f)}{q}\right) = \left(\frac{f_0(1)f_0(-1)}{q}\right)$$

for all primes $q \mid n$. Hence, those symbol are all -1 for p. The claim then follows from Theorem 3.20. □

We found one more way to construct polynomials f which are irreducible modulo a divisor of n. This one very different from the others. The power of this method is that you can choose any degree you want. The big downside is that you can't always prove that the polynomial is irreducible. This method again only works in a compositeness/primality test setting, which is the setting we are most interested in. The main idea is that we can rule out certain factorizations of f modulo the prime divisors of n.

Proposition 3.22. *Given positive integers n and d with n odd, there is an algorithm that either proves that n is composite or returns a polynomial f of degree d such that if a prime p divides n, then f splits as a product of degree e irreducible polynomials over $\mathbb{F}_p$, where $e \mid d$. The number e can vary for the different prime divisors of n.*

Proof. If n is prime, then we can find irreducible polynomials f of any degree d in expected polynomial time in $\log(n)$ and d [45]. If that algorithm fails to produce a polynomial of degree d, then n is composite. Otherwise, we continue by computing $x^n, \ldots, x^{n^{d-1}} \bmod f$ and we check that

$$f(x^{n^i}) = 0 \bmod f \text{ for all } 1 \leq i \leq d-1 \text{ and } \gcd(x^{n^i} - x^{n^j}, f) = 1 \text{ for all } 0 \leq i < j \leq d-1.$$

If one of those equalities does not hold, then n is composite. Finally, we check that $\gcd(f, f') = 1$ to ensure that f is square-free modulo all divisors of n. Let p be a prime

divisor of n and suppose that f factors into irreducibles as $f_1 \cdots f_r$ over $\mathbb{F}_p$. Now take an arbitrary $f_i(x)$. Then $F = \mathbb{F}_p[x]/(f_i(x))$ is a field, so f has at most d roots in F. We found d of them, so f splits completely in F. This means that all $f_j(x)$ split completely in F, so $\mathbb{F}_{p^{\deg(f_j)}} \subset \mathbb{F}_{p^{\deg(f_i)}}$, hence we get that $\deg(f_j) \mid \deg(f_i)$ for all j. The above is true for all i, so $e = \deg(f_i) = \deg(f_j)$ for all i, j, which also implies that $e \mid \deg(f) = d$. This number e can vary for the different prime divisors of n. □

Corollary 3.23. *Given positive integers n and d with n odd and d prime, there is an algorithm that either proves that n is composite, or returns a polynomial f of degree d such that for every prime $p \mid n$ we have that f is irreducible or splits completely over $\mathbb{F}_p$. This can vary for the different prime divisors of n.*

Proof. This clearly follows from Proposition 3.22, since the only positive divisors of d are 1 and d. □

It would be great if we could rule out the case that f completely splits, because then we can prove that f is irreducible modulo $p \mid n$. We will sketch one possible way to do this. Suppose that we know a factor a of $n^{\deg f} - 1$ that is at least $\sqrt{n}$. Note that if n is composite, then it has a prime divisor $p \leq \sqrt{n}$. Try to find an element that has order at least a in $(\mathbb{F}_p[x]/(f))^*$. If f completely splits over $\mathbb{F}_p$, then such elements can't exist in $(\mathbb{F}_p[x]/(f))^*$, since then $(\mathbb{F}_p[x]/(f))^* = ((\mathbb{F}_p)^*)^{\deg(f)}$ and $\#\mathbb{F}_p^* = p - 1 < \sqrt{n} \leq a$. Which gives a contradiction, proving that f is irreducible mod $p \mid n$. If you could find an element of order at least n, then you can show that f is irreducible modulo all divisors of n, since all prime factors of n are $\leq n$. The big problem with this method is that you have to (partially) factor $n^{\deg f} - 1$, which is a very big integer, so that can be very hard.

This finishes our exploration of constructing polynomials that are irreducible modulo a divisor of n. We already saw that we can use these polynomials in the test that follows from Theorem 3.11. But, we also found another application that uses *elliptic curves*.

3.3 Elliptic curves

We will only give a very brief description of elliptic curves, a more detailed account can be found in [46]. Given a field F with $\mathrm{char}(F) \neq 2, 3$ and $a, b \in F$, an elliptic curve E is the set of solutions $(x, y) \in F^2$ of a *Weierstrass equation*:

$$y^2 = x^3 + ax + b, \tag{3.2}$$

together with a *point at infinity* denoted by $\mathcal{O}$. We also need that E is *non-singular*, this means that $\Delta_E := -16(4a^3 + 27b^2)$ has to be non-zero in F. The *j-invariant* is defined as $-1728(4a)^3/\Delta_E$. The set E can be turned into an abelian group $(E, +)$ with identity $\mathcal{O}$. For $m \in \mathbb{N}$ we write $[m]P = P + \cdots + P$, where P appears m times. Elliptic curves have been of great interest because of their applications in cryptography, but also in primality tests and integer factorization algorithms. Currently, the (heuristically) fastest primality proving algorithm is the Atkin-Morain ECPP test. A fast version [32]

of this algorithm has a heuristic running time of $\widetilde{\mathcal{O}}(\log(n)^4)$. We now give a brief sketch of that algorithm.

Given a prime n, you can consider elliptic curves defined over $\mathbb{Z}/n\mathbb{Z}$. But, if n is not prime, then $E(\mathbb{Z}/n\mathbb{Z})$ is not an elliptic curve by definition. However, we can pretend that that is not a problem and still do computations in E. It can be shown [31, § 5.4.2] that given $P, Q \in E(\mathbb{Z}/n\mathbb{Z})$, the addition algorithm either correctly computes $P + Q$ in $E(\mathbb{Z}/n\mathbb{Z})$, or it returns a factor of n. The latter case is certainly not a problem, because then we have proven that n is composite. Those "elliptic curves" are sometimes called pseudo-elliptic curves, or just elliptic curves for short.

A fundamental result of elliptic curves over finite fields is the Hasse bound.

Theorem 3.24. *Let E be an elliptic curve over a finite field of q elements. Then*

$$\big|\#E - (q+1)\big| \leq 2\sqrt{q}.$$

We can now prove the core result that Atkin and Morain use in their ECPP test. The result and proof are very similar to Proposition 2.29 and the method described below Corollary 3.23. We follow the proof of [31, Theorem 5.5.1].

Proposition 3.25. *Suppose we have an integer n and a point P on an elliptic curve E defined over $\mathbb{Z}/n\mathbb{Z}$ with the following properties. Suppose there are integers m, s such that $s > (n^{1/4}+1)^2$ and $s \mid m$ and $[m]P = \mathcal{O}$ and $[m/q_i]P \neq \mathcal{O}$ for all primes $q_i \mid s$. Then n is prime.*

Proof. Suppose that n is not prime and let p be a prime divisor of n with $p \leq \sqrt{n}$. Let E_p be the elliptic curve E viewed modulo p, also denote P_p by the point P reduced modulo p. This is possible, since if (3.2) holds modulo n, then it is also true modulo p. Now, $[s][m/s]P_p = \mathcal{O}$, so s is a multiple of the order of $[m/s]P_p \in E_p$. Furthermore, $[m/q_i]P_p \neq \mathcal{O}$ for all primes $q_i \mid s$, because otherwise the point addition algorithm would have returned the factor p of n. This means that $[m/s]P$ has order s in E_p. Hence $s \mid \#E_p$, so

$$\#E_p \geq s > (n^{1/4}+1)^2 \geq (\sqrt{p}+1)^2.$$

Now we use Theorem 3.24 to see that

$$\#E_p \leq p + 1 + 2\sqrt{p} = (\sqrt{p}+1)^2 < \#E_p,$$

which is a contradiction. Thus n is prime. □

We now check if the assumptions of Proposition 3.25 are reasonable. If n is prime, then we can find a point P by taking a random integer x and then computing a square root of $x^3 + ax + b \bmod n$. If this process fails, then n is composite. Finding the integer m is harder. The Goldwasser–Kilian ECPP test finds m using Schoof's algorithm [42] to compute the order of E. The running time of Schoof's algorithm is polynomial in $\log(n)$, but it is not very fast in practice. Another way to find m is by using *complex multiplication*. That method has as input a prime p and constructs an elliptic curve

E such that you know the order of E. It would take way to long to explain it all, so we refer to [31]. The complex multiplication method is often used in practice because it is much faster than Schoof's algorithm. Yet another way to find m is by restricting ourselves to *supersingular curves*, which we will consider later in this chapter.

Although we now know that we can compute the integer m, we still have to compute a large factor of m. This can be very hard in general, since $m \in \mathcal{O}(n)$. However, the power of elliptic curves is that we are not restricted to one group order. If we can't find a large factor of the order of one elliptic curve, then we can take another and try again.

We will now propose a compositeness test that combines the ECPP test of Atkin and Morain with Grantham's test. The idea is that elliptic curves can be defined over any finite field, so why not take an extension of a finite field. The bridge between Grantham's test and elliptic curves is the following map. Given an elliptic curve E over a finite field $\mathbb{F}_{p^d} = \mathbb{F}_p[x]/(f)$, the map

$$\begin{aligned} \sigma_p : E(\mathbb{F}_{p^d}) &\longrightarrow E(\mathbb{F}_{p^d}) \\ (g(x), h(x)) &\longmapsto (g(x)^p, h(x)^p) \\ \mathcal{O} &\longmapsto \mathcal{O} \end{aligned}$$

is called the *Frobenius endomorphism*. The freshman's dream implies that this map is well defined. The fact that it is an endomorphism that sends $\mathcal{O}$ to $\mathcal{O}$ tells us that it is a group homomorphism from $E(\mathbb{F}_{p^d})$ to itself. The fundamental property of the Frobenius endomorphism is that it satisfies the following characteristic equation:

$$\sigma_p^2 - [t_p]\sigma_p + [p] = [0], \tag{3.3}$$

where $t_p = p + 1 - \#E(\mathbb{F}_p)$ is called the *trace of Frobenius*. This means that for all $P = (g(x), h(x)) \in E(\mathbb{F}_{p^d})$, we have that

$$(g(x)^{p^2}, h(x)^{p^2}) - [t_p](g(x)^p, h(x)^p) + [p](g(x), h(x)) = \mathcal{O}.$$

This gives rise to the following compositeness test for an integer n. Use one of the results such as Theorem 3.16 to get a polynomial f that is irreducible modulo some $p \mid n$. Then check if (n, f) passes Grantham's test, if not, then n is composite. Construct an elliptic curve $E : y^2 = x^3 + ax + b$ with $\gcd(\Delta_E, n) = 1$ such that if n is prime, you know that the order of $E(\mathbb{Z}/n\mathbb{Z})$ is an integer m. We have already stated that this can be done using various techniques. Using this "guess" for $\#E$, we can compute $t_n = n + 1 - m$. Now pick a random point $P = (g(x), h(x))$ on $E((\mathbb{Z}/n\mathbb{Z})[x]/(f))$ by randomly choosing $g(x)$ and then computing a square root $h(x)$ of $g(x)^3 + ag(x) + b$ in $(\mathbb{Z}/n\mathbb{Z})[x]/(f)$. This can be done using a square root algorithm that works in general finite fields [18, Algorithm 14.15]. As always, if the process fails, then n is composite. After that, we check that the following equation holds

$$(g(x)^{n^2}, h(x)^{n^2}) - [t_n](g(x)^n, h(x)^n) + [n](g(x), h(x)) = \mathcal{O} \tag{3.4}$$

in $E((\mathbb{Z}/n\mathbb{Z})[x]/(f))$. If not, then n is composite.

The link with Grantham's test lies in the nth power map. Similarly what we did in Proposition 3.12, if we also check that $g(x^n) = g(x)^n \bmod (f, n)$ and $h(x^n) = h(x)^n \bmod (f, n)$, then since $x^n = x^{p^k} \bmod (f, p)$ for some k, we know that $\sigma_n(P) = \sigma_p^k(P) \bmod (f, p)$. So (3.4) implies that

$$\sigma_p^{2k}(P) - [t_n]\sigma_p^k(P) + [n]P = \mathcal{O} \tag{3.5}$$

in $E((\mathbb{Z}/p\mathbb{Z})[x]/(f))$. At the same time, we know that

$$\sigma_p^2(P) - [t_p]\sigma_p(P) + [p]P = \mathcal{O} \tag{3.6}$$

in $E((\mathbb{Z}/p\mathbb{Z})[x]/(f))$, where $t_p = p + 1 - \#E(\mathbb{F}_p)$. We would like to combine those two equations to get information about p for our compositeness test. However, this is quite difficult, since p and t_p are both unknown. Fortunately, in some cases we do know t_p, for example when the elliptic curve E is *supersingular*.

We say that $E(\mathbb{F}_p)$ is supersingular if $t_p = 0 \bmod p$. If $p > 3$, then Theorem 3.24 implies that $t_p = 0$. This will make it a lot easier to combine equations (3.5) and (3.6). We will see that there is a very nice link with what we did in Lemma 3.10. The next proposition helps us constructing supersingular elliptic curves. A proof can be found in [51, Proposition 4.37].

Proposition 3.26. *Let $p > 3$ be a prime. Then $E : y^2 = x^3 + 1$ defined over $\mathbb{F}_p$ is supersingular if and only if $p = 2 \bmod 3$ and $E : y^2 = x^3 + x$ defined over $\mathbb{F}_p$ is supersingular if and only if $p = 3 \bmod 4$.*

As usual, the above proposition is only true for prime numbers p. So, if we want to use it in a compositeness test for an integer n, we can try to reduce modulo some prime divisor of n.

Proposition 3.27. *Given positive integers n and m with $\gcd(n, 6) = 1$ and $n = 2 \bmod 3$ or $n = 3 \bmod 4$, there is an algorithm that either proves that n is composite, or returns an elliptic curve E and a polynomial f of degree 2^m such that there is a prime divisor $p \mid n$ such that $E(\mathbb{F}_p)$ is supersingular and f is irreducible over $\mathbb{F}_p$.*

Proof. We will show that the (first) proof of Theorem 3.16 for irreducible polynomials mod $p \mid n$ can be adapted such that we also get a supersingular elliptic curve modulo the same prime p.
First suppose that $n = 3 \bmod 4$. In the proof of Theorem 3.16, we saw that n in that case has a prime divisor p which is also $3 \bmod 4$ and the polynomial f of degree 2^m we constructed was irreducible over $\mathbb{F}_p$. Take the elliptic curve $E : y^2 = x^3 + x$ from Proposition 3.26, then $E(\mathbb{F}_p)$ is supersingular.
Now suppose that $n = 2 \bmod 3$, then n has a prime divisor p that is also $2 \bmod 3$. We now take $E : y^2 = x^3 + x$, then $E(\mathbb{F}_p)$ is supersingular. We don't know the value of

$$\left(\frac{3}{n}\right) = (-1)^{\frac{n-1}{2}}\left(\frac{n}{3}\right) = (-1)^{\frac{n+1}{2}}$$

yet, since that depends on $n \bmod 4$. So, first suppose that $n = 1 \bmod 4$. Then $\left(\frac{3}{n}\right) = -1$. Now check that $-1 \bmod n$ is indeed a square, then all primes dividing n are $1 \bmod 4$. Hence $\left(\frac{3}{p}\right) = -1$, so we can use Proposition 3.14 with $a = 3$ to get a polynomial of degree 2^m that is irreducible over $\mathbb{F}_p$. Finally, suppose that $n = 3 \bmod 4$. This time $\left(\frac{3}{n}\right) = 1$, and we can check that it is indeed a square mod n. This implies that $\left(\frac{3}{p}\right) = 1$, so $p = 3 \bmod 4$, where p is the prime divisor of n which is $2 \bmod 3$. Find an integer a with $\left(\frac{a^2+4}{n}\right) = -1$. Then by computing a square root of $-(a^2+4) \bmod n$ like before, we can show that $\left(\frac{a^2+4}{p}\right) = -1$. Hence, we get our desired polynomial from Proposition 3.15. □

We can now use Proposition 3.27 in the compositeness test described above Proposition 3.26. Suppose that $n = 3 \bmod 4$ or $n = 2 \bmod 3$, then we get an elliptic curve E that is supersingular for some prime $p \mid n$. If we check that $\gcd(6, n) = 1$, then $p > 3$, so $t_p = 0$. We also take $t_n = 0$, because if n is prime, then $E(\mathbb{F}_n)$ is also supersingular. Now, σ is a group homomorphism, so equation (3.6) implies that

$$\sigma_p^{2k}(P) = [-p]\sigma_p^{2k-2}(P) = [p^2]\sigma_p^{2k-4}(P) = \cdots = [(-p)^k]P.$$

Combining with equation (3.5), we get that

$$[-n]P = [(-p)^k]P$$

in $E((\mathbb{Z}/p\mathbb{Z})[x]/(f))$. Now, the polynomial f from Proposition 3.27 has degree $d = 2^m$ and we expect that all of $x, x^p, \ldots, x^{p^{2^m-1}}$ are different roots of f. If that's not the case, then n is composite, else we know that $\gcd(k, 2^m) = 1$, which means that k is odd. We can check this by computing $\gcd(x^{p^i} - x^{p^j}, f)$ for all $0 \leq i < j \leq d-1$. Let a be the order of $P \in E(\mathbb{F}_{p^d})$, then $n = p^k \bmod a$. Which is the exact same relation as the one we got below Lemma 3.10. The only possible difference is that the modulus a might be different.

However, a divides the order of $E(\mathbb{F}_{p^d})$ and since $E(\mathbb{F}_p)$ is supersingular, there is an explicit formula for the order of that group [46, Exercise 5.15]

$$\#E(\mathbb{F}_{p^d}) = \begin{cases} p^d + 1 & \text{if } d \text{ is odd;} \\ (p^{d/2} - (-1)^{d/2})^2 & \text{if } d \text{ is even.} \end{cases}$$

Now note that $p^d + 1 \mid p^{2d} - 1 = \#\mathbb{F}_{p^{2d}}^*$ and $p^{d/2} - (-1)^{d/2} \mid p^d - 1 = \#\mathbb{F}_{p^d}^*$. Which means that if a point $P \in E(\mathbb{F}_{p^d})$ has order a, where $E(\mathbb{F}_p)$ is supersingular, then there is also an element in $\mathbb{F}_{p^{2d}}^*$ of order a. So, Grantham's test on supersingular elliptic curves reduces to the usual Grantham's test in finite fields.

This means that using supersingular curves in our compositeness test is not very useful, because it is not better than just working in finite fields. However, the strength of elliptic curves is precisely that you are not restricted to a single group. The downside of using an elliptic curve with $t_p \neq 0$ is that you can't combine equations (3.5) and (3.6) very easily to get an explicit result. But, using our elliptic compositeness test with those

elliptic curves is still a strong test, since it is at least as strong as Grantham's test, and probably even stronger.

It should be noted that using supersingular elliptic curves in compositeness tests has been done before. Gordon [20] introduced the so-called *elliptic pseudoprimes*. Given an integer n and an elliptic curve E defined modulo n such that E is supersingular if n is prime, the integer n is an elliptic pseudoprime for (E, P) if $P \in E$ and $[n+1]P = \mathcal{O}$. We can [47] of course extend this by defining n to be an elliptic Carmichael number for E if n is an elliptic pseudoprime for (E, P) for all $P \in E$. Examples for E can be found in Proposition 3.26, but these are not the only ones, as we will see in the next paragraph.

3.4 Non-residue modulo all divisors of n

The goal of this paragraph is to prove the following new result.

Theorem 3.28. *Given an integer n, there is an algorithm that either shows that n is composite or returns an integer a with*

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) = -1 \quad \textit{for all primes } p \mid n.$$

If the generalized Riemann hypothesis (GRH) is true, then the algorithm has expected running time $\widetilde{\mathcal{O}}(\log(n)^3)$.

We stress that the GRH is only needed for proving the running time, not the correctness of the algorithm. We will first look at the two main corollaries of this theorem. The first one is an improvement of Theorem 3.16.

Corollary 3.29. *Given positive integers n and m with n odd, there is an algorithm that either proves that n is composite, or returns a polynomial of degree 2^m that is irreducible modulo all prime divisors of n.*

Proof. Suppose that we get an integer a from the algorithm of Theorem 3.28. We will adapt the first proof of Theorem 3.16 using this a
First suppose that $n = 1 \bmod 4$. We can follow the same steps as in the first proof of Theorem 3.16, but we don't have to find a non-residue modulo n, we use the integer a instead. Then Proposition 3.14 gives a polynomial of degree 2^m that is irreducible modulo all $p \mid n$.
Now suppose that $n = 3 \bmod 4$. This case is slightly more work, since now we need an integer b with $\left(\frac{b^2+4}{n}\right) = -1$, so we can't use a directly. But, if we find such b, then we can try to compute the square root of $a(b^2+4) \bmod n$. If it exists, then

$$\left(\frac{b^2+4}{p}\right) = \left(\frac{a}{p}\right) = -1 \quad \text{for all } p \mid n.$$

Then by following the rest of the first proof of Theorem 3.16, we get a polynomial of degree 2^m that is irreducible modulo all $p \mid n$. □

This gives another big improvement for the test described in Theorem 3.11. Because it makes it easier to satisfy the assumption that there is a prime divisor p of n such that we have an element of large order.

The second corollary is also interesting since it can strengthen a lot of existing compositeness and primality tests.

Corollary 3.30. *Given an integer n, there is an algorithm that either shows that*

a) n is composite, or

b) n is prime or has at least 3 (not necessarily distinct) prime divisors.

If the generalized Riemann hypothesis (GRH) is true, then the algorithm has expected running time $\widetilde{\mathcal{O}}(\log(n)^3)$.

Proof. Suppose that we get an integer a from the algorithm of Theorem 3.28. We will prove a slightly stronger result, namely that n has an odd number of prime divisors. Because if n has $2m$ prime divisors, then

$$-1 = \left(\frac{a}{n}\right) = (-1)^{2m} = 1.$$

So, if n is composite, then n has at least 3 prime divisors. □

We could try to use Corollary 3.30 to improve compositeness tests like the Miller-Rabin test. It might be possible to improve the probability $1/4$ from that test to $1/8$ using the fact that n has 3 prime factors. But, the complexity of one run of a Miller-Rabin test is $\widetilde{\mathcal{O}}(\log(n)^2)$. So combining them is probably not worth it, because we can do $\widetilde{\mathcal{O}}(\log(n))$ runs of Miller-Rabin in the time that we run the algorithm from Corollary 3.30, which is more than enough.

But, the combination with the ECPP test might be more promising. Because then we can weaken the assumption in Proposition 3.25 that we need an integer s such that $s > (n^{1/4}+1)^2$ to having an integer s such that $s > (n^{1/6}+1)^2$. This is an improvement of a factor $\mathcal{O}(n^{1/6})$, which is significant, since factoring the integer m in Proposition 3.25 is generally very hard. Furthermore, one round of fast ECPP takes $\widetilde{\mathcal{O}}(\log(n)^3)$ [32], so Corollary 3.30 won't worsen the complexity. However, ECPP is usually implemented such that it tries to find integers m that are of the form $a \cdot q$, where a is a small integer and q a large probable prime, so that we can take $s = q$. It then proceeds recursively by proving that q is prime with a different elliptic curve modulo q. With that in mind, it doesn't really matter if the bound on s is weakened, since s is already big enough anyways.

Instead, the combination of Corollary 3.30 with ECPP is better when we try we search for an m that can be easily factored as $b \cdot c$, where b consists of small prime factors and $b > (n^{1/6}+1)^2$. Because then we could prove the primality of n in one round of ECPP, instead of the recursive process. But, this would need a more detailed analysis to see which version is faster, since integers with such a large smooth factor might be a lot scarcer than the ones of the form $a \cdot q$ from before.

At first we tried to prove Theorem 3.28 using properties of finite fields such as Proposition 3.2. Using those properties we could produce an integer a that was very likely to be a non square modulo all divisors, but we could not find a proof. Fortunately, we did eventually succeed using a very different approach. Coincidentally, this method uses supersingular elliptic curves again. Our algorithm uses two existing algorithms. The first one constructs a supersingular curve and the second one checks if an elliptic curve is supersingular or not. Using those two algorithms, we will produce an elliptic curve E that is supersingular modulo all divisors of n. In Proposition 3.26 we already saw that $E : y^2 = x^3 + x$ is supersingular over $\mathbb{F}_p$ if and only if $\left(\frac{-1}{p}\right) = -1$. Continuing this idea, we will construct elliptic curves E and certain integers a such that $E(\mathbb{F}_p)$ is supersingular if and only if $\left(\frac{a}{p}\right) = -1$.

Unfortunately, we won't explain the two algorithms in much detail, this is because the theory behind it would take a lot of work to set up. However, we will explain how the correctness of our algorithm follows from the other two.

The first algorithm is by Bröker [9, Algorithm 2.4] and can be seen as a generalization of Proposition 3.26. Recall that for an elliptic curve $E : y^2 = x^3 + ax + b$, the j-invariant $j(E)$ is defined as $j(E) = -1728(4a)^3/\Delta_E$. It can be shown that two elliptic curves over an algebraically closed field K are isomorphic if and only if they have the same j-invariant [46, Proposition 1.4b]. The algorithm first finds a j which is a j-invariant of a supersingular elliptic curve. It accomplishes this by computing a root of the *Hilbert class polynomial* $P_K(x) \in \mathbb{Z}[x]$ of a certain quadratic field K. After that it constructs an elliptic curve E with j-invariant j. The input of the algorithm is an odd prime p and the output is a supersingular curve E over $\mathbb{F}_p$.

Algorithm 3 Constructing supersingular curves

```
1:  function ConstructSupersingular(p)
2:      if p = 3 mod 4 then
3:          return E : y^2 = x^3 + x
4:      end if
5:      find the smallest prime q such that q = 3 mod 4 and (-q/p) = -1
6:      if q = 3 then
7:          return E : y^2 = x^3 + 1
8:      end if
9:      let K = Q(sqrt(-q)) and compute P_K(x)
10:     reduce P_K(x) mod p and compute a root j ∈ F_p
11:     compute a = 27j/(4(1728-j)) mod p
12:     return E : y^2 = x^3 + ax - a
13: end function
```

Bröker proves that his algorithm always returns a supersingular curve and that the expected running time is $\widetilde{\mathcal{O}}(\log(p)^3)$, assuming the generalized Riemann hypothesis (GRH). The GRH is only used for the bound on the running time, not for the correctness. In the proof he shows that the elliptic curve in line 12 is supersingular if

and only if the prime q satisfies $\left(\frac{-q}{p}\right) = -1$. Note that by Proposition 3.26, the elliptic curve in lines 3 and 7 are supersingular if and only if $\left(\frac{-1}{p}\right) = -1$ and $\left(\frac{-3}{p}\right) = -1$ respectively. We thus see that every elliptic curve returned by Algorithm 3 can be associated with an integer a such that $\left(\frac{a}{p}\right) = -1$.

We can use Algorithm 3 in a compositeness test as follows. If we run the algorithm on an integer n, then either a step fails, showing that n is composite, or it returns an elliptic curve E. We don't know yet if E is supersingular modulo prime divisors of n, because Algorithm 3 is only correct for prime numbers. To circumvent this problem, we use the following algorithm.

The second algorithm is by Sutherland [48, Algorithm 2]. It takes as input an elliptic curve E and a prime $p > 3$ and returns true if $E(\mathbb{F}_p)$ is supersingular and false otherwise. Sutherland proved that the expected running time of his algorithm is $\widetilde{\mathcal{O}}(\log(p)^3)$.

Algorithm 4 Identifying supersingular curves

```
1: function CheckSupersingular(E, p)
2:     try to factor Φ_2(x, j(E)) as (x − j_1)(x − j_2)(x − j_3) in F_{p^2}
3:     if Φ_2(x, j(E)) can't be factored like that in F_{p^2} then
4:         return false
5:     end if
6:     set j'_i = j(E) for i = 1, 2, 3
7:     compute m = ⌊log_2(p)⌋ + 1
8:     for k = 1, ..., m do
9:         for i = 1, 2, 3 do
10:            compute f_i(x) = Φ_2(x, j_i)/(x − j'_i) ∈ F_{p^2}[x]
11:            update j'_i = j_i
12:            try to find a root r_i of f_i(x) in F_{p^2}
13:            if f_i(x) does not have a root in F_{p^2} then
14:                return false
15:            end if
16:            update j_i = r_i.
17:        end for
18:    end for
19:    return true
20: end function
```

The algorithm uses the so-called *modular polynomials* $\Phi_l(x, y) \in \mathbb{Z}[x, y]$ for positive integers l. It can be shown that [48] if j_1 and j_2 are roots of a $\Phi_l(x, y) \bmod p$, then there are elliptic curves E_1, E_2 over $\mathbb{F}_p$ such that $j_i = j(E_i)$ for $i = 1, 2$ and $\#E_1(\mathbb{F}_p) = \#E_2(\mathbb{F}_p)$. So, $E_1(\mathbb{F}_p)$ is supersingular if and only if $E_2(\mathbb{F}_p)$ is supersingular. The algorithm tries to find a "long" chain of roots of $\Phi_2(x, y)$, starting with $x = j(E)$, where

E is the elliptic curve of the input and

$$\begin{aligned}\Phi_2(x,y) = {} & x^3+y^3-x^2y^2+1488(x^2y+xy^2)-162000(x^2+y^2)\\ & +40773375xy+8748000000(x+y)-157464000000000.\end{aligned}$$

Note that this polynomial is symmetric. Sutherland showed that such a long chain exists if and only if $E(\mathbb{F}_p)$ is supersingular. He also showed that his algorithm finds the longest chain, hence we can prove or disprove that E is supersingular.

This algorithm is again only correct for prime numbers p, but we will now show that we can adapt both Algorithm 3 and 4 to be useful in a compositeness test setting.

We first consider Algorithm 3. We have to slightly change this algorithm to "work" for general integers n. After line 7, we also check that -1 really is a square mod n. If that process fails, then n is composite. We do this to ensure that all prime divisors of n are 1 mod 4. Furthermore, instead of just returning the elliptic curve, we also return the corresponding integer a such that $\left(\frac{a}{n}\right) = -1$. So, in line 3 of Algorithm 3 we have $a = -1$, in line 7 we have $a = -3$ and in line 12 we have $a = -q$. We call this algorithm the *adapted Algorithm* 3.

Now we consider Algorithm 4. We also have to alter this algorithm a little bit, since it uses finite fields. We can't create "$\mathbb{F}_{n^2}$", because we don't know if n is prime or not. However, we can create $\mathbb{Z}/n\mathbb{Z}[x]/(g(x))$, where g has degree 2 and $\left(\frac{\Delta(g)}{n}\right) = -1$, and try to find roots of $\Phi_2(x,y)$ in that ring. If the algorithm fails, then we know that n is composite, else we continue with the algorithm. We call this algorithm the *adapted Algorithm* 4.

We can now state the algorithm where Theorem 3.28 refers to. The input is an integer $n > 3$, the algorithm either proves that n is composite or returns an integer a such that $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) = -1$ for all primes $p \mid n$.

Algorithm 5 Non-residue modulo all divisors

```
1:  function FindGlobalNonresidue(n)
2:      if gcd(n, 6) > 1 then
3:          return n is composite
4:      end if
5:      run the adapted Algorithm 3 with input n and let (E, a) be the output
6:      if the process fails then
7:          return n is composite
8:      end if
9:      run the adapted Algorithm 4 with input (E, n) and let b be the output
10:     if the process fails or b = false then
11:         return n is composite
12:     end if
13:     return a
14: end function
```

Proof of Theorem 3.28. We claim that Algorithm 5 has the properties stated in the theorem.
Since both Algorithm 3 and 4 are correct when when the input is prime, we know that if one of those algorithms fails, then n is composite. So, assume that neither Algorithm fails. If $b =$ false, then we also know that n is composite. Since if n is prime, then Algorithm 3 produces a supersingular curve over $\mathbb{F}_n$, so Algorithm 4 would in that case return true.
Finally, we have the case that Algorithm 5 reaches line 10, where it returns an integer a. In this case, we know that all of the roots j_i and j_i' needed in the adapted Algorithm 4 have been found in the ring $R = \mathbb{Z}/n\mathbb{Z}[x]/(g(x))$. Let p be a prime divisor of n. If we reduce R modulo p, then we either get $\mathbb{F}_{p^2}$ or $\mathbb{F}_p \times \mathbb{F}_p$. In both cases, we see that the roots j_i and j_i' also lie in $\mathbb{F}_{p^2}$. Then since Algorithm 4 is correct for primes, we know that $E(\mathbb{F}_p)$ is supersingular.
Now, in the paragraph below Algorithm 3, we noted that Bröker proved that the integer a that the adapted Algorithm 3 returns satisfies $\left(\frac{a}{p}\right) = -1$ if and only if $E(\mathbb{F}_p)$ is supersingular. Thus we know that $\left(\frac{a}{p}\right) = -1$. Then since p was an arbitrary prime divisor of n, we know that $\left(\frac{a}{q}\right) = -1$ for all prime divisors q of n.
Now we look at its running time. Algorithm 3 and 4 both have expected running time $\widetilde{\mathcal{O}}(\log(p)^3)$ for a prime p, assuming GRH. In the adapted versions we fill in an integer n which might be composite, but then it either aborts earlier or finishes as quickly. We only added a square root computation in Algorithm 3. We can compute square roots mod n using the Tonelli-Shanks algorithm [1] in $\widetilde{\mathcal{O}}(\log(n)^2)$, or the algorithm fails, proving that n is composite. Hence the total expected running time of Algorithm 5 is $\widetilde{\mathcal{O}}(\log(n)^3)$, assuming GRH. □

Bibliography

[1] Adleman, L., Manders, K. and Miller, G. *On Taking Roots in Finite Fields*, in Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (1977), 175–177.

[2] Agrawal, M., Kayal, N. and Saxena, S. *PRIMES is in $\mathcal{P}$*, Annals of Mathematics, **160** (2004), 781–793.

[3] Alford, W.R., Granville, A. and Pomerance, C. *There are infinitely many Carmichael numbers*, Ann. Math. **140** (1940), 703–722.

[4] Andrén, D. *On the Complexity of Matrix Reduction over Finite Fields*, Adv. Appl. Math. **39.4** (2007), 428–452.

[5] Bach, E. *Explicit Bounds for Primality Testing and Related Problems*, Math. Comp. **55** (1990), 355–380.

[6] Bernstein, D. J. *Fast multiplication and its applications*, in Buhler, J. P. and Stevenhagen, P. (Eds.), *Algorithmic Number Theory* vol. 44, Cambridge Univ. Press, Cambridge, 2008, 325–384.

[7] Bernstein, D. J. *Detecting perfect powers in essentially linear time*, Mathematics of Computation **67** (1998), 1253–1283.

[8] Brand, L. *The Companion matrix and its Properties*, Amer. Math. Monthly **71.6** (1964), 629–634.

[9] Bröker, R. *Constructing Supersingular Elliptic Curves*, J. Comb. Number Theory **1.3** (2009), 269–273.

[10] Bruce, J. W. *A Really Trivial Proof of the Lucas-Lehmer Test*, American Mathematical Monthly, **100.4** (1993), 370–371.

[11] Buchmann, J. and Shoup, V. *Constructing Nonresidues in Finite Fields and the Extended Riemann Hypothesis*, Math. Comp. **65** (1996), 1311–1326.

[12] Celler, F. and Leedham-Green, C.R. *Calculating the order of an invertible matrix*, Groups and Computation II, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **28**, AMS (1997), 55–60.

[13] Cohen, S. D. *The Explicit Construction of Irreducible Polynomials Over Finite Fields*, Des. Codes Cryptogr. **2** (1992), 169–174.

[14] Conrad, K. *The Miller-Rabin test*, accessed on 1 July 2020, `https://kconrad.math.uconn.edu/blurbs/ugradnumthy/millerrabin.pdf`.

[15] Conrad, K. *The minimal polynomial and some applications*, accessed on 1 July 2020, `https://kconrad.math.uconn.edu/blurbs/linmultialg/minpolyandappns.pdf`.

[16] Damgård, I. B. and Frandsen, G. S. *An Extended Quadratic Frobenius Primality Test with Average and Worst Case Error Estimates*, Journal of Cryptology **19.4** (2006), 489–520.

[17] Gall, F. *Powers of tensors and fast matrix multiplication*, Proc. 39th Int. Symp. Symbolic Algebraic Comput. (2014), 296–303.

[18] Gathen, J. and Gerhard, J. *Modern Computer Algebra* (3rd ed.), Cambridge Univ. Press, Cambridge, 2013.

[19] GIMPS, *GIMPS Discovers Largest Known Prime Number:* $2^{82,589,933} - 1$, accessed on 1 July 2020, `https://www.mersenne.org/primes/?press=M82589933`.

[20] Gordon, D. M. *On the Number of Elliptic Pseudoprimes*, Math. Comp. **52** (1989), 231–245.

[21] Grantham, J. *Frobenius Pseudoprimes*, Math. Comp. **70** (2001), 873–891.

[22] Grantham, J. *A Probable Prime Test with High Confidence*, J. Number Theory **72.1** (1998), 32–47.

[23] Hodges, J. H. *The matrix equation* $X^2 - I = 0$ *over a finite field*, Amer. Math. Monthly **65.7** (1958), 518–520.

[24] Keller-Gehrig, W. *Fast algorithms for the characteristic polynomial*, Theor. Comp. Sci. **36** (1985), 309–317.

[25] Lang, S. *Undergraduate Algebra* (3rd ed.), Springer, 2005.

[26] Lenstra, H. W. and Pomerance, C. *Primality testing with Gaussian periods*, J. Eur. Math. Soc. **21.4** (2019), 1229–1269.

[27] Lidl, R. and Niederreiter, H. *Finite fields* (2nd ed.), Encyclopedia of Mathematics and its Applications **20**, Cambridge University Press, Cambridge, 1997.

[28] Maxfield, M. W. *The order of a matrix under multiplication (modulo m)*, Duke Math. J. **18.3** (1951), 619–621.

[29] Meyn, H. *On the Construction of Irreducible Self-Reciprocal Polynomials Over Finite Fields*, Applicable Algebra in Eng., Comm. and Comp. **1** (1990), 43–53.

[30] Miller, J. C. P. and Spencer Brown, D. J. *An algorithm for evaluation of remote terms in a linear recurrence sequence*, Comput. J. **9.2** (1966), 188–190.

[31] Morain, F. *Implementation of the Atkin-Goldwasser-Kiliann primality testing algorithm*, Research Report 911, INRIA (1988).

[32] Morain, F. *Implementing the Asymptotically Fast Version of the Elliptic Curve Primality Proving Algorithm*, Math. Comp. **76** (2007), 493–505.

[33] Müller, S. *A Probable Prime Test with Very High Confidence for* $n \equiv 1 \bmod 4$, in Advances in Cryptology—ASIACRYPT 2001 (Gold Coast), volume 2248 of Lecture Notes in Computer Science, pages 87–106. Springer, Berlin, 2001.

[34] Müller, S. *A Probable Prime Test with Very High Confidence for* $n \equiv 3 \bmod 4$, Journal of Cryptology **16.2** (2003), 117–139.

[35] Munafo, R. *Notable Properties of Specific Numbers*, accessed on 4 July 2020, `http://mrob.com/pub/math/numbers-19.html`

[36] Niven, A. *Fermat's theorem for matrices*, Duke Math. J. **15.3** (1948), 823–826.

[37] Rabin, M. O. *Probabilistic Algorithm for Testing Primality*, J. Number Theory **12** (1980), 128–138.

[38] Rödseth, Ö. J. *A note on primality tests for* $N = h \cdot 2^n - 1$, BIT **34** (1994), 451–454.

[39] Rosser, J. B. and Schoenfield, L. *Approximate formulas for some functions of prime numbers*, Ill. J. Math. **6** (1962), 64–94.

[40] Schönhage, A. *Schnelle Berechnung von Kettenbruchentwicklungen*, Acta Informatica **1** (1971), 139–144 (in German).

[41] Schönhage, A. and Strassen, V. *Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292 (in German).

[42] Schoof. R. *Elliptic Curves over Finite Fields and the Computation of Square Roots mod p*, Math. Comp. **44** (1985), 483–494.

[43] Seysen, M. *A Simplified Quadratic Frobenius Primality Test*, Cryptology ePrint Archive, Report 2005/462, `https://eprint.iacr.org/2005/462.pdf`.

[44] Shoup, V. *Searching for Primitive Roots in Finite Fields*, Math. Comp. **58** (1992), 369–380.

[45] Shoup, V. *Fast Construction of Irreducible Polynomials over Finite Fields*, J. Symb. Comput. **17.5** (1994), 371–391.

[46] Silverman, J. H. *The Arithmetic of Elliptic Curves* (2nd ed.), Springer, 2009.

[47] Silverman, J. H. *Elliptic Carmichael numbers and elliptic Korselt criteria*, arXiv:1108.3830.

[48] Sutherland, A. V. *Identifying Supersingular Elliptic Curves*, LMS J. Comp. and Math. **15** (2012), 317–325.

[49] Swan, R. G. *Factorization of polynomials over finite fields*, Pacific J. Math. **12** (1962), 1099–1106.

[50] Vaskouski, M., Kondratyonokb, N. and Prochorovb, N. *Primes in quadratic unique factorization domains*, J. Number Theory **168** (2016), 101–116.

[51] Washington, L. C. *Elliptic Curves* (2nd ed.), Chapman & Hall/CRC, Boca Raton, 2008.