Irreducibility Certificates for Polynomials with Integer Coefficients

Bart Kirkels

August 2004

Voorwoord

Dit is 'm dan: mijn doctoraalscriptie. Na vijf jaar wiskunde gestudeerd te hebben, ben ik het laatste half jaar met onderzoek bezig geweest en de resultaten daarvan zijn in dit verslag te lezen.
Deze scriptie kwam er natuurlijk met de hulp van velen, die ik nu graag wil bedanken.
In de eerste plaats mijn afstudeerbegeleider Wieb Bosma. Hij heeft het onderzoek in goede banen geleid, en maakte altijd tijd om er over te praten, ook al had hij het vaak erg druk. Bedankt voor de fijne samenwerking Wieb!
Ook wil ik Bas Spitters, de tweede lezer, bedanken voor het herhaaldelijk doorlezen en becommentariëren van dit stuk, ook al zat hij midden in een verhuizing.
De officieuze derde lezer is Roel Willems. Hem wil ik bedanken voor het lezen en corrigeren van deze scriptie en het prettige samenwerken bij (onder andere) Commutatieve Algebra, Recursietheorie en het implementeren van de MPQS.
Tijdens de studie heb ik ook (meer dan) voldoende afleiding en koffie gehad, waarvoor ik Desda graag bedank. Ook alle studenten, met name iedereen uit mijn jaar: bedankt voor de supergezellige tijd!
Ik heb de afgelopen vijf jaar zo veel prachtigs geleerd dat ik daarvoor alle docenten en de rest van de staf wil bedanken. Ook iedereen waar ik mee samengewerkt heb in commissies en besturen wil ik bedanken voor de goede samenwerking.
Ook nog veel dank aan mijn familie en vrienden die me gesteund hebben en (in hele positieve zin) afgeleid hebben naast de studie. Voor hun is er achter in deze scriptie een Nederlandse, niet-wiskundige, samenvatting.
En ten slotte wil ik Sanne bedanken, voor al het fijne dat we tot nu toe al meegemaakt hebben, en ook voor het nakijken van de samenvatting natuurlijk.

Veel plezier met het lezen van deze scriptie!

Bart Kirkels
Nijmegen, augustus 2004

Contents

Chapter 1

Introduction

Let f be an integer polynomial. We are interested in its factorisation. So we ask a computer algebra system what it is. If f is reducible then we will get its factors and can easily check the multiplication. But what if the answer is that our polynomial is irreducible? How can the system convince us? Well, usually it really does not, but there are certain criteria that can guarantee the irreducibility of polynomials. Irreducibility itself is a negative property in the following sense: there *do not exist* non-trivial factors. The criteria need a positive property, such as the existence of a certain number. When we know this number we only have to check whether it is really correct or not, and have proved irreducibility in that way. This proving can be done on a computer, using a so called Proof Assistant.

In this thesis some of these criteria will be investigated. I will describe an algorithm that assigns an irreducibility certificate (based on two criteria) to every irreducible polynomial in $\mathbb{Z}[X]$. I will show that these certificates are almost always easy to check. I have also proved the correctness of a criterion in the Proof Assistent Coq.

To achieve this the notions of an irreducibility certificate and of an irreducibility criterion will be made more precise in Chapter 2. Some certificates, that do not suffice for our purposes, are then given as examples.

In Chapter 3 some practical certificates are given and for every certificate it is proved that its existence for a polynomial $f \in \mathbb{Z}[X]$ guarantees irreducibility of f. One of these certificates is the 'modulo p-certificate'. It is based on factorizing f modulo primes.

In Chapter 4 this certificate is investigated using Galois theory. It is proved that for a polynomial f the existence of a modulo p-certificate is equivalent to having a special condition on the Galois group of f. Now Galois theory is used to prove that for polynomials of prime degree there always exists a certificate. For polynomials of composite degree it is proved that for almost every such f there exists a certificate. We also see that these certificates are easy to check.

In Chapter 5 we give an algorithm that assigns a certificate to every irreducible polynomial. I have implemented this algorithm in the Computer Algebra System Magma. This algorithm tries to assign a modulo p-certificate, using the factorisations of the polynomial f modulo the primes below some bound (based on the degree of f). If a modulo p-certificate has not been found, we can assume that it does not exist and we now use the modulo-factorisations of f to find a ‘Hensel-certificate’. Such a certificate exists for every irreducible polynomial (but can be hard to check). An other advantage of the Hensel-part of our algorithm is that it will find a non-trivial factor of f if f is reducible.

I have also implemented algorithms that assign the other certificates of Chapter 3 to irreducible polynomials (when possible). All of these algorithms are tested in Chapter 6.

In Chapter 7 the subject of formalizing mathematics on a computer is introduced. Programs that can be used for formalizing are called Proof Assistants. Coq is the Proof Assistant that I have used. Examples of using Coq (in the setting of this thesis) are given. I have formalized an irreducibility criterion in Coq; I will sketch how this was done. The formalizing process can be combined with the computations of Magma. This combination can be used to formally prove irreducibility of polynomials.

In Computer Algebra Systems the factorisation of a polynomial is usually determined by using the LLL-algorithm [20]. The mathematical theory for proving the correctness of this algorithm is at this moment too advanced to formalize in Coq. That is the reason why we want to be able to prove irreducibility in another way.

Eventually, in Chapter 8, I give a short non-mathematical summary of this thesis for everyone who is interested in what I have done.

Chapter 2

Certificates: Definitions and Examples

In this chapter we shall first explain the notion of an irreducibility certificate. Then we take a look at two specific certificates. The first one seems to be quite good, but generating these certificates is very time-consuming. The second one is an example of a conditional certificate, and we shall see how it depends on a conjecture.

By Gauss' lemma we can prove the irreducibility of a polynomial over $\mathbb{Q}$, by proving the irreducibility of a corresponding polynomial over $\mathbb{Z}$. That is why we will give irreducibility certificates for polynomials over $\mathbb{Z}$. So in this chapter f will be a polynomial in $\mathbb{Z}[X]$.

2.1 Definition

Definition 2.1 *A* criterion *is an existential condition; by this we mean that it is of the form: 'there exist mathematical objects M such that ... '.*

Definition 2.2 *More specifically: an* irreducibility criterion *is an existential condition for a polynomial f that guarantees the irreducibility of f.*

A famous example is Eisenstein's criterion which will be part of our investigation in the next chapter.

Definition 2.3 *An* irreducibility certificate *for f and an irreducibility criterion, consists of mathematical objects M that satisfy the criterion for f.*

Definition 2.4 *A* checking-algorithm *for a criterion is a procedure to decide whether a certificate M satisfies the criterion or not.*

The main purpose of this thesis is to give certificates together with a correctness proof. This can be achieved by formalizing the criterion and the checking-algorithm in a so called 'proof assistant'. Chapter 7 will be about formalizing, but for now it is enough to know that the proof of the sufficiency of the criterion should not be too advanced, so that it can be formalized.

Of course we would also like a checking-algorithm to be fast. It would be nice if it would require an amount of computation polynomial in the size of f. (By the *size* of f we mean the number of bits required to represent all coefficients of f.) But we shall also see certificates of which we can not guarantee an easy check, but that are on average quite fast to check.

There is also another aspect, that of *finding* certificates. We hope to find certificates for every irreducible polynomial, but for most criteria there are irreducible polynomials that do not have a certificate, or for which it takes a lot of time to find a certificate.

2.2 Cantor's Certificate

In 1981 David Cantor [6] showed that for every irreducible polynomial there exists an irreducibility certificate that can be checked in time polynomial in the size of f.

First some definitions:

Definition 2.5 *Let $f \in \mathbb{Z}[X]$. The* height *of f, denoted by $H(f)$, is the maximum of the absolute values of its coefficients.*

Definition 2.6 *A triple of polynomials $(f, g, h) \in \mathbb{Z}[X]^3$, with degrees n, k, l respectively, is* cantorian *if:*

- *f and g have the same degree, say $n = k \geq 1$*
- *$l = n - 1$*
- *g is monic (i.e. $g_k = 1$) and $g_0 \neq 0$*
- *$f(X)$ divides $g(h(X))$*
- *either $|g_{k-1}| > 1 + \sum_{j=0}^{k-2} |g_j|$ or $g_{k-1} = 0$ and $g_{k-2} > 1 + \sum_{j=0}^{k-3} |g_j|$*

The following theorem, proved by Cantor, states that a cantorian triple can act as an irreducibility certificate:

Theorem 2.7 *If (f, g, h) is cantorian, then f is irreducible.*

I will now sketch the proof (it is in [6, Lemma 1 and 2]).
First it is proved that $g \in \mathbb{Z}[X]$, satisfying the cantorian conditions, is irreducible. To conclude it is proved that if f divides $g(h(X))$ with g irreducible, then f itself is irreducible.

And so we have the *Cantor-criterion* and the corresponding *Cantor-certificate*:

Certificate 2.8 (Cantor) *A triple (f, g, h) of polynomials in $\mathbb{Z}[X]$.*

Criterion 2.9 *There exists a triple (f, g, h) of polynomials in $\mathbb{Z}[X]$, that is cantorian.*

Cantor has also proved that such a certificate, with the polynomials bounded in height, can always be found:

Theorem 2.10 *If f is an irreducible polynomial then there are polynomials g and h such that (f, g, h) is cantorian and $H(g), H(h) \leq (128)^{2 \cdot \deg(f)^2} H(f)^{3 \cdot \deg(f)^3}$.*

Again, the proof is in [6, Lemma's 4 to 7] and we shall sketch it:
A huge $n-1$-dimensional cube, that is divided in many small cubes, is created, as well as linear forms depending on the (complex) roots of f. We let these forms act on vectors and in this way get vectors in the cube. Now the pigeonhole principle is used to find two vectors that are in the same little cube. Out of these two vectors we can form $h \in \mathbb{Z}[X]$ that satisfies the cantorian conditions. From h it is easy to construct g, and eventually it is proved that g also satisfies the cantorian conditions.

Because g and h are bounded we see that we can check whether a triple (f, g, h) is cantorian or not, in time polynomial in the size of f. So now we have easy-to-check certificates for every irreducible polynomial; a checking-algorithm just has to check the (easy) cantorian conditions on bounded polynomials!
Alas, we do not really *have* them: in Cantor's article g and h are constructed, but as we have seen in the sketch of the proof above this is an enormous task.

But now that we have Cantor's theorem we may try to find certificates in an other way. Unfortunately there is no fast method known at this moment. We know that a solution exists for all irreducible polynomials so we can try all g and h with the correct height. Obviously the height-limit is quite high here so this criterion does not guarantee that we can find a certificate quickly.

Another problem is that the mathematics involved is quite advanced and therefore it would take a lot of time to formalize all of it. And this is what we wanted, to be absolutely certain of irreducibility!

Example Suppose we want to show that $f = X^3 + X + 1$ is irreducible, with a Cantor-certificate. Then we can start trying all polynomials g of degree 3 and all polynomials h of degree 2 with coefficients smaller than $(128)^{2 \cdot \deg(f)^2} H(f)^{3 \cdot \deg(f)^3} = 128^{18}$. But if we try we can find $g = X^3 + 10X^2 + X + 1$ and $h = -4X^2 + 3X - 6$, but it is clear that we can not be certain of finding them fast. Once we have these g and h it is easy to check that (f, g, h) is cantorian.

Coming to a conclusion we can now say that there exists a Cantor-certificate for every irreducible polynomial and that such a certificate can be checked easily. But the major drawbacks are that at this moment there is no known method to *find* certificates fast, and that it will be an enormous task to formalize the mathematics involved.

2.3 Weinberger's Certificate and the Riemann Hypothesis

A certificate which has a greater disadvantage is that of Weinberger. In [28] he proves that we can compute the number of irreducible factors of a polynomial in time polynomial in the size of f *if* the Riemann Hypothesis (RH) holds. So we can also quickly compute whether it is irreducible (i.e. the number of factors is 1), or not.

This is how it works:
It is proved that the average number of linear factors of f modulo all primes, is the number of irreducible factors of f. Therefore we factor f modulo $2, 3, 5, 7, \ldots$, until the average numbers of linear factors of f modulo a prime can be determined. The question now is where we can stop to be sure of the average number. This is where the RH is needed. With this hypothesis (which says something about the distribution of prime numbers), a bound on the primes to be used can be calculated.
The certificate for an irreducible polynomial f would then consist of all modulo-factorisations which were needed according to the RH.

We now see what the problem is: if the RH does not hold then the average after a finite number of primes might not be the total average. So this algorithm could either give a too high or too low number of irreducible factors of f as an answer. So the algorithm (and thus a certificate) relies too heavy on an unproved conjecture.

Remark Even when the hypothesis would be proved, another problem is that *in practice* the bound on the primes would be very large, so that the checking would take a lot of time.

2.4 The LLL-algorithm

In 1982 the LLL-algorithm was introduced by Lenstra, Lenstra and Lovàsz [20]. With this algorithm factorizing f can be done in time polynomial in the size of f. There have been made various improvements on this method and the algorithm used in Magma for example is based on a version by Van Hoeij [17].

So we could add an empty certificate to an irreducible polynomial, because using the LLL-algorithm we can check that is irreducible in polynomial time.
The problem for us is that we want to be completely sure that a polynomial is irreducible. As, for example, there might be bugs in a computer algebra system, we want to have a formalized algorithm. The mathematics used to prove the correctness of the LLL-algorithm is in the present state of technology too far out of reach to be formalized.

Chapter 3

The Certificates

In this chapter I will give practical irreducibility certificates for polynomials over $\mathbb{Z}$. The polynomial will be denoted by f;

$$f = f_n X^n + f_{n-1} X^{n-1} + \ldots + f_1 X + f_0 \in \mathbb{Z}[X],$$

with $n \geq 2, f_n \neq 0$, and $\gcd(f_0, \ldots, f_n) = 1$.

In every section I will first give the certificate and criterion and then prove that the criterion guarantees irreducibility. Next comes some additional information, depending on the certificate and eventually I will describe how the certificate can be checked (and how fast).

3.1 Eisenstein

The criterion of Eisenstein is well-known and, if the polynomial is in the right form, it is very easy to use, even without a computer. A disadvantage is that in practice polynomials with large degrees and coefficients have no Eisenstein-style certificate. This will be illustrated with examples in Chapter 6.

3.1.1 The Certificate

Certificate 3.1 (Eisenstein) $p \in \mathbb{N}$

Criterion 3.2 *There exists a prime p such that $p \nmid f_n, p^2 \nmid f_0$, and $p \mid f_i$ for $0 \leq i < n$.*

Theorem 3.3 *If the criterion holds for f, then f is irreducible.*

> **Proof** Suppose we have p as in the proposition. Suppose also that f is reducible, say $f = gh = (g_r X^r + \ldots + g_0)(h_{n-r} X^{n-r} + \ldots + h_0)$, with $1 \leq r < n$. Then $f_0 = g_0 h_0$, so $p \mid g_0 h_0$ but $p^2 \nmid g_0 h_0$, so let's say that $p \mid g_0, p \nmid h_0$.
> Now use the principle of induction to prove that p divides all coefficients of g: We know that $p \mid g_0$. Suppose that p divides $g_0, g_1, \ldots, g_{i-1}$ ($i < n$ of course). Now we let $h_j = 0$ if $j > n - r$ (so they are defined now). Then we have that $f_i = g_0 h_i + g_1 h_{i-1} + \ldots + g_{i-1} h_1 + g_i h_0$, and we know that $p \mid f_i$, so p divides the right hand side of the equation. The assumption is that p divides $g_0, \ldots, g_{i-1}$,

so we see that p must divide $g_i h_0$ as well. Since $p \nmid h_0$, we have that $p \mid g_i$.
Hence p divides all coefficients of g, so it divides g itself.
This means that $p \mid f$, which is a contradiction because $p \nmid f_n$.
So f is irreducible. □

Example Let $f = X^5 + 500X^4 - 15X^3 + 10X - 30$, then f is irreducible by Eisenstein, with $p = 5$. So the certificate is 5 here.

Remark Let $f \in \mathbb{Z}[X],\ a \in \mathbb{Z}$. Then $f(X)$ is irreducible $\Leftrightarrow$ $f(X + a)$ is irreducible.

Proof If $f(X)$ has a non-trivial factor $g(X)$ then $f(X + a)$ has the non-trivial factor $g(X + a)$ and vice versa. Hence $f(X)$ does not have a non-trivial factor $\Leftrightarrow$ $f(X + a)$ does not have a non-trivial factor. □

And now we have a new certificate:

Certificate 3.4 *(p, a), with $p \in \mathbb{N}, a \in \mathbb{Z}$.*

Criterion 3.5 *There exist $p \in \mathbb{N}, a \in \mathbb{Z}$ such that $f(X + a)$ is irreducible by Eisenstein, with p.*

Theorem 3.6 *If the criterion holds for f, then f is irreducible.*

The proof is clear with the remark above.

Example Let $f = X^3 + 27X^2 + 222X + 562$, then $f(X)$ does not have an Eisenstein-certificate, but $f(X - 1) = X^3 + 24X^2 + 171X + 366$ does, with $p = 3$. So f is irreducible. The certificate is $(3, -1)$.

3.1.2 Checking

To check a certificate 3.4 we have to compute $f(X + a)$ and check all coefficients. We also need a proof that p is indeed prime. Because p can not be ‘really big’ (it divides the constant coefficient for example) the complete check is quite fast.

3.1.3 Schönemann’s Theorem

The criterion of Eisenstein was formulated in 1850 [13] and was in fact a special case of a theorem already formulated by Schönemann in 1846 [26].

Theorem 3.7 *Let $F = f^n + pg \in \mathbb{Z}[X]$, with $n \geq 1, p \geq 2$ a prime, $f, g \in \mathbb{Z}[X]$ such that $\deg(f^n) > \deg(g)$ and $\overline{f}$ is irreducible in $\mathbb{F}_p[X]$ and $\overline{f}$ does not divide $\overline{g}$ in $\mathbb{F}_p[X]$. Then F is irreducible.*

Proof Suppose that $F = F_1 F_2$ is a non-trivial factorisation. Then $\overline{F} = \overline{F_1 F_2}$ in $\mathbb{F}_p[X]$. $\overline{F} = \overline{f^n}$, $\overline{f}$ is irreducible in $\mathbb{F}_p[X]$, and so it follows that there exist $u, v \in \mathbb{N},\ u + v = n$ and $g_1, g_2 \in \mathbb{Z}[X]$ such that

$$F_1 = f^u + pg_1, F_2 = f^v + pg_2,$$

with $\deg(g_1) < u \cdot \deg(f),\ \deg(g_2) < v \cdot \deg(f)$. So

$$F = f^n + pg = (f^u + pg_1)(f^v + pg_2) = f^n + p(f^u g_2 + f^v g_1 + pg_1 g_2)$$

Hence $g = (f^u g_2 + f^v g_1 + pg_1 g_2)$. Without loss of generality we may suppose that $u \leq v$, so that $g = f^u \cdot h + pg_1 g_2$, with $h = g_2 + f^{v-u} g_1 \in \mathbb{Z}[X]$. Viewing this equation modulo p we now have that $\overline{g} = \overline{f}^u \cdot \overline{h}$. So $\overline{f}$ divides $\overline{g}$ in $\mathbb{F}_p[X]$: contradiction, unless $u = 0$. But if $u = 0$, then $F_1 = 1$ and we have a trivial factorisation of F, and thus we arrive at a contradiction now as well. □

We can see that Eisenstein's criterion is a special case by taking $f(X) = X$ and $g(X) = \frac{1}{p}(f_{n-1}X^{n-1} + \ldots + f_1 X + f_0)$. As $p \mid f_i$ for $0 \leq i < n$, we see that $g \in \mathbb{Z}[X]$.

3.2 Bunyakovsky's Conjecture

In the previous chapter we have seen Weinberger's algorithm, which needed a conjecture to be certain of correct certificates. In this section we shall discuss certificates that always imply irreducibility. But there is a conjecture (of Bunyakovsky) that guarantees the existence of such certificates for every irreducible polynomial. The certificates consist of one or more integers i for which f is a prime or a unit. First we shall describe the certificates that need multiple evaluations.

3.2.1 Multiple Evaluations

These certificates need multiple evaluations. They were found in [21].
First two remarks that will be used in the proofs.

Remark 1 Let $g \in \mathbb{Z}[X]$ be a polynomial of degree $k \geq 1$. Then for $a \in \mathbb{Z}$ there are at most k integers i such that $g(i) = a$. To see this suppose that there are more than k such integers. Then the polynomial $g(i) - a$, also of degree $k \geq 1$ has more than k integer zeroes, which is impossible.

Remark 2 If f is reducible, say $f = gh$, with $\deg(g), \deg(h) \geq 1$, and $f(i)$ is prime for $i \in \mathbb{N}$, then $g(i)$ or $h(i)$ is a unit. As this is in $\mathbb{Z}$, the units are 1 and -1.

Certificate 3.8 $2n + 1$ *integers.*

Criterion 3.9 *There exist* $2n + 1$ *integers* $i_1, \ldots, i_{2n+1}$*, such that* $f(i_j)$ *is a prime or unit for all* j*.*

Theorem 3.10 *If the criterion holds for* f*, then* f *is irreducible.*

Proof Suppose $f = gh$, with $\deg(g) = k \geq 1, \deg(h) = l \geq 1$. Then from Remark 1 we know that $g(i) = 1$ for at most k integers i, also $g(i) = -1$ for at most k integers i. Thus the maximum number of times that $g(i)$ equals 1 or -1 is $2 \cdot k$. In the same way $h(i)$ can equal 1 or -1 at most $2 \cdot l$ times. As a result of Remark 2, the maximum possible number of distinct values of i for which $f(i)$ can be a prime or a unit is $2(k + l) = 2n$. This is a contradiction with our assumption, so f is irreducible. □

Example Let $f = X^4 + 3X^3 + 17X^2 - 89X + 3$. Then we have the following table:

X	-5	-2	-1	0	2	4	5	7	8
$f(X)$	1123	241	107	3	-67	367	983	3643	6011

So we have 9 $(= 2 \cdot \deg(f) + 1)$ prime values and we have that f is irreducible.
The corresponding certificate is $(-5, -2, -1, 0, 2, 4, 5, 7, 8)$.

We can do with fewer primes, but then we need an extra proposition and some conditions on the distribution of these evaluation points.

Proposition 3.11 *Let $g \in \mathbb{Z}[X]$, with $\deg(g) = d$ and $i_1, i_2 \in \mathbb{Z}$, such that $|i_1 - i_2| > 2$ and $|g(i_1)| = |g(i_2)| = 1$. Then $g(i_1) = g(i_2)$.*

> **Proof** Suppose $g(i_1) = 1, g(i_2) = -1$. Subtracting these two equations we get
>
> $$g_d(i_1^d - i_2^d) + \ldots + g_2(i_1^2 - i_2^2) + g_1(i_1 - i_2) = 2.$$
>
> All terms in the sum are divisible by $i_1 - i_2$, so $(i_1 - i_2) \mid 2$. This is in contradiction with our assumption that $|i_1 - i_2| > 2$. In the same way we can arrive at a contradiction if $g(i_1) = -1, g(i_2) = 1$. So $g(i_1) = g(i_2)$. □

Certificate 3.12 *$n + 1$ integers.*

Criterion 3.13 *There exist $n + 1$ integers i_j that differ pairwise by more than 2, such that $f(i_j)$ is a prime or unit for all j.*

Theorem 3.14 *If the criterion holds for f, then f is irreducible.*

> **Proof** Suppose $f = gh$, with $\deg(g), \deg(h) \geq 1$. With the previous proposition and Remark 1 we know that there are at most $\deg(g)$ integers i differing pairwise by more than 2, for which $|g(i)| = 1$. In the same way we have that there are at most $\deg(h)$ integers i differing pairwise by more than 2, for which $|h(i)| = 1$. By Remark 2 there can now only be $n(= \deg(g) + \deg(h))$ i's (differing pairwise by more than 2) for which $f(i)$ is a prime or a unit. This is in contradiction with our assumption, so f is irreducible. □

Example Let $f = X^4 + 3X^3 + 17X^2 - 89X + 3$. Then we have the following table:

X	-8	-5	0	4	7
$f(X)$	4363	1123	3	367	3643

So we have 5 (=$\deg(f)$ + 1) prime values and we have that f is irreducible.
The certificate is $(-8, -5, 0, 4, 7)$.

3.2.2 One Evaluation

For this certificate, by John Brillhart [4], we just need one prime evaluation.

Certificate 3.15 $x_0 \in \mathbb{Z}$

Criterion 3.16 *There exist $m \in \mathbb{N}^*$ greater than the moduli of the (complex) zeros of f, and $x_0 \in \mathbb{Z}$ with $|x_0| \geq m + 1$, such that $f(x_0)$ is prime.*

Theorem 3.17 *If the criterion holds for f, then f is irreducible.*

> **Proof** Suppose $f = gh$, with $\deg(h) \geq 1$. Then m is certainly greater than the moduli of the zeros of g and h.
> $|f(x_0)| = |g(x_0)|\ |h(x_0)|$ is prime.
> Let $\deg(h) = r$, then h_r is its leading coefficient. Let $\alpha_1, \ldots, \alpha_r$ be the zeros of h. Then $|h(x_0)| = |h_r| \prod_{i=1}^{r} |x_0 - \alpha_i| > 1$, because $|h_r| \geq 1, |x_0| \geq m + 1$ and for all $i : m > \alpha_i$. So $|h(x_0)|$ is prime. If $\deg(g) \geq 1$ then (in the same way) $|g(x_0)| > 1$, which gives rise to a contradiction, so g is constant and $|g(x_0)| = 1$, hence $g = \pm 1$, and we have that f is irreducible. □

Example Let $f = X^4 + 3X^3 + 17X^2 - 89X + 3$. The complex zeros of f were calculated and were approximately: $0.034, 2.708, -2.871 + 4.941i, -2.871 - 4.941i$. So the moduli are less than 1, 3, 6 and 6. Hence we can take $m = 6$.
We now see that $f(7) = 3643$ (which is prime), so f is irreducible.
The certificate simply is 7.

Example One larger example:
Let $f = X^{37} + 90823490832X^{19} - 4082408240240000333$. $m = 5$ suffices, and $f(-150) =$ –3276246613611855449562426656685944831261561146820068359375000040824082402400000333 completes the proof. Of course a problem now is: how do we know that the monstrous evaluation is really prime? For that purpose we might for example use Pocklington's certificate [24, 25]. But when primes become this big this certificate will take a very long time to check.

3.2.3 Bunyakovsky's Conjecture

So we have seen the certificates, but do they always exist?

To investigate this, we first need a definition:

Definition 3.18 *Let $f \in \mathbb{Z}[X]$. The* fixed divisor *of f, denoted by d_f, is the largest positive integer d such that $d \mid f(a)$ for all $a \in \mathbb{Z}$.*

Example Let $f = X^2 + 9X - 4$. Now $f(i)$ is always even, $f(0) = -4$ and $f(1) = 6$, so here $d_f = 2$.

We see that if $d_f > 1$, then $f(i)$ can be prime for only finitely many i's. In 1857 Bunyakovksy stated the following conjecture [5]:

Conjecture 3.19 *If f is irreducible, then $d_f^{-1} f(i)$ is prime for infinitely many i.*

At this moment we only know this conjecture to be true for the special case that $\deg(f) = 1$. Then it follows from Dirichlet's theorem on primes in arithmetic progressions. The conjecture is generally believed to be true.

So, if Bunyakovsky's conjecture holds we always have certificates for irreducible polynomials. But what we really need is an even stronger conjecture that puts an upper bound on where to find a prime evaluation.

3.2.4 Checking

A certificate consists of the i's for which $f(i)$ is prime or a unit. We then have to calculate the $f(i)$'s and prove that they are prime (or a unit). These primes may become monstruous as we have seen in an example, so we really can't say how long it takes to check this certificate. (And finding a certificate we can be completely certain of will of course cost a lot more time).

But in practice, when we trust the primality check of Magma and try to prove irreducibility for some smaller polynomials (something like $\deg(f) \leq 20$ and coefficients $\leq 10^6$), it works quite well, especially Brillhart's certificate 3.15. For test results, see Chapter 6.

3.3 Modulo p

In this section p will always be a (positive) prime number.

In practice, factoring over small finite fields is easier than over $\mathbb{Z}$. Unfortunately an irreducible factor of f might factor over a finite field. But a modulo-factorisation can still provide us with some information.

3.3.1 Irreducibility over Finite Fields

Certificate 3.20 $p \in \mathbb{N}$

Criterion 3.21 *There exist a prime p such that $\overline{f}$ is irreducible over $\mathbb{F}_p$, and p does not divide the leading coefficient f_n.*

Theorem 3.22 *If the criterion holds for f, then f is irreducible.*

Proof If $f = gh$ in $\mathbb{Z}[X]$ then $\overline{f} = \overline{g}\overline{h}$ in $\mathbb{F}_p[X]$.
So if $\overline{f}$ is irreducible over $\mathbb{F}_p$ and $p \nmid f_n$, then f is irreducible over $\mathbb{Z}$. □

Example Let $f = X^4+3X^3-2X^2+X-17$ in $\mathbb{Z}[X]$, so that in $\mathbb{F}_3[X] : \overline{f} = X^4+X^2+X+\overline{1}$, which is irreducible (see next proposition). We conclude that f is irreducible.
The certificate is 3.

To check a certificate we need a way to prove irreducibility in $\mathbb{F}_p[X]$. For this purpose we have the next proposition:

Proposition 3.23 *A polynomial $f \in \mathbb{F}_p[X]$ with $n = \deg(f) \geq 1$ is irreducible if and only if:*

(i) $f \mid X^{p^n} - X$, *and*

(ii) $\gcd(X^{p^{n/t}} - X, f) = 1$ *for all prime divisors t of n.*

> **Proof** The proof is in [15, page 382]. We use the fact that $X^{p^d} - X \in \mathbb{F}_p[X]$ is, for any $d \geq 1$, the product of all monic irreducible polynomials in $\mathbb{F}_p[X]$ whose degree divides d. To prove this some algebra of finite fields is needed, like Fermat's little theorem. □

3.3.2 Modulo-combinations

In this subsection the 'modulo p'-method will first be introduced, the reasons for the use of this method will be given, and we will take a look at the certificates of this method.

With certificate 3.20 we have seen that it is useful to factor our polynomial f modulo a prime p. (Remark: In Chapter 6 we prove that we only need certificates for monic polynomials. For these polynomials we automatically have that p does not divide the leading coefficient.) For this certificate we were only interested in primes p such that f modulo p is irreducible, but we can take advantage of (almost) any prime p.

We know that a factor of f remains a factor when working modulo p (and these factors may be reducible over $\mathbb{F}_p$). Let D_p be the set of degrees of all factors (not necessarily irreducible) of f modulo p, and D the set of degrees of all factors of f (in $\mathbb{Z}[X]$). Then we have that $D \subseteq D_p$. This holds for any prime, so if $p_1, \ldots, p_k$ are all different primes then $D \subseteq \bigcap_{i=1}^k D_{p_i}$. So we have the next powerful certificate and criterion:

Certificate 3.24 *A set of natural numbers $(p_1, \ldots, p_k)$.*

Criterion 3.25 *There exist primes $p_1, \ldots, p_k$ such that $\bigcap_{i=1}^k D_{p_i} = \{0, \deg(f)\}$.*

Example Let $f = X^5 - 45X^4 - 261X^3 - 8404X^2 - 18078X - 135764$. For $p = 3$ we have that $\overline{f} = X^5 + 2X^2 + 1 = (X^3 + 2X^2 + 2X + 2)(X^2 + X + 2)$, so that $D_3 = \{0, 2, 3, 5\}$. For $p = 5$ we have that $\overline{f} = X^5 + 4X^3 + X^2 + 2X + 1 = (X + 1)(X^4 + 4X^3 + X + 1)$, so $D_5 = \{0, 1, 4, 5\}$. Now $D_3 \cap D_5 = \{0, 5\}$, hence the primes 3 and 5 form an irreducibility certificate for f.

Implementation

I have implemented an algorithm based on the 'modulo p'-method in the Computer Algebra System Magma [2]. The algorithm is available online at `http://www.math.ru.nl/~bosma/students/kirkels`.
It works with a related concept, that of partitions of the degrees of factors of f.

Definition 3.26 *By P_p we shall denote the sequence of degrees of irreducible factors of f modulo p (in non-decreasing order, for uniqueness), so P_p is a partition of* $\deg(f)$. *P_p is called the* decomposition type *of f modulo p.*

Example In the example above we have that $P_3 = [2, 3]$ and $P_5 = [1, 4]$. For $f = X^5$ we have that $P_p = [1, 1, 1, 1, 1]$ for every prime p. So P_p has to be a tuple: it may contain the same number several times.

Definition 3.27 *Let P_1 and P_2 be two partitions. We say that P_1 is a* parent *of P_2 if every element of P_1 is a sum of elements of P_2 and every element of P_2 appears exactly once in these sums.*

Example $[1, 2, 3, 4]$ is a parent of $[1, 1, 1, 2, 2, 3]$: take 1+2 and 1+3 as 3 and 4. Or take 1+1 and 2+2 as 2 and 4.

And now we can formulate a certificate in the partition-style:

Certificate 3.28 *A set of natural numbers $(p_1, \ldots, p_k)$.*

Criterion 3.29 *Primes $p_1, \ldots, p_k$ such that the only common parent of all the P_{p_i} is* $[\deg(f)]$.

These partitions give us more information than all the degrees:

Example We compare $f = X^3$ and $g = X^3 + X$. Then $P_3(f) = [1, 1, 1], P_3(g) = [1, 2]$, as $g = X(X^2 + 1)$, so there is a difference. But $D_3(f) = D_3(g) = \{0, 1, 2, 3\}$. So the *partition* modulo 3 tells us that g has a factor of degree 2, while all the *degrees* do not give any information, because every degree is possible.

What we also know is that: $(p_1, \ldots, p_k)$ is an irreducibility certificate for f of type 3.28 if and only if it is an irreducibility certificate for f of type 3.24. This means that the partitions can only give more information when irreducibility can not yet be proved. The advantage of having more information will be used in the final algorithm, in Chapter 5.

Reasons for using this algorithm

There are various reasons for using this algorithm:

- We almost always get a certificate. This will be made precise in Chapter 4.
- Certificates are generated very fast. Test results will show this, see Chapter 6.
- We can make the certificates in such a way that they are easy to check.
  We will treat that now.

3.3.3 Checking

Easier-to-check Certificates

Until now the certificates simply consisted of the primes for which the modulo-factorisations proved irreducibility. But in the process of computing these primes we of course compute these modulo-factorisations. And the certificate will be larger, but much easier to check if we incorporate these factorisations.

Example Let $f = X^5 - 45X^4 - 261X^3 - 8404X^2 - 18078X - 135764$, as in the previous examples. The certificate was simply $(3, 5)$, but now we add the factorisations, so that the certificate becomes

$$(\{3, (X^3 + 2X^2 + 2X + 2)(X^2 + X + 2)\}, \{5, (X + 1)(X^4 + 4X^3 + X + 1)\}).$$

Checking Certificates

In proposition 3.23 we gave a simple way to prove irreducibility over a finite field. We use this to check a certificate.

When we get a certificate we do the following:

1. For every prime: check that it is prime. As we will see later primes will be quite small, so if we trust a small prime-list this check can be very fast.
2. For every prime p and the factorisation of f modulo p: check that it is a factorisation modulo the prime.
3. For every factor: check that it is irreducible using proposition 3.23.
4. Construct D_p for every prime and check that the intersection of all of them really is $\{0, \deg(f)\}$.

The next chapter will give more information on how many factorisations we have to check, and whether we will always find a certificate or not. This will be accomplished by looking at the corresponding Galois theory.

Chapter 4

Galois Theory and Modulo-Factorisations

In the previous chapter we have seen several irreducibility certificates. We will take a closer look at certificate 3.28 which combines modulo p-factorisations. In this chapter that certificate is the only one that we will discuss.

In the first section we will see examples of irreducible polynomials that do not have such a certificate. Then we will discuss some Galois theory, this will make the situation clearer and afterwards we can use results from Galois theory to understand the existence of our certificate better.

4.1 Polynomials without Certificate

There are irreducible polynomials that are reducible modulo every prime. This has first been observed by D. Hilbert [16].

The standard example of a polynomial which has this property is in the next example.

Example Let $f = X^4 + 1$. Then f is irreducible, but reducible modulo every prime.

> **Proof** $f(X+1) = X^4+4X^3+6X^2+4X+2$, which is irreducible by Eisenstein's criterion with $p = 2$. So f is irreducible over $\mathbb{Z}$.
> For $p = 2$ we have that $\overline{f} = (X + 1)^4$. For p odd there are three different cases:
> $p \equiv 1 \mod 4$, then -1 is a square in $\mathbb{F}_p$, so $\overline{f} = (X^2-\sqrt{-1})(X^2+\sqrt{-1}) \in \mathbb{F}_p[X]$
> $p \equiv 7 \mod 8$, then 2 is a square in $\mathbb{F}_p$, so $\overline{f} = (X^2 + \sqrt{2}X + 1)(X^2 - \sqrt{2}X + 1)$
> $p \equiv 3 \mod 8$, then -2 is a square in $\mathbb{F}_p$,
> so $\overline{f} = (X^2 + \sqrt{-2}X - 1)(X^2 - \sqrt{-2}X - 1)$
> So $\overline{f}$ is reducible modulo every prime p. □

The information on squares in finite fields (using quadratic reciprocity) can be found in [1, page 181] for example.

With this example we see that we even have something stronger: the combination of modulo-factorisations will only tell us that *if* there exists a non-trivial factor then it must have degree 2.

As we are interested in finding certificates, we would also like to know for which polynomials no certificate can be found. In order to find some structure, we tried to find more polynomials of degree 4 that do not have a modulo p-certificate. Later we will see that degree 4 is the first interesting degree. We have found a big group of polynomials for which no modulo p-certificate can be found. There are other polynomials of degree 4 that do not have such a certificate, but we shall also see that they are related with the following group of polynomials:

Examples Let $a, b \in \mathbb{Z}$. Then we have proved that $f = X^4 + aX^2 + b^2$ has a factor of degree 2 modulo every prime p.

Proof

First we derive sufficient conditions.
Let p be a prime. From now on we work modulo p, so a, b are in $\mathbb{F}_p$ now. Then we have to prove that there are numbers c, d, e and f in $\mathbb{F}_p$ such that
$X^4 + aX^2 + b^2 = (X^2 + cX + d)(X^2 + eX + f) =$
$X^4 + (c+e)X^3 + (d + f + ce)X^2 + (cf + de)X + df.$
We take $e = -c$ to have the term of degree 3 correct:
$X^4 + aX^2 + b^2 = X^4 + (d + f - c^2)X^2 + (cf - cd)X + df.$
Taking $c = 0$ or $d = f$ fixes our linear term.

$c = 0$ **:** $X^4 + aX^2 + b^2 = X^4 + (d + f)X^2 + df$
So we take $f = a - d$ to get:
$X^4 + aX^2 + b^2 = X^4 + aX^2 + (da - d^2)$ and now we know:
If there is $d \in \mathbb{F}_p$ such that $da - d^2 = b^2$ then f has a factor of degree 2.

$d = f$ **:** $X^4 + aX^2 + b^2 = X^4 + (2d - c^2)X^2 + d^2$
Now we can either take $d = b$ or $d = -b$:

$d = b$ **:** $X^4 + aX^2 + b^2 = X^4 + (2b - c^2)X^2 + b^2$
What we need now is that $2b - c^2 = a$, thus $2b - a = c^2$.
In terms of the Legendre symbol (and taking $a, b \in \mathbb{Z}$):
If $(\frac{2b-a}{p}) = 1$ then f has a factor of degree 2.

$d = -b$ **:** $X^4 + aX^2 + b^2 = X^4 + (-2b - c^2)X^2 + b^2$
What we need now is that $-2b - c^2 = a$, thus $-2b - a = c^2$.
In terms of the Legendre symbol (and taking $a, b \in \mathbb{Z}$):
If $(\frac{-2b-a}{p}) = 1$ then f has a factor of degree 2.

Now we shall show that these found conditions can always be fulfilled, i.e.:
For every prime p, for all $a, b \in \mathbb{F}_p$ we have that
$(\frac{-2b-a}{p}) = 1$ or $(\frac{2b-a}{p}) = 1$ or that there exists $d \in \mathbb{F}_p$ such that $da - d^2 = b^2$.

For $p = 2$ this is trivial: if $a = 0$ then $d = b$ does it.
Otherwise $a = 1$ (modulo p) and then $(\frac{2b-a}{p}) = (\frac{2b-1}{2}) = (\frac{1}{2}) = 1$.

So from now on p is odd.

If p | a : We must prove that $(\frac{2b}{p}) = 1$ or $(\frac{-2b}{p}) = 1$ or $(\frac{-b^2}{p}) = 1$.
If $p \equiv 1 \mod 4$ we have that $(\frac{-1}{p}) = 1$, so then $(\frac{-b^2}{p}) = (\frac{-1}{p})(\frac{b^2}{p}) = 1 \cdot 1 = 1$.
Otherwise we have that $(\frac{-1}{p}) = -1$, so $(\frac{2b}{p}) = 1$ or $(\frac{-2b}{p}) = 1$, which had to be proved. **If p | b :** We can take $d = 0$ and are done.
So we can assume that $p \nmid a$ and $p \nmid b$.

Suppose that $(\frac{2b-a}{p}) = (\frac{-2b-a}{p}) = -1$, then $(\frac{(2b-a)(-2b-a)}{p}) = 1$. So we have $c := \sqrt{(2b-a)(-2b-a)} \in \mathbb{F}_p$. We can divide by 2 as p is odd.
So we have $d := \frac{c+a}{2} \in \mathbb{F}_p$. Now $da - d^2 = \frac{ca+a^2}{2} - (\frac{c+a}{2})^2 = \frac{2ac+2a^2}{4} - \frac{c^2+2ac+a^2}{4} = \frac{a^2-c^2}{4} = \frac{a^2-(2b-a)(-2b-a)}{4} = \frac{a^2+4b^2-a^2}{4} = b^2$. So in this case we have a correct d.

The only remaining cases are $(\frac{2b-a}{p}) = 0$ and $(\frac{-2b-a}{p}) = 0$. In the first case $a = 2b$ in $\mathbb{F}_p$ and we can take d to be b so that $da - d^2 = ba - b^2 = 2b^2 - b^2 = b^2$. In the second case $a = -2b$ in $\mathbb{F}_p$ and we can take d to be $-b$ so that $da - d^2 = -ba - b^2 = 2b^2 - b^2 = b^2$.

So we see now that f always has a factor of degree 2 modulo a prime p. □

And now we have that every irreducible polynomial of the form $f = X^4 + aX^2 + b^2$ is reducible modulo every prime.

By trying more irreducible polynomials it seemed as if there were only examples of polynomials of composite degree that were reducible modulo every prime. This will become clear as we look at some Galois theory next.

4.2 Galois Theory

Let $f \in \mathbb{Z}[X]$ be primitive, monic and irreducible over $\mathbb{Z}$, with degree n. Let G be the Galois group of the splitting field K of f over $\mathbb{Q}$. Then each element σ of G (i.e. each automorphism of K) can be seen as a permutation of the n roots of f and in that way it has a unique decomposition into disjoint cycles, say of lengths $\lambda_1 \leq \ldots \leq \lambda_r$. Because $\lambda_1 + \ldots + \lambda_r = n$ we have that $\lambda = (\lambda_1, \ldots, \lambda_r)$ is a partition of n. We shall call λ the *cycle type* of σ.

Definition 4.1 *Let λ be a partition of n, then we denote by $H_\lambda (\subseteq G)$ the set of automorphisms of K that have cycle type λ.*

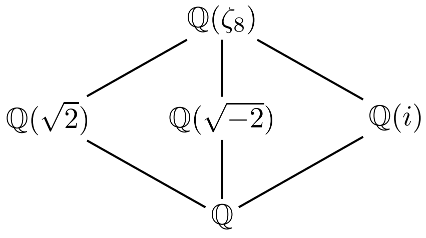
Definition 4.2 *Again λ is a partition of n. By $\mu(\lambda)$ we denote the relative frequency with which the cycle type λ occurs in G. So $\mu(\lambda) = \#H_\lambda / \#G$.*

We have already seen partitions of n: when we discussed combining modulo-factorisations. And now we have a beautiful theorem of Frobenius connecting these partitions:

Theorem 4.3 (Frobenius) *The density of the set of primes p for which f has a given decomposition type λ exists, and is equal to $\mu(\lambda)$.*

The existence of the so called Frobenius substitution proves a stronger result, namely that all factorisations modulo a prime have their decomposition type equal to a cycle type that appears in G. Chebotarëv has proved a stronger density theorem that uses this substitution. More information about Frobenius' theorem and a proof of Chebotarëv's density theorem can be found in [27].

Example Let $f = X^4 + 1$. Then f is the minimal polynomial of ζ_8, so the splitting field of f is $\mathbb{Q}(\zeta_8)$ and we have the following diagram of the subfields:



The Galois group of f is the Klein group V_4. This group consists of four elements: the identity and three elements of cycle type $(2, 2)$. So we now know that modulo one fourth of all primes f splits in four linear factors. And for all other primes f splits in two factors of degree 2.

And so we have now proved that there is no modulo p-certificate for $f = X^4 + 1$. In general to determine for which polynomials we can not find a certificate, it may be very helpful to consider Galois groups of polynomials.

But determining a Galois group of a polynomial is time-consuming and hard to check, so we will look at the situation in a very general way and use the Galois groups just to make general statements about the modulo p-certificates.

4.2.1 Which Galois groups can Occur?

We have already mentioned that G can be seen as a group permutating the roots of f, hence as a subgroup of S_n.

To talk about the Galois groups that can occur we need the following definition:

Definition 4.4 *A subgroup G of $S(Z)$ is called* transitive *if for every ordered pair (x, y) of elements of Z there exists $\sigma \in G$ such that $\sigma(x) = y$.*

And we have the following proposition which is proved in [18, page 45], or can be found in any other standard text book in Galois theory.

Proposition 4.5 *f is irreducible $\Leftrightarrow$ G is transitive.*

So we see that the Galois group of an irreducible polynomial of degree n is a transitive subgroup of S_n. And our question has become more precise: which of these groups can not provide its corresponding polynomials with a certificate and how often / when do these groups occur? Results on these questions are treated in the next section.

4.3 Results in Galois Theory

In this section we will deal with the following topics:

1. Polynomials with prime degree.
2. Polynomials with composite degree, especially 4 and 6.
3. Swinnerton-Dyer polynomials.
4. The Galois inverse problem.
5. The distribution of Galois groups.

4.3.1 Polynomials with Prime Degree

Let f be a polynomial of prime degree q. Then the Galois group G of f acts transitively on the set of the q roots of f, and so G possesses an element σ of order q which acts cyclically on the roots of f in the splitting field K of f. So the cycle type of σ is (q) so there are infinitely many primes p for which the decomposition type of f modulo p is (q) as well, in other words: f is irreducible modulo these primes.

So we always get a certificate if f has prime degree!

4.3.2 Polynomials with Composite Degree

In [3] it is proved that for every $n \in \mathbb{N}$ that is composite, there is an irreducible polynomial f of degree n that is reducible modulo every prime.

A sketch of the proof:
A transitive soluble permutation group G of degree n, which does not possess an element of cycle type (n), is constructed. Then $f \in \mathbb{Z}[X]$ is constructed such that G works transitively on the n roots of f. By the previous section we know that f then can never be irreducible modulo a prime.

It is possible that an irreducible polynomial is reducible modulo every prime but still has an irreducibility certificate, we shall see this for a polynomial of degree 4, with Galois group A_4.

It seems likely that for every composite degree there probably are irreducible polynomials for which we can not find a certificate this way. We will now investigate the degrees 4 and 6.

Degree 4

Let f be an irreducible polynomial of degree 4. Then the Galois group G of f is a transitive subgroup of S_4 as we have already seen. S_4 has five transitive subgroups: S_4, D_4, A_4, V_4 and C_4.

S_4, D_4 and C_4 all contain an element of order 4, so in these cases there are primes modulo which f is irreducible, and we are guaranteed of a certificate. For example: if $G \simeq C_4$ then two elements are of order 4, one element has cycle type $(2,2)$ and the identity element has cycle type $(1,1,1,1)$. Hence in this case the set of primes modulo which f is irreducible has density $\frac{1}{2}$ in the set of all primes.

If $G \simeq A_4$ then G contains 8 elements of cycle type $(1,3)$, 3 elements of cycle type $(2,2)$ and the identity element. As the only common parent of $(1,3)$ and $(2,2)$ is (4) we are sure to get a certificate.

But we have seen that there are irreducible polynomials that don't have a certificate. And this is where Klein's group V_4 comes in. As it has just has the identity element and 3 elements with cycle type $(2,2)$ we see that we can't get a certificate for a polynomial with this Galois group.

And now we can conclude that for modulo p-certificates:

$$\text{An irreducible polynomial of degree 4 has no certificate} \Leftrightarrow G \simeq V_4$$

Galois group V_4

Let f have Galois group $G \simeq V_4$. We can then denote the 4 elements of G by e, a, b and c, with e the identity element and $a^2 = b^2 = c^2 = e$. We then also have that $ab = c$. If we let K be the splitting field of f over $\mathbb{Q}$, we have the following Galois diagrams:

$$\begin{tikzcd}
 & <e> \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
<a> \arrow[dr, dash] & <b> \arrow[d, dash] & <ab> \arrow[dl, dash] \\
 & G \simeq V_4 &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & K \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
M_1 \arrow[dr, dash] & M_2 \arrow[d, dash] & M_3 \arrow[dl, dash] \\
 & \mathbb{Q} &
\end{tikzcd}$$

M_1, M_2 and M_3 are quadratic intermediate fields of $K : \mathbb{Q}$.

As the intermediate fields are quadratic we know that there are $d, e \in \mathbb{Z}$, such that K can be written as $\mathbb{Q}(\sqrt{d}, \sqrt{e}, \sqrt{de}) = \mathbb{Q}(\sqrt{d} + \sqrt{e})$. Now we determine the minimal polynomial of $\alpha := \sqrt{d} + \sqrt{e}$ (which must be of degree 4):

$\alpha^2 = d + 2\sqrt{de} + e$, so $(\alpha^2 - (d+e))^2 = 4de$, thus $\alpha^4 - 2(d+e)\alpha^2 + (d+e)^2 - 4de = 0$. As $(d+e)^2 - 4de = d^2 - 2de + e^2 = (d-e)^2$, the minimal polynomial is of the form $X^4 + aX^2 + b^2$. This is the form we found by trying to find polynomials without a certificate in Magma. We know that there are polynomials that have Galois group $G \simeq V_4$, that are not of this form; $X^4 + 4X^3 + 5X^2 + 2X + 1$ for example. But we have now proved that the splitting field K of such a polynomial also is the splitting field of a polynomial of the form $X^4 + aX^2 + b^2$.

Degree 6

For degree 6 we can do something similar as for degree 4. There are 16 transitive subgroups of S_6, of which seven do not have an element of cycle type (6). All seven do contain at least one element of cycle type $(3, 3)$. Six out of seven in addition have an element with one of the following cycle types: $(2, 4), (2, 2, 2)$ or $(1, 5)$. And thus there is only one subgroup for which there are no certificates. This subgroup has order 12 and is generated by the elements $(1, 4)(2, 5)$ and $(1, 3, 5)(2, 4, 6)$. We shall denote this group by P_6. There are polynomials which have P_6 as Galois group, with Magma I have found $f = X^6 + 8X^4 + 11X^2 - 4$ for example.

Higher Degrees

For both degrees 4 and 6 we have found just one Galois group for which no certificate can be found, but for higher degrees the total number of transitive subgroups grows quite hard and with that the number of groups for which there is no certificate grows. For example: for degree 9 there are 34 transitive subgroups of S_9 of which 5 do not give a certificate and for degree 8 we already have 50 transitive subgroups of S_8, of which 24 do not give a certificate. The question whether all subgroups occur as Galois groups will be discussed in subsection 4.3.4.

4.3.3 Swinnerton-Dyer Polynomials

Definition 4.6 *Let $i \in \mathbb{N}^*$. Then we define the ith Swinnerton-Dyer polynomial as*

$$f = \prod (X \pm \sqrt{2} \pm \sqrt{3} \pm \sqrt{5} \pm \ldots \pm \sqrt{p_i}) \in \mathbb{Z}[X]$$

where the product runs over all (2^i) possible combinations of $+$ and $-$ signs, and where p_i is the ith prime.

From Galois theory we know that f is irreducible. But for any prime p, $\mathbb{F}_{p^2}$ contains all the square roots modulo p, so $\sqrt{2} \mod p, \ldots, \sqrt{p_i} \mod p$ are all in $\mathbb{F}_{p^2}$. Hence f modulo p splits in linear factors over $\mathbb{F}_{p^2}$. And so we have that f modulo p splits in factors of degree ≤ 2 over $\mathbb{F}_p$.

So in the best situation (to find a certificate) we have a prime p such that f splits in 2^{i-1} quadratic factors modulo p. And so for $i \geq 2$ we can not find a certificate with our algorithm for the Swinnerton-Dyer polynomials.

4.3.4 The Galois Inverse-problem

We have now seen that the Galois group of an irreducible polynomial f is a transitive subgroup of $S_{\deg(f)}$. A somewhat inverse question is:
Let G be a transitive subgroup of S_n. Does there exist an irreducible polynomial for which the Galois group is G?

This is the so called Galois inverse-problem and until now it has not been solved. However it is conjectured to be true and for $\deg(f) \leq 15$ explicit realizations for every transitive subgroup have been constructed in [19]. In Magma these polynomials form a special database; for every transitive group G (of order $\leq$ 15) one can ask a corresponding polynomial with the command `PolynomialWithGaloisGroup`.

4.3.5 Distribution of Galois groups

So for the usage of our algorithm we will assume that every transitive group can occur as a Galois group. But what is known about the distribution of the Galois groups over the irreducible polynomials?

Well, in [14] it is proved that almost every irreducible polynomial with integer coefficients has its Galois group equal to the full symmetric group. There are elements with cycle type (n) in S_n, so we can always find a certificate in this case. This is what happens in the article:

Definition 4.7 *By $E_n(N)$ we denote the number of monic polynomials in $\mathbb{Z}[X]$ of degree n with all coefficients in absolute value $\leq N$ that do* not *have S_n as their Galois group.*

Definition 4.8 *By $R_n(N)$ we denote the number of reducible monic polynomials in $\mathbb{Z}[X]$ of degree n with all coefficients in absolute value $\leq N$.*

$R_n(N)$ can be estimated by $\ll N^{n-1}$ and in the article cited above it is proved that $E_n(N) \ll N^{n-\frac{1}{2}}(\log N)^{1-\gamma}$ where $\gamma = \gamma_n > 0$.

And so we see that almost every polynomial which does not have S_n as its Galois group, is reducible. In fact we have that the set of the irreducible polynomials that do not have the full symmetric group as Galois group has density zero in the set of all irreducible polynomials.

Having the full symmetric group as Galois group is not a necessary condition for finding certificates, so we can can conclude that almost every irreducible polynomial (namely the ones with Galois group S_n or another group that guarantees certificates) has a certificate.

Chapter 5

Modulo p: Improved

In the previous chapter we have seen that there are irreducible polynomials f for which we can not find a certificate of the form 3.28 with the modulo p-algorithm.
In this chapter we shall give an algorithm that will provide a certificate in these cases. It is based on the modulo-factorisations that have been computed.

5.1 The Hensel-algorithm

Suppose we have a polynomial f of which we think that it is irreducible (because that is what Magma tells us for example). Then we start looking for a certificate with the modulo p-algorithm. After trying some primes we still have not found a certificate. But at that point we do have some information: all the modulo-factorisations. An example to illustrate this and to show what to do now:

Example Let $f = X^6+8X^4+11X^2-4$. In the previous chapter we have seen that there is no certificate for f. The factorisation of f modulo $p = 3$ is: $\overline{f} = (X^3+X^2+2)(X^3+2X^2+1) \in \mathbb{F}_p[X]$, so we know that if f is reducible then it has a factor of degree 3. We know even more: if f is reducible, say $f = gh$ then we can say (without loss of generality) that $\overline{g} = X^3 + X^2 + 2$ and $\overline{h} = X^3 + 2X^2 + 1 \in \mathbb{F}_p[X]$.

Remark As the discriminant of f is $2^{12}31^4$, 3 does not divide it. We can not use a prime here that divides the discriminant, because if $p \mid \mathrm{disc}(f)$ we would not have a squarefree factorisation of f modulo p. The guarantee of such a squarefree factorisation is needed for Hensel-lifting the factorisation. This will be treated later in this section.

We can now use a bound on the coefficients of possible factors by Mignotte [22]. To state a theorem based on this bound [8, Section 3.5.1], we first need a definition:

Definition 5.1 *For f we define $|f| = (\sum_{i=0}^{n} |f_i|^2)^{1/2}$.*

So in our case $|f| = \sqrt{1 + 64 + 121 + 16} = \sqrt{202}$.

The theorem that bounds the coefficients of a polynomial that divides f is the following:

Theorem 5.2 *If $g = X^k + \ldots + g_1X + g_0 \in \mathbb{Z}[X]$ divides f then we have for all j that*

$$|g_j| \leq \binom{k-1}{j}|f| + \binom{k-1}{j-1}|f_n|.$$

So in our case we have a possible monic factor of degree 3, say $g = X^3 + g_2X^2 + g_1X + g_0$. Then the theorem tells us that:

$$\begin{aligned} |g_2| &\leq \binom{2}{2}|f| + \binom{2}{1} < 17 \\ |g_1| &\leq \binom{2}{1}|f| + \binom{2}{0} < 30 \\ |g_0| &\leq |f| + 1 < 16 \end{aligned}$$

So the coefficients of g must all be < 30 in absolute value. We call this bound on the maximum of the absolute values of the coefficient the *Mignotte bound* and denote it by B. What we do now is Hensel-lift our factorisation in $\mathbb{F}_p[X]$ to a power of p that is greater than $2B$. (More information on Hensel-lifting, and the proof of uniqueness can be found in [15, Section 15.4].) We now know that $g = \overline{g}$, as all coefficients must be less than B, in absolute value.

In our case we lift the factorisation into $\mathbb{F}_{81}[X]$. The factorisation then becomes: $\overline{f} = (X^3 - 14X^2 + 21X + 2)(X^3 + 14X^2 + 21X - 2)$. And now we know that *if* f is reducible then its factors have to be in this factorisation. But we can easily see that these factors do not divide f and so we are certain that f is irreducible!

In the unlikely case that we would have found a factor (for instance because of a bug in the system), we now would have known that f is reducible! So this algorithm actually prevents us from errors.

5.2 The Hensel-certificate

Certificate 5.3 $q \in \mathbb{N}$

Criterion 5.4 *There exists a prime power q that is greater than two times the Mignotte bound, and such that all factors of f modulo q do not divide f.*

The proof that this criterion guarantees irreducibility of f has been given above.

5.3 The Improved Algorithm: Combining 3.28 and 5.3

Let $\deg(f) = n$. To find one of these two certificates for f we do the following:

- If n is prime then we will find a modulo p-certificate, so we start searching and know by the theory of Chapter 4 that we will find such a certificate.
- Otherwise n is composite and this will be our algorithm:
  1. Try to find a modulo p-certificate, with the primes $\leq 30 \cdot n$, while keeping the modulo-factorisations in memory. The bound has been found by testing.
  2. If we have a certificate now, we are done. Otherwise we pick out a prime for which f has the fewest factors modulo p.
  3. Use p and the factorisation of f modulo p to find a Hensel-certificate (which can always be found). And we are done.

We now have an algorithm that can output two kinds of certificates. But we can easily decide which of the two we have, as will be discussed next.

5.4 Checking

If the certificate consists of more than one number we know that it is a modulo p-certificate (3.28). Otherwise, if the one number is not prime we know that we have a Hensel-certificate (5.3) and if the number is prime we compute the factorisation of f modulo it. If $\overline{f}$ is irreducible we are done and otherwise we know we have a Hensel-certificate.

If our modulo p-algorithm works, we now have an upper bound on the prime numbers, so the checking (3.3.3) can be done really quick now.
Otherwise checking the 'Hensel'-certificate means:

1. Compute f's Mignotte bound B and check that $q > 2B$.
2. Compute the factorisation F of f modulo q.
3. Check the irreducibility of the factors found, over $\mathbb{F}_p$, with proposition 3.23.
4. Check that all possible factors are not factors.

In fact we also need to check that we have tried *all* possible factors, but this should be very clear from the algorithm used.

5.5 Worst Case Scenario

We now know that we most of the time will get a modulo p-certificate, and else we probably have a modulo-factorisation with not so many factors. But the Swinnerton-Dyer polynomials are the worst polynomials that can be encountered: as we have seen the factors of their factorisation modulo a prime always have degree ≤ 2. And then we have a lot of factors to check.

Example Let f be the 4th Swinnerton-Dyer polynomial, which is of degree $2^4 = 16$. Suppose f splits in quadratic factors modulo some prime. Then we have 8 factors that we Hensel-lift. Then we first have to check if any of these lifted factors divide f, next all combinations of 2 and 3 factors and then half of all possible combinations of 4 factors (this is because if we have checked that such a combination is not a factor, the complementary combination can also not be a factor!). In total these are $\binom{8}{1} + \binom{8}{2} + \binom{8}{3} + \frac{1}{2}\binom{8}{4} = 8+28+56+35 = 127$ possible factors, and it is clear that for larger Swinnerton-Dyer polynomials this number grows exponentially in the degree of f. For the 5th Swinnerton-Dyer polynomial we have to check $32,767$ possible factors, for example.

Chapter 6

Test Results

Before we give the test results, we first state some remarks.

6.1 Remarks

Polynomials we have to Certify

Remark We only need certificates for monic polynomials, because we can check irreducibility of a non-monic polynomial by that of a monic one.

> **Proof** Let $f = f_nX^n + f_{n-1}X^{n-1} + \ldots + f_1X + f_0 \in \mathbb{Z}[X]$, with $n \geq 2,\ f_n \neq 0$ and $\gcd(f_0, \ldots, f_n) = 1$. Suppose that f is reducible, say $f = gh$, with $\deg(g) = k \geq 1, \deg(h) = n - k \geq 1$.
> **Claim:** The monic polynomial $f_n^{n-1}f(\frac{X}{f_n})$ is reducible.
> **Proof** $f_n = g_kh_{n-k}$, so $f_n^{n-1}f(\frac{X}{f_n}) = g_kh_{n-k}f_n^{n-2}g(\frac{X}{f_n})h(\frac{X}{f_n}) = (f_n^{k-1}h_{n-k}(g(\frac{X}{f_n})))\ (f_n^{n-k-1}g_k(h(\frac{X}{f_n})))$. □
> So the irreducibility of this monic polynomial proves the irreducibility of f. □

The Modulo p-certificate

In Chapter 3 we have seen Eisenstein's criterion and we have seen that for this criterion it sometimes works to look at $f(X + a)$ for some $a \in \mathbb{Z}$. This will not work for the modulo p-algorithm, as the Galois groups corresponding to $f(X)$ and $f(X+a)$ are the same. So:

Remark If there does not exist a certificate for $f(X)$ then neither does there exist one for $f(X + a)$.

Now let $n \in \mathbb{N}^*$, $f = X^2 + n!$. Then f is irreducible in $\mathbb{Z}[X]$, as $\sqrt{-n!} \notin \mathbb{Z}$. But we also see that for $p \leq n : \overline{f} = X^2 \in \mathbb{F}_p[X]$, so a certificate can only be found if $p > n$. With the theory of Chapter 4 we know that such a certificate will be found. And so we can conclude with the final remark:

Remark There is no $m \in \mathbb{N}$ such that all modulo p-certificates contain only primes $< m$.

6.2 Testing

For this test I have implemented the following certificates in Magma:
The improved modulo p-certificate (3.28 combined with 5.3).
Two certificates based on Bunyakovsky's conjecture: certificate 3.12
and Brillhart's certificate (3.15).
The extended certificate of Eisenstein (3.4).

Some of the following polynomials are said to have *random coefficients*. By this I mean: Let f be of degree n. Then the coefficient of the term of degree i is a random number which is in absolute value $\leq 10^{n+1-i}$.

All tested polynomials are monic (because of the first remark in this chapter), and denounced irreducible by Magma.

We try to find certificates for the following groups of polynomials:
(A3) All monic polynomials of degree 3 with coefficients in $[-5, \ldots, 5]$.
(R4) A set of 100 polynomials of degree 4 with random coefficients.
(S8) A set of 100 polynomials of degree 8 with coefficients in $[-100, \ldots, 100]$ (random).
(R8) A set of 100 polynomials of degree 8 with random coefficients.
(R17) A set of 100 polynomials of degree 17 with random coefficients.
(S100) A set of 10 polynomials of degree 100 with coefficients in $[-100, \ldots, 100]$ (random).
(R100) A set of 10 polynomials of degree 100 with random coefficients.

We will now look at the performance of the algorithms on all these sets of polynomials.

6.2.1 Modulo p-algorithm

The test results are in the following table. By the average number of primes we mean the average number of primes in a certificate. The smaller the primes in a certificate are, the easier we can check this certificate. So we also give the largest prime that occurs in the certificates, as well as the average (rounded of to one decimal) of all the primes occuring.

Set of polynomials	A3	R4	S8	R8	R17	S100	R100
Average number of primes	1	1.3	1.9	2.0	2.8	3.6	3.7
Largest prime	41	29	37	31	43	31	13
Average of all primes	3.9	5.3	5.5	5.7	5.9	5.4	4.9

For every polynomial we found a modulo p-certificate, so no Hensel-lifting was needed. The primes are all very small (≤ 43), so the certificates can be checked easy. This also means that we can find these certificates very fast.

Conclusion

The modulo p-algorithm has a good performance on all sets of polynomials. Certificates are found quickly and can be checked easily.

6.2.2 Algorithms based on Bunyakovsky's Conjecture

Brillhart's certificate 3.15 is in this test always easier to check than certificate 3.12. This is because for every tested polynomial f, the certificate 3.12 contains at least one prime evaluation that is greater than the prime evaluation in Brillhart's certificate.

Because of its better performance we only give the test results for the algorithm that assigns Brillhart's certificate (if this is possible).

For all polynomials of degree ≤ 17 such a certificate is found. As a reminder: a certificate now consists of a number $x_0 \in \mathbb{Z}$. The smaller the absolute value of this x_0 is, the sooner we find it. These are the test results on the x_0's found:

Set of polynomials	A3	R4	S8	R8	R17
Largest x_0 in absolute value	20	283	204	202	502
Average absolute value of x_0	5.5	61	68	69	96

The primes that we find seem to grow exponentially in the degree of the polynomial, as we can see in the test results on the primes:

Set	Average of all primes	Largest prime
A3	246	6347
R4	115322138	5629706083
S8	80602014447841089	4404810068836754723
R8	37703978841734883	1893697149435378341
R17	115850744758137640950336926116401219582360968	a 46-digit prime
S100	–	a 238-digit prime

We have used our algorithm for one polynomial of degree 100, and it took more than 1000 seconds of processor time in Magma to find the 238-digit prime. So for large polynomials this certificate is not practical to find or check.

Conclusion

For polynomials of low degree certificates are found quickly. But already for polynomials of degree 4 the primes are quite large. So *if* we trust the primality check in Magma then this algorithm gives good certificates in a fast way for not too large polynomials.

6.2.3 Eisenstein

For the set A3 we find a certificate for 266 of the 1002 polynomials (taking $|a| \leq 100$). For the set R4 we find a certificate for 3 out of 100 polynomials (also taking $|a| \leq 100$). For all other sets we do not find certificates, even for large a. For large a we have the disadvantage that the coefficients of $f(X + a)$ become huge, so that trying to find a criterion takes a lot of processor time.

Conclusion

If a polynomial is in the right form, then the Eisenstein-criterion works very well. But for larger polynomials, it is not practical to try to find a certificate, as there are too few certificates to be found in this way.

6.2.4 A Swinnerton-Dyer Polynomial

For all polynomials that were tested a modulo p-certificate could be found. So for these polynomials we have not needed the Hensel-part of our algorithm. To check the performance of the Hensel-part we have tried the algorithms on the 4th Swinnerton-Dyer polynomial (of degree 16), denoted here by f.

With Eisenstein, for $|a| \leq 100$, no certificate can be found.

Brillhart's certificate is found in 0.4 seconds processor time. It consists of $x_0 = 108$. $f(108) = 338615958064361660344430273649169$, a 33-digit prime number.

When we use the modulo p-method we do not find a suitable combination for $p < 480$, so we start Hensel-lifting and find $q = 7^{11}$ as our Hensel-certificate. This is computed in 2.08 seconds processor time.

So for this f a Brillhart-certificate is found faster, but to check it will take longer than checking our Hensel-certificate. So even in this case we prefer the modulo p-method.

6.2.5 Final Conclusion

For low degree the certificates of Eisenstein or Brillhart can be faster, and in this case the Brillhart-certificate can be checked fast (as the average prime number found is low). *If* an Eisenstein certificate is found, it can always be checked fast. But the modulo p-method also gives good certificates quickly.

For the modulo p-algorithm we see from our tests that the prime numbers found in these certificates are in general very low (all were ≤ 43), the number of primes in a certificate increases slightly for higher degrees. So the conclusion is that this is a very good method, that almost always gives easy-to-check certificates.

Chapter 7

Formalizing Mathematics

In this chapter I will introduce the notion of *formalizing mathematics*. A program for doing this on a computer, the proof assistant Coq [9], will be used to give examples. For this thesis I have formally proved the correctness of theorem 3.22 in Coq, which will be discussed next. Eventually, I will say something about combining the Magma and Coq systems. Such a combination of systems would allow us to rigorously prove irreducibility of polynomials using a computer.

7.1 Introduction

7.1.1 What is formalizing?

Normally, mathematics (like the theorems proved in this thesis) is *informal*. This means that it is written in a sort of natural human language, and that not every small detail of a proof is completely worked out. We can *formalize* mathematics by formulating everything in a logical system. A proof then only consists of the reasoning steps allowed by this system, and it has to be complete, in order to *be* a proof.

The process of checking a proof can be automated if it is formal: all allowed reasoning steps are known and so a proof can be checked step by step. A program in which mathematics is formalized is called a *Proof Assistant (PA)*. In a PA the human user enters the mathematical notions, theorems and proofs, while the program checks if all details are correct. This can be compared to a word processor in which a spell checker is used to check the correctness of the single words.

For checking large pieces of software and hardware these PAs are used in the industry. A problem for mathematicians with these industrial programs is that they contain a large kernel of axioms and deduction rules. Hence there is a reasonable chance of having conflicting axioms, which has actually happened in such systems. So if we have checked software we can be quite sure of its correctness, but not completely, because of possible errors in the checking software. The great advantage of such a system is its great performance, which is of course needed in the industry.

We, as theoretical mathematicians, are of course more interested in the correctness of our proofs. For this purpose PAs with a small proof kernel are used. Some years ago the practice was to create complete formalized proofs (by hand) and then let the PA check it. Most recent systems are interactive, in the following sense: the user states a lemma, and proves it step by step. The PA checks all these steps.

We can be sure of the correctness of a PA such as Coq, as its proof kernel is small. This is called the *De Bruijn criterion for proof checkers*, called after prof. de Bruijn who was one of the first to develop a proof checker for mathematics on a computer in the 1960s, the Automath system [23].

Of course a very fundamental problem is that we can never be totally sure of anything, but if we accept the (few) simple rules in the proof kernel, then we know that all formalized proofs are correct, as they consist only of cases of these rules. In Coq, the program I have used, this kernel consists of deduction rules for a so called type theory: a formal language in which mathematics can be expressed.

I shall finish this subsection by stating advantages of formalizing mathematics (using a PA):

- The proofs are correct.
- The mathematics already formalized has a high accessibility. Therefore it is practical to formalize more mathematical theory by building on what has been formalized.
- It is a good way of sharing mathematics. There are also programs to represent the formalized mathematics in a nice way.

I will give examples of formalizing mathematics in Coq in the next section, but first we look at formalizing in a wider perspective: Computer Mathematics.

7.1.2 Computer Mathematics

With *Computer Mathematics*, we mean the activities performed on a computer by mathematicians, these are:

Presentation: This thesis, for instance, is typed in LaTeX. The typesetting of mathematics and mathematical symbols is an important activity that is done with a computer.

Computation: The computer can be used as a very powerful calculator, but in the field of *Computer Algebra (CA)* the computer is also used to perform symbolic computations. This means that we can for example calculate in an exact way with algebraic numbers. For instance, $\sqrt{2}$ is kept in memory by its minimal polynomial, $X^2 - 2$, and by doing so we obtain exact answers, we do not round off.

Proving: Formalizing mathematics. We have discussed this in the previous subsection.

The very longterm goal of the field of Computer Mathematics is to create the *ideal mathematical workspace*. This is a computer system in which mathematicians can perform all of their mathematical work. That includes defining new notions, perform (symbolic) computations, proving and publishing. In this workspace it would be as easy as doing mathematics on paper, but then with all the help of a computer.

If I for instance would write this thesis in such a workspace, I could compute and prove in the document that I am writing. It is obvious that such a workspace has not yet been realized, but it is clear that such a system has advantages for mathematicians.

7.2 Coq and a Criterion

7.2.1 Introducting Coq with Examples

All Coq-commands are typed in '`this style`'. In Coq there is a universe for datatypes and a universe for propositions. We start with some propositional logic:

Let A and B be propositions. This is denoted by '`Variables A B : Prop.`' in Coq. Then we can state the following lemma:

'`Lemma triv1 : (A /\ B) -> B.`' When we load this lemma the interactive proving-session starts and we are in the following environment, where we have to find a proof of $(A \wedge B) \rightarrow B$:

```
1 subgoal
A : Prop
B : Prop
-----------
(A /\ B) -> B
```

So the goal is under the line here, and the variables are above the line. We now need a proof of $(A \wedge B) \rightarrow B$, to do this we can give a proof of B using a proof of $A \wedge B$. This is done with the '`intro`' command. We want to give this proof of $A \wedge B$ the name HAB (as it becomes a **H**ypothesis in the proof), so we give the command '`intro HAB.`' and get:

```
1 subgoal
A : Prop
B : Prop
HAB : A /\ B
-----------
B
```

We want to use the proof HAB to obtain a proof of A and a proof of B. To do so we eliminate HAB by ‘`elim HAB.`’. By doing this we still have the proof of $A \wedge B$; we now get a proof of A and a proof of B *as well*. So we have:

```
1 subgoal
A : Prop
B : Prop
HAB : A ∧ B
-----------
A -> B -> B
```

Now we can obtain the proofs HA for A and HB for B by ‘`intros HA HB.`’:

```
1 subgoal
A : Prop
B : Prop
HAB : A ∧ B
HA : A
HB : B
-----------
B
```

Now we have a proof of B, namely HB and this is what we need. So we can state ‘`exact HB`’ and the proof is completed.

During the interactive process we have gone through, Coq has created a complete proof object for our lemma in the type theory-language. The check of this proof can now be done by the small type-checking part, and this is done by giving the command ‘`Qed.`’. The proof object we have can now be checked by any type-checking program, so anyone interested in the correctness of our proof can test it with his or her favorite type-checking program.

The lemma we have just proved is trivial to us, but it took some steps to obtain a formal proof. As mathematicians, we really want Coq to be able to figure out such a simple, straight-forward proof by itself, by trying some rules.
For this purpose *tactics* have been written. They combine several commands and try to find a proof. In the case of our lemma using the simple tactic ‘`trivial.`’ finds a proof at once.

We shall now first define some mathematical objects, functions and properties of objects in Coq. This is done to give an impression of what is possible in Coq, and to show that the language is quite readable to mathematicians.

We have the set `nat` of natural numbers in Coq. They are defined inductively as follows:

```
Inductive nat : Set :=
| O : nat
| S : nat -> nat.
```

So '`nat`' consists of elements of the form '`(S (S ...(S O)...))`'. As we have given no meaning to members of the set '`nat`', every element is automatically different from any other. This implies that there are infinitely many different elements in '`nat`'. The definition is inductive, hence *all* elements of '`nat`' are of the form '`(S (S ...(S O)...))`'.
We can think of '`O`' as the number zero and of '`S`' as the successor function.
Then '`(S (S (S O)))`' represents the number three for example.

Next functions on '`nat`' are defined in Coq. Addititition and multiplication can for instance be defined in a recursive way. The notation is improved as well, so that Coq can represent '`(S (S (S (S O))))`' by '`4`', for example. We take a look at some definitions based on '`nat`':

The property that `n` in `nat` is even can be defined in the following natural way:

```
Definition IsEven (n:nat) := exists k:nat, n = 2*k.
```

We can now prove `(IsEven 6)` by explicitly giving $k = 3$. To have a better check for even numbers we could also prove formally that the definition is equivalent with: the last decimal digit is even. This would require defining more mathematical notions, but it would afterwards be easy to prove a number even.

This subsection is concluded with the definition of a function: $n \mapsto 2n$:

```
Definition Double (n:nat):nat := 2*n.
```

By the second `:nat` we mean that the outcome (`2*n`) is in `nat`. If this would not be clear to Coq we would have to prove this to finish the definition.

7.2.2 Formalizing Theorem 3.22

My objective was to formally prove theorem 3.22:

If f is irreducible in $\mathbb{F}_p[X]$, then f is irreducible in $\mathbb{Z}[X]$.

To prove this the mathematical notions that are needed have to be formalized first. These are: finite fields of prime characteristic ($\mathbb{F}_p$), polynomials and some definitions like '`(irreducible f)`'.

For the formalisation I have used the C-CoRN-library[1] [10, 12] for Coq, developed by the Foundations Group of Computer Science of the University of Nijmegen.
Polynomials have been formalized in C-CoRN. Rings have been formalized as well and the set of all rings is denoted by '`Ring`'. For a ring R ('`Variable R : Ring.`') we define a polynomial with coefficients in R as:

```
Inductive poly : Type :=
| poly_zero : poly
| poly_linear : R -> poly -> poly.
```

So we have the zero polynomial ('`poly_zero`') and polynomials of the form $c + X * f$ ('`(poly_linear c f)`') with c in R, and f a polynomial. In C-CoRN, there are also lemmas for working with polynomials, and definitions of degrees and coefficients for example.
The notion of a finite field of prime characteristic, $\mathbb{F}_p$, has been formalized by Vince Barany. In his files he defines the notions of the greatest common divisor, primality of integers, and modular arithmetic and ultimately proves that $\mathbb{F}_p$ is a field for every prime p. He also states and proves lots of lemma's. These files are now online on the C-CoRN-website [12].

Polynomials and $\mathbb{F}_p$ thus have been formalized and can be used. For R, we have the formal polynomial ring over R by `(poly_ring R)`.

We now define '`Variable p : positive.`', a positive integer, and state
'`Hypothesis Hpprime : (IsPrime p).`', so that p is prime.
Now we can define $\mathbb{F}_p$, $\mathbb{F}_p[X]$ and $\mathbb{Z}[X]$ by

```
Definition fp := (Fp p Hprime).
Definition fpx := (poly_ring fp).
Definition zx := (poly_ring Z_as_Ring).
```

Lots of lemmas had to be proved, an example:

```
              Lemma fpx_resp_coef : forall (f:zx)(n:nat),
        (zfp (nth_coeff n f)) [=] (nth_coeff n (zxfpx f)).
```

This lemma states that for $f \in \mathbb{Z}[X]$, the n-th coefficient modulo p of f equals the n-th coefficient of *f modulo p*. Because much has been proved already, the proof is quite short:

```
induction f.
intuition.
induction n.
intuition.
astepl (zfp (nth_coeff n f)).
astepr (nth_coeff n (zxfpx f)).
apply (IHf n).
Qed.
```

[1] Most definitions in C-CoRN start with a '`c`', for reasons that are not of interest here. For readibility we drop this '`c`' in the definitions in this chapter.

We use induction on f and n and later use the induction hypothesis for f, `IHf`. The `astepl` and `astepr` commands rewrite the left and right side of an equation, respectively.

And finally the theorem was stated and proved:

```
                Theorem irrcrit : forall f:zx,
(irreducible fp (zxfpx f)) -> (irreducible _ f).
```

The complete formalisation is available online at
`http://www.math.ru.nl/~bosma/students/kirkels`.

7.3 Combination with Magma

As we have seen, formalized proofs are certainly correct. But computation in a PA is very slow. When we multiply in `nat` for example, we have to make a lot of calculation steps as all these natural numbers are of the form '`(S (S ...(S 0)...))`'. On the other hand we have *Computer Algebra Systems (CASs)*, such as Magma, that can compute very fast, but that may contain bugs. So it would be very nice to be able to combine these two types of systems, in our case Coq and Magma. This combination is possible in several ways:

- In subsection 7.1.2 we have already mentioned the ideal mathematical workspace. In such a workspace both systems are embedded and equally important.
- We could also build in a small theorem prover in Magma, for proving the correctness of algorithms.
- And finally we can use Magma to help Coq with its proofs.

These last two combinations are, of course, easier to realize than the workspace. And as there will probably always be specialized CASs for specific areas of mathematics, the last kind of combination will always be of interest.

For us this last combination gives a way to formally prove irreducibility, using certificates:

Suppose that we want to prove `(irreducible f)` in Coq, for f in $\mathbb{Z}[X]$. We have seen that certificates can help us. But finding a certificate is difficult, especially in a PA. So this is why we want Magma to find it. In Coq we have to prove the theorem: *'If there exists a certificate, then f is irreducible.'*, so that if Magma gives us a certificate, Coq only has to verify the correctness of this certificate.

An example: we have proved theorem 3.22. So to prove the irreducibility of f over $\mathbb{Z}$ it suffices to prove that $\overline{f}$ is irreducible over $\mathbb{F}_p$ for some prime p. Magma can help us find this p, so that we only have to check that p really is prime and that $\overline{f} \in \mathbb{F}_p[X]$ is irreducible. For this last fact a formalisation of proposition 3.23 can be used. Now Magma can again be of assistance by verifying the conditions of that proposition. For example: we have to check that $\overline{f} \mid X^{p^n} - X$. To help us Magma can perform the division and return $\overline{g}$ such that $\overline{f} \cdot \overline{g} = X^{p^n} - X$. Now Coq can check this multiplication (this is easy) and in this way it is proved that $\overline{f}$ indeed divides $X^{p^n} - X$.

And so we see that to prove irreducibility with certificate 3.20 we also need a formalisation of proposition 3.23, and a program that can communicate the outcome of a computation in Magma to Coq.

The program Maplemode [11] can be used to let Maple compute for Coq. Bas Spitters and I have made some adaptations to Maplemode, with the help of Dan Synek, such that we could let Magma do some computations for Coq. The factorizing of polynomials is not yet implemented, as more theory has to be formalized for this.

So the combination of Magma and Coq has been initiated, but there is more work to do before we can formally prove irreducibility in Coq, using computations in Magma.

7.4 Conclusions

The proof assistant Coq was introduced, and used to formally prove theorem 3.22. I have also shown how Coq can be used to formally prove the irreducibility of polynomials in $\mathbb{Z}[X]$, using certificates and the computer algebra system Magma for finding certificates.

At this moment, formalizing mathematics is practiced by few mathematicians. One of the reason is that formalizing is not a ‘mathematician-friendly’ activity. Some problems are:

- The formalized proofs must be given in full detail. Tactics have improved this, but it still is a long road from an informal to a formal proof.
- The mathematics that has been formalized is not very advanced. There are not many areas of mathematics of which recent results can be easily formalized. In the C-CoRN-project a constructive proof of the fundamental theorem of algebra (FTA) has been formalized, but it took a full year to formalize all mathematics involved and prove everything.
- Formalizing is still disconnected from the other Computer Mathematics-activities. The ideal workspace has not been realized, but there is more and more interaction between different systems.

Combining different systems is the pioneering work for the workspace, so it will be of use to formalize more certificates and to actually prove the irreducibility of (large) polynomials using a Proof Assistant.

Chapter 8

Samenvatting

Deze scriptie heeft als titel 'Irreducibiliteitscertificaten voor veeltermen met gehele coëfficienten'. Wat hiermee bedoeld wordt vertel ik zo, maar ik zal eerst een andere (meer wereldse) situatie beschrijven, die de sfeer van deze scriptie duidelijk zal maken.

Stel je voor dat je een lot koopt bij het postkantoor. De loting is een beetje raar, want de trekking blijft voor je verborgen. Na de trekking wil je weten of je iets gewonnen hebt en je gaat dus weer naar het postkantoor. Daar word je botweg verteld dat je geen prijs hebt. Tja, wat doe je dan? Aangezien er geen trekkingslijst is kun je moeilijk in protest gaan, maar je hebt geen enkele goede reden om te geloven dat je écht geen prijs hebt...

In deze scriptie nemen *veeltermen met gehele coëfficienten* (vanaf nu noem ik ze gewoon *veeltermen*) de plaats van de loten in. De vraag is of zo'n veelterm *reducibel* is, ofwel of je lot een prijs oplevert. Als je prijs hebt krijg je die en ben je tevreden. Net zo krijgen we voor een reducibele veelterm een *niet-triviale factor*.

Als een veelterm geen niet-triviale factoren heeft noemen we de veelterm *irreducibel*, niet reducibel dus. (Het komt dus overeen met een lot dat geen prijs oplevert.) De vraag is nu hoe we er zeker van kunnen zijn dat een veelterm irreducibel is. Bij de loten is het makkelijk: als je maar de trekkingslijst hebt, kun je nagaan of je een prijs hebt. Maar een probleem voor de veeltermen is, dat er daar oneindig veel van zijn, en daarvan zijn er ook weer oneindig veel reducibel en oneindig veel irreducibel. Dus een lijst opstellen van reducibele veeltermen heeft geen zin.

Wat er in deze scriptie eigenlijk gebeurt is dat ik naar speciale eigenschappen van irreducibele veeltermen ga kijken. Voorbeeldje bij de loten:

Stel dat je weet dat alle loten die in de prijzen zijn gevallen eindigen op 37. Als je nu een lot hebt dat eindigt op 13 heb je pech, maar je hebt dan wel de *zekerheid* dat je geen prijs hebt. Zo een speciale eigenschap, waarmee je zeker weet dat je geen prijs hebt (in dit geval het eindigen op 13) noemen we nu een *Geen-Prijs-Bewijs*, ofwel een *GPB*. Voor irreducibele veeltermen heet dit een *irreducibiliteitscertificaat.*

In deze scriptie gebeurt het volgende:

Eerst maak ik duidelijk wat ik precies bedoel met een irreducibiliteitscertificaat en geef er een paar voorbeelden van, ongeveer zoals hierboven, maar dan wat moeilijker en uitgebreider. Vervolgens ga ik één soort certificaat beter bestuderen, met behulp van Galoistheorie. In de lotenwereld was Galois degene die als eerste precies kon zeggen hoe de loten gemaakt worden en de loting tot stand komt. Zo kom ik er achter voor hoeveel loten, zonder prijs, we een GPB kunnen vinden. En ik vind ook grote groepen loten waarvoor ik zeker weet dat ik een GPB kan vinden als ze geen prijs hebben.

Daarna breid ik de GPBs die ik onderzocht heb uit, zodat ik voor élk lot dat niet in de prijzen valt een GPB kan vinden. Door uitgebreid te testen laat ik daarna zien dat deze GPBs eenvoudiger te controleren zijn dan een aantal andere soorten GPBs. De conclusie is dus dat ik een goede GPB heb gemaakt, dat voor elk lot zonder prijs gevonden kan worden.

Hoofdstuk 7 zal ik zonder loten proberen uit te leggen. Dat hoofdstuk gaat over het formaliseren van wiskunde in een computer. Hiermee bedoel ik, dat ik samen met een computer formele, correcte bewijzen ga geven. Ik heb van een irreducibiliteitscertificaat formeel bewezen, met een computer dus, dat het de irreducibiliteit van een veelterm impliceert. De systemen om wiskunde te formaliseren staan nog enigszins in de kinderschoenen, maar ik denk dat ze steeds vaker gebruikt zullen worden.

Nu, zoals beloofd, een uitleg van de titel. Een *veelterm* is een functie van de vorm $f_nX^n + f_{n-1}X^{n-1} + \ldots + f_1X + f_0$. Een voorbeeld is

$$X^4 + 3X^3 - 6X^2 + 8.$$

De *coëfficienten* zijn de getallen voor de verschillende X-machten, hier dus 1, 3, -6, 0 en 8. De *graad* is de hoogste macht van X, hier dus 4.

We bekijken veeltermen met *gehele* coëfficienten, dus $\frac{1}{2}X^2$ doet bijvoorbeeld niet mee. Je kunt met veeltermen ongeveer rekenen zoals met getallen, zo hebben we bijvoorbeeld $(X^2 + 7X - 6) + (3X^2 - 8X + 1) = 4X^2 - X - 5$ en $(X^2 + 2X + 3) \times (3X^2 + 5X - 7) = 3X^4 + 11X^3 + 12X^2 + X - 21$. Het rekenen met veeltermen wordt voor een deel op de middelbare school geleerd.

Met de vermenigvuldiging van net zien we dat $X^2 + 2X + 3$ een deler van $3X^4 + 11X^3 + 12X^2 + X - 21$ is. Zo'n deler noemen we nu een *factor*. We noemen een factor *niet-triviaal* als de graad 1 of hoger is, oftewel: als er een X in voorkomt. Als een veelterm geen niet-triviale factoren heeft, noemen we die veelterm *irreducibel* (dat betekent: niet te ontbinden). Ik hoop hiermee het onderwerp van mijn scriptie begrijpelijk uit te hebben gelegd.

Bibliography

[1] Tom M. Apostol. *Introduction to Analytic Number Theory.*
Springer-Verlag (1976)

[2] Wieb Bosma, John Cannon and Catherine Playoust. *The Magma algebra system. I. The user language*, in J. Symbolic Comp. 24, No. 3–4, pp. 235–265 (1997)
The Magma website: `http://magma.maths.usyd.edu.au/magma`

[3] Rolf Brandl. *Integer Polynomials that are Reducible Modulo all Primes*,
in The American Mathematical Monthly, Vol. 93, No. 4, pp. 286 – 288 (1986)

[4] John Brillhart. *Note on Irreducibility Testing*,
in Math. of Computation, Volume 35, No. 152, pp. 1379–1381 (1980)

[5] V. Bunyakovsky. *Sur les diviseurs numériques invariables des fonctions rationelles entières*, in Mem. Acad. Sci. St. Petersberg, Volume 6, pp. 305–329 (1857)

[6] David G. Cantor. *Irreducible Polynomials with Integral Coefficients Have Succinct Certificates*, in Journal of Algorithms, 2, pp. 385–392 (1981)

[7] Olga Caprotti and Martijn Oostdijk. *How to formally and efficiently prove prime(2999)*, in M. Kerber and M. Kohlhase, eds., Symbolic Computation and Automated Reasoning (Calculemus '00), A K Peters, Ltd., pp. 114–125 (2000)

[8] Henri Cohen. *A Course in Computational Algebraic Number Theory.*
Springer-Verlag (1993)

[9] The Coq Development Team. *The Coq Proof Assistent Reference Manual, Version 8.0*, available online: `http://pauillac.inria.fr/coq/doc/main.html`, ©INRIA (2004)

[10] Luís Cruz-Filipe, Herman Geuvers and Freek Wiedijk. *C-Corn: the Constructive Coq Repository at Nijmegen*, to appear in MKM 2004, Springer-Verlag (2004)

[11] David Delahaye and Micaela Mayero. *A Maple Mode for Coq*,
available online at `http://coq.inria.fr/contribs/MapleMode.html` (2002)

[12] Foundations Group, Computer Science, University of Nijmegen. *Constructive Coq Repository at Nijmegen*. Available online at `http://c-corn.cs.kun.nl`.

[13] F.G.M. Eisenstein. *Über die Irreductibilität und einige andere Eigenschaften der Gleichung, von welcher die Theilung der ganzen Lemniscate abhängt*,
in J. reine angew. Math. 39, pp. 166–167 (1850)

[14] P.X. Gallagher. *The Large Sieve and Probabilistic Galois Theory*, in Analytic Number Theory (Proc. Sympos. Pure Math., Vol. XXIV), pp. 91–101 (1973)

[15] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press (1999)

[16] D. Hilbert. *Über diophantische Gleichungen*, in Nachr. Ges. der Wissenschaften zu Göttingen, pp. 48–54 (1897)

[17] Mark van Hoeij. *Factoring polynomials and the knapsack problem*, in Journal of Number Theory, 95, pp. 167–189 (2002)

[18] Frans Keune. *Galoistheorie.* Syllabus, Department of Mathematics, University of Nijmegen (2000)

[19] Jürgen Klüners and Gunter Malle. *Explicit Galois realization of transitive groups of degree up to 15. Algorithmic methods in Galois theory*, in J. Symbolic Comp. 30, No. 6, pp. 675–716 (2000)

[20] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovàsz. *Factoring Polynomials with Rational Coefficients*, in Math. Ann. 261 (1982)

[21] Math pages, author unknown. `http://www.mathpages.com/home/kmath406.htm`

[22] Maurice Mignotte. *An inequality about factors of polynomials*, in Math. Comp. 28, pp. 1153–1157 (1974)

[23] R.P. Nederpelt, J.H. Geuvers and R.C. de Vrijer, eds. *Selected Papers on Automath*, Vol. 133 of Studies in Logic and the Foundations of Mathematics, Elsevier (1994) The Automath Archive is available online at `http://www.win.tue.nl/automath`.

[24] H.C. Pocklington. *The Determination of the Prime or Composite Nature of Large Numbers by Fermat's Theorem*, in Proc. Cambridge Phil. Soc. 18, pp. 29–30, (1914)

[25] V.R. Pratt. *Every prime has a succinct certificate*, in SIAM J. Comput. Vol. 4, pp. 214–220 (1974)

[26] T. Schönemann. *Grundzüge einer allgemeinen Theorie der höhern Congruenzen, deren Modul eine reelle Primzahl ist*, in J. reine angew. Math. 31, pp. 269–325 (1846)

[27] P. Stevenhagen and H.W. Lenstra, Jr. *Chebotarëv and his Density Theorem*, in The Mathematical Intelligencer, Vol. 18, No. 2, pp. 26–37 (1996)

[28] P.J. Weinberger. *Finding the Number of Factors of a Polynomial*, in Journal of Algorithms, 5, pp. 180–186 (1984)