5. Factorization

In the previous chapters we have discussed methods for solving systems of polynomial or algebraic equations of the form

$$\begin{aligned} f_1(x_1,\ldots,x_n) &= 0 \;, \\ &\vdots \\ f_m(x_1,\ldots,x_n) &= 0 \;. \end{aligned}$$

Of course, if we can factor for instance the polynomial f_1 into

$$f_1 = f_1^{(1)} \cdot f_1^{(2)} \;,$$

the problem is reduced to solving the two simpler systems

$$\begin{aligned} f_1^{(1)}(x_1,\ldots,x_n) &= 0 \;, & \qquad f_1^{(2)}(x_1,\ldots,x_n) &= 0 \;, \\ &\vdots & &\vdots \\ f_m(x_1,\ldots,x_n) &= 0 \;, & f_m(x_1,\ldots,x_n) &= 0 \;. \end{aligned}$$

In this chapter we discuss factorization of univariate and also multivariate polynomials over various coefficient domains. Typically we will assume that the polynomial to be factored is squarefree. This can always be achieved by the method described in Section 3.3.

5.1 Factorization over finite fields

Similar to what we have done for the computation of gcds of polynomials, we will reduce the computation of the factors of an integral polynomial to the computation of the factors of the polynomial modulo a prime number. So we have to investigate this problem first, i.e. we consider the problem of factoring a polynomial $a(x) \in \mathbb{Z}_p[x]$, p a prime number. W.l.o.g. we may assume that $\mathrm{lc}(a) = 1$.

In the sequel we describe E.R. Berlekamp's algorithm for factoring squarefree univariate polynomials in $\mathbb{Z}_p[x]$. Throughout this section let $a(x)$ be a squarefree polynomial of degree n in $\mathbb{Z}_p[x]$, p a prime number, having the following factorization into irreducible factors

$$a(x) = \prod_{i=1}^{r} a_i(x) \;.$$

Applying the algorithm CRA of Section 3.2 not to integers but to polynomials, we get that for every choice of $s_1,\ldots,s_r \in \mathbb{Z}_p$ there exists a uniquely determined polynomial $v(x) \in \mathbb{Z}_p[x]$ such that

$$\begin{array}{l} v(x) \equiv s_i \bmod a_i(x) \quad \text{for } 1 \le i \le r, \\ \text{and} \\ \deg(v) < \deg(a_1) + \ldots + \deg(a_r) = n. \end{array} \tag{5.1.1}$$

This is proven in [Win96], Theorem 3.1.11. In (5.1.1) it is essential that a is squarefree, i.e. the a_i's are relatively prime.

Lemma 5.1.1. *For every a_i, a_j, $i \neq j$, there exist $s_1, \ldots, s_r \in \mathbb{Z}_p$ such that the corresponding solution $v(x)$ of* (5.1.1) *generates a factorization $b \cdot c$ of a with $a_i \mid b$ and $a_j \mid c$.*

Proof: If $r = 1$ there is nothing to prove. So assume $r \geq 2$. Choose $s_i \neq s_j$ and the other s_k's arbitrary. Let v be the corresponding solution of (5.1.1). Then

$$a_i(x) \mid \gcd(a(x), v(x) - s_i) \quad \text{and} \quad a_j(x) \not| \gcd(a(x), v(x) - s_i). \qquad \square$$

So we could solve the factorization problem over $\mathbb{Z}_p$, if we could get a complete overview of the solutions $v(x)$ of (5.1.1) for all the choices of $s_1, \ldots, s_r \in \mathbb{Z}_p$. Fortunately this can be achieved by linear algebra methods.

If $v(x)$ satisfies (5.1.1), then

$$v(x)^p \equiv s_i^p = s_i \equiv v(x) \bmod a_i(x) \quad \text{for } 1 \leq i \leq r.$$

So we have

$$v(x)^p \equiv v(x) \bmod a(x), \quad \text{and} \quad \deg(v) < n. \tag{5.1.2}$$

Every solution of (5.1.1) for some $s_1, \ldots, s_r$ solves (5.1.2).

But what about the converse of this implication? Is every solution of (5.1.2) also a solution of (5.1.1) for some $s_1, \ldots, s_r$? From the fact that $GF(p)$ is the splitting field of $x^p - x$, we get that

$$v(x)^p - v(x) = (v(x) - 0)(v(x) - 1) \ldots (v(x) - (p - 1)).$$

So if $v(x)$ satisfies (5.1.2), then $a(x)$ divides $v(x)^p - v(x)$ and therefore every irreducible factor $a_i(x)$ must divide one of the factors $v(x) - s$ of $v(x)^p - v(x)$. Thus, every solution of (5.1.2) is also a solution of (5.1.1) for some $s_1, \ldots, s_r$. In particular, there are exactly p^r solutions of (5.1.2).

Fermat's Little Theorem says: *if the prime number p does not divide the integer a, then* $a^{p-1} \equiv 1 \mod p$ (Theorem 2.5.1 in [Win96]). Moreover, in $\mathbb{Z}_p[x]$ we have

$$(a(x) + b(x))^p = a(x)^p + b(x)^p$$

(Theorem 2.5.2 in [Win96]). So the solutions of (5.1.2) constitute a vector space over $\mathbb{Z}_p$: for a, b solution of (5.1.2) and $m \in \mathbb{Z}_p$ we have

$$\begin{aligned} (a + b)^p &= a^p + b^p = a + b\ , \\ (m \cdot a)^p &= m^p a^p = m \cdot a\ . \end{aligned}$$

We can get a complete overview of the solutions of (5.1.2), if we can compute a basis for this vector space.

Let the $(n \times n)$–matrix $Q(a)$ over $\mathbb{Z}_p$,

$$Q(a) = Q = \begin{pmatrix} q_{0,0} & \cdots & q_{0,n-1} \\ \vdots & & \vdots \\ q_{n-1,0} & \cdots & q_{n-1,n-1} \end{pmatrix} ,$$

be defined by $x^{pk} \equiv q_{k,n-1}x^{n-1} + \ldots + q_{k,1}x + q_{k,0} \mod a(x)$ for $0 \leq k \leq n-1$.
I.e., the entries in the k–th row of Q are the coefficients of $x^{pk} \mod a(x)$. Using the representation of $v(x) = v_{n-1}x^{n-1} + \ldots + v_0$ as the vector $(v_0, \ldots, v_{n-1})$, we have

$$v \cdot Q = v$$
$$\Longleftrightarrow$$
$$v(x) = \sum_{j=0}^{n-1} v_j x^j = \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} v_k \cdot q_{k,j} x^j \equiv \sum_{k=0}^{n-1} v_k x^{pk} = v(x^p) = v(x)^p \mod a(x) .$$

We summarize all these results in the following theorem.

Theorem 5.1.2. *With the notation used above, a polynomial $v(x) = v_{n-1}x^{n-1} + \ldots + v_0$ in $\mathbb{Z}_p[x]$ solves (5.1.2) if and only if the vector $(v_0, \ldots, v_{n-1})$ is in the null–space of the matrix $Q - I$ (I the identity matrix of dimension n), i.e. $v \cdot (Q - I) = (0, \ldots, 0)$.* □

In a basis for the null-space of $Q - I$ we must find v's which split the factors, as in Lemma 5.1.1.

Now we are ready to formulate Berlekamp's algorithm for factoring squarefree univariate polynomials in $\mathbb{Z}_p[x]$.

```
FACTOR_B(Berlekamp factorization algorithm)
for a given prime number $p$
and a squarefree polynomial $a(x)$ in $\mathbb{Z}_p[x]$,
the list $F$ of prime factors of $a$ is determined.
(1) Form the $(n \times n)$–matrix $Q$ over $\mathbb{Z}_p$, where the $k$–th line $(q_{k,0}, \ldots, q_{k,n-1})$
    of $Q$ satisfies
    $x^{pk} \mod a(x) = q_{k,n-1}x^{n-1} + \ldots + q_{k,0}$, for $0 \leq k \leq n-1$ ;
(2) By column operations transform the matrix $Q - I$ into (e.g. lower–right)
    triangular form;
    From the triangular form read off the rank $n - r$ of the matrix $Q - I$;
    [There are exactly $r$ linearly independent solutions $v^{[1]}, \ldots, v^{[r]}$
    of $v \cdot (Q - I) = 0$.
    Let $v^{[1]}$ be the trivial solution $(1, 0, \ldots, 0)$.
    So (after interpretation of vectors as polynomials) there are
    $p^r$ solutions $t_1 \cdot v^{[1]} + \ldots + t_r \cdot v^{[r]}$ of (5.1.2),
    and therefore $r$ irreducible factors of $a(x)$]
(3) If $r = 1$, then $a(x)$ is irreducible and we set $F := [a]$;
    Otherwise, compute $\gcd(a(x), v^{[2]}(x) - s)$ for $s \in \mathbb{Z}_p$
    and put the factors of $a$ found in this way into the list $F$;
    As long as $F$ contains fewer than $r$ factors, choose the next
    $v^{[k]}(x)$, $k = 3, \ldots, r$, and compute $\gcd(f(x), v^{[k]}(x) - s)$ for $f$ in $F$;
    Add the factors found in this way to $F$;
    [Ultimately, $F$ will contain all the factors of $a(x)$.]   □
```

Example 5.1.3. Let us use FACTOR_B for factoring the polynomial

$$a(x) = x^5 + x^3 + 2x^2 + x + 2$$

in $\mathbb{Z}_3[x]$. First we have to check for squarefreeness. $a'(x) = 2x^4 + x + 1$, so $\gcd(a, a') = 1$ in $\mathbb{Z}_3[x]$ and therefore $a(x)$ is squarefree.

The rows of the (5×5)–matrix Q are the coefficients of $x^0, x^3, x^6, x^9, x^{12}$ modulo $a(x)$. So

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 1 & 2 \\ 0 & 1 & 1 & 2 & 2 \\ 2 & 0 & 2 & 1 & 1 \end{pmatrix} .$$

$Q - I$ can be transformed into the triangular form

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

We read off $r = 2$, i.e. there are 2 irreducible factors of $a(x)$. The null–space of $Q - I$ is spanned by

$$v^{[1]} = (1, 0, 0, 0, 0) \quad \text{and} \quad v^{[2]} = (0, 0, 2, 1, 0).$$

Now we get the factors by appropriate gcd computations:

$$\begin{aligned} \gcd(a(x), v^{[2]}(x) + 2) &= x^2 + x + 2, \\ \gcd(a(x), v^{[2]}(x) + 1) &= x^3 + 2x^2 + 1 \ . \end{aligned}$$

□

The basic operations in FACTOR_B are the setting up and solution of a system of linear equations and the gcd computations for determining the actual factors. The complexity of FACTOR_B is proportional to $n^3 + prn^2$, where n is the degree of the polynomial.

5.2 Factorization over the integers

Before developing algorithms for actually producing a factorization of a reducible polynomial, we might want to decide whether a given polynomial is in fact irreducible. A powerful criterion for irreducibility is due to Eisenstein, a proof can be found for instance in [vdW70].

Theorem 5.2.1. (Eisenstein's irreducibility criterion) *Let R be a ufd and $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ a primitive polynomial of positive degree n in $R[x]$. If there is an irreducible element p of R such that*
$(p \not| a_n, p|a_i$ for all $i < n$, and $p^2 \not| a_0)$ or $(p \not| a_0, p|a_i$ for all $i > 0$, and $p^2 \not| a_n)$,
then $f(x)$ is irreducible in $R[x]$. □

Univariate polynomials

According to the corollary to Gauss' lemma (Theorem 3.1.5) factorizations of univariate integral polynomials are essentially the same in $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$. For reasons of efficiency we concentrate on the case of integral polynomials. The factorization of integers is a much harder problem than the factorization of polynomials. For this reason we do not intend to factor the content of integral polynomials. Throughout this section we assume that the polynomial to be factored is a primitive non-constant polynomial.

The problem of factoring a primitive univariate integral polynomial $a(x)$ consists in finding pairwise relatively prime irreducible polynomials $a_i(x)$ and positive integers m_i such that

$$a(x) = \prod_{i=1}^{r} a_i(x)^{m_i}.$$

As for polynomials over finite fields we will first compute a squarefree factorization of $a(x)$. By application of SQFR_FACTOR our factorization problem is reduced to the problem of factoring a primitive squarefree polynomial. So from now on let us assume that $a(x)$ is primitive and squarefree.

As in the case of polynomial gcds we would like to use the fast factorization algorithm modulo a prime p. However, the approach of choosing several primes and combining the results by the Chinese remainder algorithm does not work for factorization. We do not know which of the factors modulo the different primes correspond to each other. So we choose a different approach. The problem of factorization over $\mathbb{Z}$ is reduced to factorization modulo p and a subsequent lifting of the result to a factorization modulo p^k. If k is high enough, the integer factors can be constructed.

Theorem 5.2.2. *Let $a(x) \in \mathbb{Z}[x]$ be primitive and squarefree. Let p be a prime number not dividing $\mathrm{lc}(a)$. Let $a_1(x), \ldots, a_r(x) \in \mathbb{Z}_p[x]$ be pairwise relatively prime such that $a \equiv a_1 \cdot \ldots \cdot a_r \bmod p$ and $\mathrm{lc}(a_1) = \mathrm{lc}(a) \bmod p$, $\mathrm{lc}(a_2) = \ldots = \mathrm{lc}(a_r) = 1$. Then for every natural number k there are polynomials $a_1^{(k)}(x), \ldots, a_r^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$ with $\mathrm{lc}(a_1^{(k)}) = \mathrm{lc}(a) \bmod p^k$, $\mathrm{lc}(a_2^{(k)}) = \ldots = \mathrm{lc}(a_r^{(k)}) = 1$ such that*

$$a(x) \equiv a_1^{(k)}(x) \cdot \ldots \cdot a_r^{(k)}(x) \bmod p^k \quad \text{and} \quad a_i^{(k)}(x) \equiv a_i(x) \bmod p \quad \text{for} \quad 1 \le i \le r\ .$$

Proof: We proceed by induction on k. For $k = 1$ we can obviously choose $a_i^{(1)} = a_i$ and all the requirements are satisfied.

So now assume that the $a_i^{(k)}$ satisfy the requirements. I.e. for some $\hat{d}(x) \in \mathbb{Z}_p[x]$ we have

$$a - \prod_{i=1}^{r} a_i^{(k)} \equiv p^k \hat{d} \mod p^{k+1}.$$

We replace the leading coefficient of $a_1^{(k)}$ by $(\mathrm{lc}(a) \bmod p^{k+1})$. Then for some $d(x) \in \mathbb{Z}_p[x]$ we have

$$a - \prod_{i=1}^{r} a_i^{(k)} \equiv p^k d \mod p^{k+1},$$

where $\deg(d) < \deg(a)$. We will determine $b_i(x) \in \mathbb{Z}_p[x]$ with $\deg(b_i) < \deg(a_i)$ such that

$$a_i^{(k+1)} = a_i^{(k)} + p^k b_i.$$

Using this ansatz, we get

$$a - \prod_{i=1}^{r} a_i^{(k+1)} \equiv \underbrace{a - \prod_{i=1}^{r} a_i^{(k)}}_{p^k d} - p^k \Big(\sum_{i=1}^{r} b_i \underbrace{\prod_{j=1, j \neq i}^{r} a_j}_{=:\tilde{a}_i} \Big) \bmod p^{k+1}.$$

So the $a_i^{(k+1)}$'s will constitute a factorization modulo p^{k+1} if and only if

$$d \equiv \sum_{i=1}^{r} b_i \cdot \tilde{a}_i \bmod p.$$

A solution is guaranteed by an appropriate generalization of the extended Euclidean algorithm (Theorem 3.1.2), applied to polynomials:

Fact: Let K be a field (e.g., $\mathbb{Z}_p$),
$a_1, \ldots, a_r \in K[x]$ pairwise relatively prime,
$c \in K[x]$ s.t. $\deg(c) < \deg(a_1 \cdots a_r)$.
Then there exist $u_1, \ldots, u_r \in K[x]$ with
$c = \sum_{i=1}^{r} (u_i \prod_{j=1, j \neq i}^{r} a_j)$ and $\deg(u_i) < \deg(a_i)$ for $1 \leq j \leq r$.

A proof of this fact can be found in [Win96], Theorem 3.1.2 and Exercise 11 in Section 3.1.

This completes the proof. □

The proof of Theorem 5.2.2 is completely constructive and can be turned into an algorithm LIFT_FACTORS such that:

$$\text{LIFT\_FACTORS}(a, [a_1, \ldots, a_r], p, k) = [a_1^{(k)}, \ldots, a_r^{(k)}] \; .$$

Putting these results together, we get the Berlekamp-Hensel algorithm FACTOR_BH for factoring primitive univariate squarefree polynomials over the integers.

```
**FACTOR_BH**(Berlekamp-Hensel factorization algorithm)
for a given primitive squarefree polynomial $a$ in $\mathbb{Z}[x]$,
determine the list of primitive irreducible factors $F = [a_1, \ldots, a_r]$ of $a$.
(1)  Choose a prime number $p$ such that $p \not| \mathrm{lc}(a)$ and $a$ is squarefree modulo $p$
     (i.e. $p$ does not divide the discriminant of $a$);
(2)  $[u_1, \ldots, u_s] :=$ FACTOR_B$(a, p)$;
     Normalize the $u_i$'s such that $\mathrm{lc}(u_1) = \mathrm{lc}(a) \bmod p$ and $\mathrm{lc}(u_2) = \ldots = \mathrm{lc}(u_s) = 1$;
(3)  Determine a natural number $B$ which bounds the absolute value of
     any coefficient in a factor of $a$ over the integers
     (for instance, use the Landau–Mignotte bound, Theorem 3.2.1);
     Choose $k$ such that $p^k \geq 2|\mathrm{lc}(a)|B$;
(4)  $[v_1, \ldots, v_s] :=$ LIFT_FACTORS$(a, [u_1, \ldots, u_s], p, k)$;
(5)  [combine factors]
     $\overline{a} := a$;
     $C := \{2, \ldots, s\}$;    [$v_1$ will be included in the last factor]
     $F := [\,]$ ;
     $m := 0$;
     **while** $m <| C |$ **do**
          $m := m + 1$;
          **forall** $\{i_1, \ldots, i_m\} \subseteq C$ **do**
               [integers modulo $p^k$ are centered around 0, i.e.
               the representation of $\mathbb{Z}_{p^k}$ is $\{q \mid -p^k/2 < q \leq p^k/2\}$]
               $\tilde{b} := (\mathrm{lc}(\overline{a}) \cdot v_{i_1} \cdot \ldots \cdot v_{i_m} \bmod p^K)$, interpreted as a polynomial in $\mathbb{Z}[x]$ ;
               $b :=$ primitive part of $\tilde{b}$ ;
               **if** $b \mid \overline{a}$ **then**
                    add $b$ to the list $F$ ;
                    $\overline{a} := \overline{a}/b$;
                    $C := C \setminus \{i_1, \ldots, i_m\}$ ;
     add $\overline{a}$ to the list $F$ ;     □
```

Step (5) is necessary, because irreducible factors over the integers might factor further modulo a prime p. In fact, there are irreducible polynomials over the integers which factor modulo every prime number. An example of this is $x^4 + 1$.

The complexity of FACTOR_BH would be polynomial in the size of the input except for step (5). Since in step (5), in the worst case, we have to consider all possible combinations of factors modulo p, this might lead to a combinatorial explosion, rendering the algorithm FACTOR_BH exponential in the size of the input. Nevertheless, in practical examples the combinations of factors does not present an insurmountable problem. Basically all the major computer algebra systems employ some variant of FACTOR_BH as the standard factoring algorithm for polynomials over the integers.

Example 5.2.3. We want to factor the primitive squarefree integral polynomial

$$a(x) = 6x^7 + 7x^6 + 4x^5 + x^4 + 6x^3 + 7x^2 + 4x + 1.$$

We use FACTOR_BH. A suitable prime is 5, $a(x)$ stays squarefree modulo 5.

By an application of the Berlekamp algorithm FACTOR_B, $a(x)$ is factored modulo 5 into

$$a(x) \equiv \underbrace{(x-2)}_{u_1} \cdot \underbrace{(x^2-2)}_{u_2} \cdot \underbrace{(x^2+2)}_{u_3} \cdot \underbrace{(x^2-x+2)}_{u_4} \bmod 5.$$

By an application of LIFT_FACTORS we lift this factorization to a factorization modulo 25, getting

$$a(x) \equiv \underbrace{(6x+3)}_{v_1} \cdot \underbrace{(x^2-7)}_{v_2} \cdot \underbrace{(x^2+7)}_{v_3} \cdot \underbrace{(x^2+9x-8)}_{v_4} \bmod 25.$$

The Landau–Mignotte bound for a is rather big. Let us assume that by some additional insight we know that $K = 2$ is good enough for constructing the integral factors. Now we have to try combinations of factors modulo 25 to get the factors over the integers. So we set $\overline{a} := a$ and $C := \{2, 3, 4\}$. Testing the factors v_2, v_3, v_4 we see that only v_4 yields a factor over the integers:

$$a_1(x) := \text{primitive part of}(\text{lc}(\overline{a}) \cdot v_4 \bmod 25) = 3x^2 + 2x + 1.$$

So now $\overline{a} := \overline{a}/a_1 = 2x^5 + x^4 + 2x + 1$. The combination of v_2 and v_3 yields the factor

$$a_2(x) := \text{primitive part of}(\text{lc}(\overline{a}) \cdot v_2 \cdot v_3 \bmod 25) = x^4 + 1.$$

We set $\overline{a} := \overline{a}/a_2 = 2x + 1$. Now C has become empty, and the last factor is

$$a_3(x) := \overline{a}(x) = 2x + 1.$$

FACTOR_BH returns $F = [a_1, a_2, a_3]$, i.e. the factorization

$$a(x) = (3x^2 + 2x + 1) \cdot (x^4 + 1) \cdot (2x + 1). \qquad \square$$

In 1982 A.K. Lenstra, H.W. Lenstra, L. Lovász designed a factorization algorithm for integer polynomials, the complexity of which is only polynomial in the size of the input. Their idea relies of the computation of shortest vectors in lattices. For details we refer to [Win96], Section 5.3.

Multivariate polynomials

L. Kronecker, in (Kronecker 1882), describes a method for reducing the factorization of a multivariate polynomial over a unique factorization domain I to the factorization of a univariate polynomial over I. The transformation introduces an exponential step in the factorization algorithm.

When we are dealing with multivariate polynomials over the integers, we need not use the quite time consuming Kronecker algorithm, but we can instead use evaluation homomorphisms to construct a lifting approach. Let

$$f(x_1, \dots, x_{n-1}, x_n)$$

be a primitive squarefree (w.r.t. x_n) polynomial in $\mathbb{Z}[x_1, \ldots, x_n]$. We choose an evaluation point $(a_1, \ldots, a_{n-1}) \in \mathbb{Z}^{n-1}$ which preserves the degree and squarefreeness, factor the univariate polynomial

$$f(a_1, \ldots, a_{n-1}, x_n),$$

and finally lift this factorization modulo the prime ideal

$$\mathcal{P} = \langle x_1 - a_1, \ldots, x_{n-1} - a_{n-1} \rangle$$

to a factorization moludo $\mathcal{P}^k$ for high enough k. Here we only give an example.

Example 5.2.4. Let us factor the squarefree integral polynomial

$$\begin{aligned} f(x_1, x_2, x) &= x^3 + ((x_1+2)x_2 + 2x_1 + 1)x^2 + \\ &\quad ((x_1+2)x_2^2 + (x_1^2 + 2x_1 + 1)x_2 + 2x_1^2 + x_1)x + \\ &\quad (x_1+1)x_2^3 + (x_1+1)x_2^2 + (x_1^3 + x_1^2)x_2 + x_1^3 + x_1^2. \end{aligned}$$

First we choose an evaluation point that preserves the degree and squarefreeness and has as many zero components as possible.
$(x_1, x_2) = (0, 0)$: $f(0, 0, x) = x^3 + x^2$ is not squarefree, but
$(x_1, x_2) = (1, 0)$: $f(1, 0, x) = x^3 + 3x^2 + 3x + 2$ is squarefree.
By the change of variables $x_1 = w + 1, x_2 = z$ we move the evaluation point to the origin,

$$\begin{aligned} f(w+1, z, x) &= x^3 + 3x^2 + 3x + 2 + \\ &\quad w^3 + (2x+4)w^2 + (2x^2 + 5x + 5)w + \\ &\quad (w+2)z^3 + ((x+1)w + (3x+2))z^2 + \\ &\quad (w^3 + (x+4)w^2 + (x^2 + 4x + 5)w + (3x^2 + 4x + 2))z. \end{aligned}$$

By $f_{ij}(x)$ we denote the coefficient of $w^j z^i$ in f.

We factor f_{00} (i.e. f evaluated at $(w, z) = (0, 0)$) in $\mathbb{Z}[x]$, getting

$$x^3 + 3x^2 + 3x + 2 = \underbrace{(x+2)}_{g_{00}} \underbrace{(x^2 + x + 1)}_{h_{00}}.$$

Degree bounds for w and z in factors of

$$f(w+1, z, x) = g(w, z, x)h(w, z, x),$$

are $\deg_w(g), \deg_w(h) \leq 3$, and $\deg_z(g), \deg_z(h) \leq 3$.

We lift g_{00} and h_{00} to highest degrees in w and z. We use the ansatz

$$\begin{aligned} g(w, z, x) &= g_{00}(x) + g_{01}(x)w + g_{02}(x)w^2 + g_{03}(x)w^3 + \\ &\quad (g_{10}(x) + g_{11}(x)w + g_{12}(x)w^2 + g_{13}(x)w^3)z + \\ &\quad (g_{20}(x) + g_{21}(x)w + g_{22}(x)w^2 + g_{23}(x)w^3)z^2 + \\ &\quad (g_{30}(x) + g_{31}(x)w + g_{32}(x)w^2 + g_{33}(x)w^3)z^3\ , \end{aligned}$$

and analogously for $h(w, z, x)$.
First we lift to a factorization of $f(w+1, 0, x)$: a formal multiplication of g and h leads to the equations

$$\begin{aligned} f_{01} &= g_{01}h_{00} + g_{00}h_{01}, \\ f_{02} &= g_{00}h_{02} + g_{01}h_{01} + g_{02}h_{00}, \\ f_{03} &= g_{00}h_{03} + g_{01}h_{02} + g_{02}h_{01} + g_{03}h_{00}. \end{aligned}$$

These equations can be solved by a modification of the extended Euclidean algorithm, yielding

$$f(w+1,0,x) = ((x+2)+1\cdot w)\,((x^2+x+1)+(x+2)w+w^2).$$

Now we lift to a factorization of $f(w+1,z,x)$: again by the extended Euclidean algorithm we successively solve

$$\begin{array}{rcl} f_{10} & = & g_{00}h_{10}+g_{10}h_{00}, \\ f_{11}-g_{01}h_{10}-g_{10}h_{01} & = & g_{00}h_{11}+g_{11}h_{00}, \\ f_{20}-g_{10}h_{10} & = & g_{00}h_{20}+g_{20}h_{00}. \end{array}$$

All the other equations have 0 as their left hand sides.

This leads to the factor candidates

$$f(w+1,z,x) = ((x+2)+w+(2+w)z)\,\cdot\,((x^2+x+1)+(x+2)w+w^2+xz+z^2),$$

which are the actual factors. By resubstituting $w = x_1 - 1, z = x_2$ we finally arrive at the factorization

$$f(x_1,x_2,x) = (x+x_1x_2+x_1+x_2+1)\,\cdot\,(x^2+(x_1+x_2)x+x_1^2+x_2^2). \qquad \square$$

5.3 Factorization over algebraic extension fields

We describe an algorithm that has been presented in [vdw70] and slightly improved by B. Trager in 1976.

Let K be a computable field of characteristic 0 such that there is an algorithm for factoring polynomials in $K[x]$. Let α be algebraic over K with minimal polynomial $p(y)$ of degree n. Throughout this section we call K the *ground field* and $K(\alpha)$ the *extension field.* Often we will write a polynomial $f(x) \in K(\alpha)[x]$ as $f(x, \alpha)$ to indicate the occurrence of α in the coefficients. Let $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$ be the roots of $p(y)$ in a splitting field of p over K. By ϕ_j, $1 \le j \le n$, we denote the canonical field isomorphism that takes α into α_j, i.e.

$$\begin{array}{rcll} \phi_j : K(\alpha) & \longrightarrow & K(\alpha_j) & \\ \alpha & \longmapsto & \alpha_j & \\ a & \longmapsto & a & \quad \text{for all } a \in K. \end{array}$$

ϕ_j can be extended to $\phi_j : K(\alpha)[x] \longrightarrow K(\alpha_j)[x]$ by letting it act on the coefficients.

We will reduce the problem of factorization in $K(\alpha)[x]$ to factorization in $K[x]$. This reduction will be achieved by associating a $g \in K[x]$ with the given $f \in K(\alpha)[x]$ such that the factors of f are in a computable 1–1 correspondence with the factors of g, i.e.

$$\begin{array}{rcl} f \in K(\alpha)[x] & \longleftrightarrow & g \in K[x] \\ \text{factors of } f & \overset{1-1}{\longleftrightarrow} & \text{factors of } g. \end{array}$$

A candidate for such a function is the *norm*, which maps an element in the extension field to the product of all its conjugates over K. This product is an element of K.

$$\begin{array}{rcl} \mathrm{norm}_{[K(\alpha)/K]} : K(\alpha) & \longrightarrow & K \\ \beta & \longmapsto & \prod_{\beta' \sim \beta} \beta', \end{array}$$

where $\beta' \sim \beta$ means that β' is conjugate to β relative to $K(\alpha)$ over K. I.e. if $\beta = q(\alpha)$ is the normal representation of β in $K(\alpha)$, then

$$\mathrm{norm}_{[K(\alpha)/K]}(\beta) = \prod_{i=1}^{n} q(\alpha_i).$$

If the field extension is clear from the context, we write just $\mathrm{norm}(\cdot)$ instead of $\mathrm{norm}_{[K(\alpha)/K]}(\cdot)$. Since the norm is symmetric in the α_i's, by the fundamental theorem on symmetric functions is can be expressed in terms of the coefficients of p and thus lies in K. The norm can be generalized from $K(\alpha)$ to $K(\alpha)[x]$ by defining the norm of a polynomial $h(x, \alpha)$ to be $\prod_{i=1}^{n} h(x, \alpha_i)$, which can be computed as

$$\mathrm{norm}(h(x, \alpha)) = \mathrm{res}_y(h(x, y), p(y)).$$

Clearly the norm can be generalized to multivariate polynomials. One important property of the norm is multiplicativity, i.e.

$$\mathrm{norm}(f \cdot g) = \mathrm{norm}(f) \cdot \mathrm{norm}(g). \tag{5.4.1}$$

Theorem 5.3.1. *If $f(x,\alpha)$ is irreducible over $K(\alpha)$, then* $\mathrm{norm}(f) = h(x)^j$ *for some irreducible $h \in K[x]$ and some $j \in \mathbb{N}$.*

Proof: Assume $\mathrm{norm}(f) = g(x)h(x)$ and g, h are relatively prime. For $1 \leq i \leq n$ let $f_i(x) = f(x, \alpha_i)$. Clearly $f = f_1$ divides $\mathrm{norm}(f) = \prod f_i$. So, since f is irreducible, $f|g$ or $f|h$. W.l.o.g. let us assume that $f|h$, i.e. $h(x) = f_1(x,\alpha)\tilde{h}(x,\alpha)$. Then $h(x) = \phi_j(h) = \phi_j(f_1)\phi_j(\tilde{h}) = f_j\tilde{h}(x,\alpha_j)$. Therefore, $f_j|h$ for $1 \leq j \leq n$. Since g and h are relatively prime, this implies that $\gcd(f_j, g) = 1$ for $1 \leq j \leq n$. Thus, $\gcd(\mathrm{norm}(f), g) = 1$, i.e. $g = 1$. □

The previous theorem yields a method for finding minimal polynomials for elements $\beta \in K(\alpha)$. Let $\beta = q(\alpha)$, $b(x) = \mathrm{norm}(x - \beta) = \mathrm{norm}(x - q(\alpha))$. $x - \beta|b(x)$, so $b(\beta) = 0$. Therefore the minimal polynomial $p_\beta(x)$ has to be one of the irreducible factors of $b(x)$. By Theorem 5.3.1, $b(x) = p_\beta(x)^j$ for some $j \in \mathbb{N}$. So $p_\beta(x)$ can be determined by squarefree factorization of $b(x)$.

$K(\alpha)[x]$ is a Euclidean domain, so by successive application of the Euclidean algorithm the problem of factoring in $K(\alpha)[x]$ can be reduced to the problem of factoring squarefree polynomials in $K(\alpha)[x]$ (see Section 3.3). From now on let us assume that $f(x,\alpha) \in K(\alpha)[x]$ is squarefree.

Theorem 5.3.2. *Let $f(x,\alpha) \in K(\alpha)[x]$ be such that $F(x) =$* $\mathrm{norm}(f)$ *is squarefree. Let $F(x) = \prod_{i=1}^r G_i(x)$ be the irreducible factorization of $F(x)$. Then $\prod_{i=1}^r g_i(x,\alpha)$, where $g_i(x,\alpha) =$* $\gcd(f, G_i)$ *over $K(\alpha)$, is the irreducible factorization of $f(x,\alpha)$ over $K(\alpha)$.*

Proof: The statement follows from

(i) every g_i divides f,

(ii) every irreducible factor of f divides one of the g_i's,

(iii) the g_i's are relatively prime, and

(iv) every g_i is irreducible.

Ad (i): This is obvious from $g_i = \gcd(f, G_i)$.

Ad (ii): Let $v(x,\alpha)$ be an irreducible factor of f over $K(\alpha)$. By Theorem 5.4.1, $\mathrm{norm}(v) = w(x)^k$ for some irreducible $w(x) \in K[x]$. $v|f$ implies $\mathrm{norm}(v)|\mathrm{norm}(f)$. Since $\mathrm{norm}(f)$ is squarefree, $\mathrm{norm}(v)$ is irreducible and must be one of the G_i's. So $v \mid g_i(x,\alpha)$.

Ad (iii): Suppose the irreducible factor v of f divides both g_i and g_j for $i \neq j$. Then the irreducible polynomial $\mathrm{norm}(v)$ divides both $\mathrm{norm}(G_i) = G_i^n$ and $\mathrm{norm}(G_j) = G_j^n$. This would mean that G_i and G_j have a common factor.

Ad (iv): Clearly every g_i is squarefree. Assume that $v_1(x,\alpha)$ and $v_2(x,\alpha)$ are distinct irreducible factors of f and that both of them divide $g_i = \gcd(f, G_i)$. $v_1|G_i$ implies $\mathrm{norm}(v_1)|\mathrm{norm}(G_i) = G_i(x)^n$. Because of the squarefreeness of $\mathrm{norm}(f)$, we must have $\mathrm{norm}(v_1) = G_i$. Similarly we get $\mathrm{norm}(v_2) = G_i$. But $(v_1 \cdot v_2)|f$ implies $\mathrm{norm}(v_1 \cdot v_2) = G_i(x)^2|\mathrm{norm}(f)$, in contradiction to the squarefreeness of $\mathrm{norm}(f)$. □

So we can solve our factorization problem over $K(\alpha)$, if we can show that we can restrict our problem to the situation in which $\mathrm{norm}(f)$ is squarefree. The following lemmata and theorem will guarantee exactly that.

Lemma 5.3.3. *If $f(x)$ is a squarefree polynomial in $K[x]$, then there are only finitely many $s \in K$ for which* $\mathrm{norm}(f(x - s\alpha))$ *is not squarefree.*

Proof: Let $\beta_1, \ldots, \beta_m$ be the distinct roots of f. Then the roots of $f(x - s\alpha_j)$ are $\beta_i + s\alpha_j$, $1 \le i \le m$. Thus, the roots of $G(x) = \mathrm{norm}(f(x - s\alpha_j)) = \prod_{k=1}^{n} f(x - s\alpha_k)$ are $\beta_i + s\alpha_k$ for $1 \le i \le m, 1 \le k \le n$. G can have a multiple root only if

$$s = \frac{\beta_j - \beta_i}{\alpha_k - \alpha_l},$$

where $k \ne l$. There are only finitely many such values. □

Lemma 5.3.4. *If $f(x, \alpha)$ is a squarefree polynomial in $K(\alpha)[x]$, then there exists a squarefree polynomial $g(x) \in K[x]$ such that $f|g$.*

Proof: Let $G(x) = \mathrm{norm}(f(x, \alpha)) = \prod g_i(x)^i$ be the squarefree factorization of the norm of f. Since f is squarefree, $f|g := \prod g_i(x)$. □

Theorem 5.3.5. *For any squarefree polynomial $f(x, \alpha) \in K(\alpha)[x]$ there are only finitely many $s \in K$ for which* $\mathrm{norm}(f(x - s\alpha))$ *is not squarefree.*

Proof: Let $g(x)$ be as in Lemma 5.3.4. By Lemma 5.3.3 there are only finitely many $s \in K$ for which $\mathrm{norm}(g(x - s\alpha))$ is not squarefree.
But $f|g$ implies $\mathrm{norm}(f(x - s\alpha))|\mathrm{norm}(g(x - s\alpha))$. If $\mathrm{norm}(f(x - s\alpha))$ is not squarefree, then neither is $\mathrm{norm}(g(x - s\alpha))$. □

```
SQFR_NORM(squarefree norm)
for $f \in K(\alpha)[x]$ squarefree,
we compute $s \in \mathbb{N}$, $g(x) = f(x - s\alpha)$, $N(x) = \mathrm{norm}(g(x, \alpha))$,
s.t. $N(x)$ is squarefree.
(1)  $s := 0$; $g(x, \alpha) := f(x, \alpha)$;
(2)  $N(x) := \mathrm{res}_y(g(x, y), p(y))$;
(3)  while $\deg(\gcd(N(x), N'(x))) \ne 0$ do
         $s := s + 1$;
         $g(x, \alpha) := g(x - \alpha, \alpha)$;
         $N(x) := \mathrm{res}_y(g(x, y), p(y))$;      □
```

So over an infinite field we can always find a transformation of the form $f(x - s\alpha)$, $s \in \mathbb{N}$, such that $\mathrm{norm}(f(x - s\alpha))$ is squarefree. These considerations give rise to an algorithm for computing a linear change of variable which transforms f to a polynomial with squarefree norm.

Now we are ready to present an algorithm for factoring polynomials over the extension field.

```
FACTOR_ALG(factorization over algebraic extension fields)
for given squarefree $f \in K(\alpha)[x]$,
the list $F$ of irreducible factors of $f$ over $K(\alpha)$ is detemined.
(1)  $[g, s, N] :=$ SQFR_NORM($f$);
(2)  $L :=$ list of irreducible factors of $N(x)$ over $K$;
(3)  if length($L$) = 1 then return([$f$]);
(4)  $F := [\,]$;
     for each $H(x)$ in $L$ do
          $h(x, \alpha) := \gcd(H(x), g(x, \alpha))$;
          $g(x, \alpha) := g(x, \alpha)/h(x, \alpha)$;
          add $h(x + s\alpha, \alpha)$ to $F$ ;
     □
```

Example 5.3.6. We apply the factorization algorithm FACTOR_ALG to the domain $\mathbb{Q}(\sqrt[3]{2})[x]$, i.e. $K = \mathbb{Q}$, α a root of $p(y) = y^3 - 2$. Let us factor the polynomial

$$f(x, \alpha) = x^4 + \alpha x^3 - 2x - 2\alpha.$$

$f(x, \alpha)$ is squarefree. First we have to transform f to a polynomial g with squarefree norm. The norm of f itself is

$$\mathrm{norm}(f) = \mathrm{res}_y(f(x, y), p(y)) = -(x^3 - 2)^3(x^3 + 2),$$

i.e. it is not squarefree. The transformation $x \mapsto x - \alpha$ does not work, but $x \mapsto x - 2\alpha$ does:

$$\begin{aligned} g(x, \alpha) &:= f(x - 2\alpha, \alpha) = x^4 - 7\alpha x^3 + 18\alpha^2 x^2 - 42x + 18\alpha, \\ N(x) &= \mathrm{norm}(g) = x^{12} - 56x^9 + 216x^6 - 6048x^3 + 11664, \end{aligned}$$

and $N(x)$ is squarefree. The factorization of $N(x)$ is

$$N(x) = (x^3 - 2)(x^3 - 54)(x^6 + 108).$$

Computing the gcd of all the factors of $N(x)$ with $g(x, \alpha)$ gives us the factorization of $g(x, \alpha)$:

$$g(x, \alpha) = (x - \alpha)(x - 3\alpha)(x^2 - 3\alpha x + 3\alpha^2),$$

which can be transformed by $x \mapsto x + 2\alpha$ to the factorization

$$f(x, \alpha) = (x + \alpha)(x - \alpha)(x^2 + \alpha x + \alpha^2). \qquad \square$$

Sometimes we want to factor a polynomial over the algebraic closure of the field of coefficients; i.e., we want to factor the polynomial into as many factors as possible over any algebraic extension field. This is called *absolute factorization.* Alsolute factorization is algorithmic. The basic idea is presented in [Win96], Section 5.5.