FACTORISATIE VAN POLYNOMEN

door

A.K. LENSTRA

1. INLEIDING

We beschrijven een methode om polynomen in één variabele met coëfficiënten in een algebraïsch getallenlichaam $\mathbb{Q}(\alpha)$ te factoriseren. Van groot belang hierbij zijn snelle algoritmen voor de factorisatie in $\mathbb{F}_q[X]$. We geven een overzicht van de op dit gebied bestaande technieken, waarbij we ook aandacht besteden aan de probabilistische methoden voor het geval dat de karakteristiek groot is. Verder maken we gebruik van een belangrijke toepassing van het lemma van Hensel: de zogenaamde 'lift'-algoritme met behulp waarvan een factor over $\mathbb{F}_q$ kan worden gelift tot een factor over $W_k(\mathbb{F}_q)$. Ook komen enkele specifiek in $\mathbb{Z}[X]$ optredende problemen met niet-monische polynomen ter sprake bij de factorisatie in $\mathbb{Z}[X]$.

De factorisatie van kwadraatvrije polynomen over $\mathbb{Z}$, of over $\mathbb{Q}(\alpha)$, gaat meestal volgens het volgende schema (niet bijvoorbeeld CLAYBROOK [6]). Eerst wordt voor een geschikt gekozen eindig lichaam $\mathbb{F}_q$ de factorisatie over $\mathbb{F}_q$ bepaald (Hoofdstukken 2, 3 en 4). De zo gevonden factorisatie wordt uitgebreid tot een factorisatie over $W_k(\mathbb{F}_q)$ (Hoofdstuk 5), waarbij k zo wordt gekozen dat de coëfficiënten van de factoren groot genoeg zijn (Hoofdstuk 6). Tenslotte worden met behulp van de factoren over $W_k(\mathbb{F}_q)$ de factoren over $\mathbb{Z}$, dan wel $\mathbb{Q}(\alpha)$, bepaald (Hoofdstukken 8 en 9). Niet-kwadraatvrije polynomen ontbinden we in het produkt van kwadraatvrije factoren (Hoofdstuk 7) waarop we de boven geschetste methode toepassen.

We gebruiken de volgende namen en notaties:

F — Een monisch irreducibel polynoom in $\mathbb{Z}[X]$ met graad m.

$\mathbb{Q}(\alpha)$ — Een algebraïsch getallenlichaam, met α nulpunt van F. Elementen van $\mathbb{Q}(\alpha)$ worden voorgesteld door $\sum_{i=0}^{m-1} q_i \alpha^i$ met $q_i \in \mathbb{Q}$.

R — De ring van gehelen van $\mathbb{Q}(\alpha)$.

$\mathbb{Z}[\alpha]$ - Een deelverzameling van $\mathbb{Q}(\alpha)$ bestaande uit elementen van de vorm $\sum_{i=0}^{m-1} k_i \alpha^i$ met $k_i \in \mathbb{Z}$.

$\mathbb{Z}/p\mathbb{Z}$ - Het lichaam met p elementen, p priem.
$\mathbb{Z}/2\mathbb{Z} = \{0,1\}$ en $\mathbb{Z}/p\mathbb{Z} = \{-\lfloor p/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor p/2 \rfloor\}$ voor p oneven.

$\mathbb{F}_q$ - Het lichaam met q elementen met $q = p^\ell$, p priem, $\ell \in \mathbb{N}$.
Bij implementatie van de aritmetische bewerkingen in $\mathbb{F}_q$ moet voor $\ell > 1$ behalve het priemgetal p ook een monisch modulo p irreducibel ℓ-de graads polynoom $G \in (\mathbb{Z}/p\mathbb{Z})[X]$ bekend zijn. De elementen van $\mathbb{F}_q$ worden dan voorgesteld door $\sum_{i=0}^{\ell-1} a_i \beta^i$ met $a_i \in \mathbb{Z}/p\mathbb{Z}$ en β een nulpunt van G.

$\mathbb{Z}/p^k\mathbb{Z}$ - Een ring met p^k elementen, p priem, $k \in \mathbb{N}$.
$\mathbb{Z}/2^k\mathbb{Z} = \{-2^{k-1}+1, \ldots, -1, 0, 1, \ldots, 2^{k-1}\}$ en
$\mathbb{Z}/p^k\mathbb{Z} = \{-\lfloor p^k/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor p^k/2 \rfloor\}$ voor p oneven.

$W_k(\mathbb{F}_q)$ - Een ring met $p^k \cdot q$ elementen, de zogenaamde afgeknotte Witt-ring. Bij implementatie van de aritmetische bewerkingen in $W_k(\mathbb{F}_q)$ hebben we eenzelfde polynoom $G \in (\mathbb{Z}/p\mathbb{Z})[X]$ nodig als bij $\mathbb{F}_q$. De elementen van de afgeknotte Witt-ring worden dan voorgesteld door $\sum_{i=0}^{\ell-1} a_i \beta^i$ met $a_i \in \mathbb{Z}/p^k\mathbb{Z}$ en β een nulpunt van G.
Het volgende plaatje is op dit moment wellicht illustratief:

$$\begin{array}{ccc} \mathbb{Z}/p^k\mathbb{Z} & \hookrightarrow & \mathbb{Z}/p\mathbb{Z} \\ \cap & & \cap \\ W_k(\mathbb{F}_q) & \hookrightarrow & \mathbb{F}_q \end{array}$$

Ten overvloede: $\mathbb{Z}/p^t\mathbb{Z}$ is niet hetzelfde als $\mathbb{F}_{p^t}$.

Als f een polynoom is:

cofactor - Als voor een polynoom geldt $g|f$, dan heet f/g de cofactor van g.

cont(f) - De grootste gemene deler van de coëfficiënten van f (content).

deg(f) - De graad van f.

f_i - De coëfficiënt van de i-de graads term van f. $f_i = 0$ voor $i < 0$ of $i > \deg(f)$.

kwadraatvrij - f heet kwadraatvrij als f geen factoren met multipliciteit groter dan 1 bevat. D.w.z., als $g|f$ dan $g^2 \nmid f$ voor iedere factor g van f ($g \neq 1$).

lc(f) - $f_{\deg(f)}$ (leading coefficient).

pp(f) - f/cont(f) (primitive part).

primitief - f heet primitief als f = pp(f).

2. PARTIËLE FACTORISATIEMETHODEN IN $\mathbb{F}_q[X]$

In dit hoofdstuk geldt dat f een monisch polynoom van graad n in $\mathbb{F}_q[X]$ is. We zullen twee elementaire methoden behandelen om een partiële factorisatie van f over $\mathbb{F}_q$ te verkrijgen. Deze methoden zijn vaak minder tijdrovend dan de volledige factorisatiemethoden over $\mathbb{F}_q$; ze worden derhalve gebruikt om de volledige factorisatie te vereenvoudigen of zelfs in bepaalde gevallen overbodig te maken.

De eerste partiële factorisatiemethode is het kwadraatvrij maken van een polynoom. We hebben hiervoor drie lemmata nodig, die we zonder bewijs vermelden.

LEMMA 2.1. $x,y \in \mathbb{F}_q$, *dan* $(x+y)^p = x^p + y^p$.
$g \in \mathbb{F}_q[X]$, *dan* $g(x)^q = g(x^q)$.

LEMMA 2.2. *Iedere factor van* f *met multipliciteit groter dan* 1 *deelt ook de afgeleide* f' *van* f.

LEMMA 2.3. $f' = 0 \iff \exists g \in \mathbb{F}_q[X]$ *met* $f(x) = g(x^p)$.

Met behulp van deze drie lemmata kunnen we nu een polynoom in $\mathbb{F}_q[X]$ kwadraatvrij maken. Veronderstel $f = \prod_{i=1}^{k} f_{(i)}^i$, en dus $\sum_{i=1}^{k} i \cdot \deg(f_{(i)}) = n$. Als de afgeleide f' van f nul is, dan geldt wegens Lemma 2.3 dat $f(x) = g(x^p)$ voor zekere $g \in \mathbb{F}_q[X]$. Wegens het feit dat ieder element uit $\mathbb{F}_q$ een unieke p-de machtswortel in $\mathbb{F}_q$ bezit, en met gebruikmaking van Lemma 2.1, kunnen we f nu schrijven als $f(x) = h(x)^p$. Pas nu de methode recursief toe op h. Indien f' ongelijk nul is berekenen we de grootste gemene deler g van f en f'. Als g triviaal is, dat wil zeggen $\deg(g) = 0$, dan is f wegens Lemma 2.2 kwadraatvrij, en we zijn klaar. Anders heeft g de volgende gedaante:

$$g = \prod_{\substack{i=2 \\ i \neq j \cdot p}}^{k} f_{(i)}^{i-1} \cdot \prod_{j=1}^{k/p} f_{(j \cdot p)}^{j \cdot p} \qquad \text{(Lemma 2.2)}.$$

We kunnen g nu wegdelen uit f, en we houden de kwadraatvrije factor $\prod_{\substack{i=1 \\ i \neq j \cdot p}}^{k} f_{(i)}$

van f over. Door ditzelfde proces nu op g toe te passen kunnen we ook de kwadraatvrije factor $\prod_{j=1}^{k/p} f_{(j\cdot p)}$ van f bepalen.

OPMERKING 2.1. Het is eenvoudig mogelijk om bij implementatie de algoritme die f kwadraatvrij moet maken zo in te richten dat alle factoren $f_{(i)}$, $i = 1,\ldots,k$, van f afzonderlijk worden gevonden, samen met hun multipliciteit i. Een gedetailleerde beschrijving van een dergelijke algoritme voor karakteristiek ongelijk nul voert nu te ver. Een duidelijke beschrijving voor het geval dat de karakteristiek nul is staat in MUSSER [21] en YUN [34].

Nu we mogen aannemen dat f kwadraatvrij is kunnen we de tweede partiële factorisatiemethode presenteren. Het betreft hier het ontbinden van f in het produkt van polynomen die op hun beurt het produkt zijn van irreducibele polynomen van gelijke graad. We gaan dus een factorisatie $f = \prod_{i=1}^{n} g_{(i)}$ van f over $\mathbb{F}_q$ bepalen, waarbij geldt dat $g_{(i)}$ het produkt is van alle irreducibele factoren van f met graad i. Merk op dat we door deze factorisatie ook weten in hoeveel irreducibele factoren f uiteenvalt. De methode berust op het volgende lemma.

LEMMA 2.4. *Het polynoom* $x^{q^k} - x \in \mathbb{F}_q[X]$ *factoriseert over* $\mathbb{F}_q$ *in het produkt van alle monische irreducibele polynomen met graden* d *zodat* $d \mid k$.

Het moge nu duidelijk zijn hoe we de $g_{(i)}$ kunnen bepalen:

$x^q - x$	is het produkt van alle monische eerstegraads polynomen, dus $g_{(1)} = \gcd(f, x^q - x)$.
$f/g_{(1)}$	bevat geen eerstegraads factoren meer, want f is kwadraatvrij.
$x^{q^2} - x$	is het produkt van alle monische irreducibele eerste- en tweedegraads polynomen, dus $g_{(2)} = \gcd(f/g_{(1)}, x^{q^2} - x)$.
.	.
.	.
.	.
$f/\prod_{i=1}^{k-1} g_{(i)}$	bevat geen factoren van graad $\leq k-1$.
	$x^{q^k} - x$ is het produkt van alle monische irreducibele polynomen met graad delend op k, dus $g_{(k)} = \gcd(f/\prod_{i=1}^{k-1} g_{(i)}, x^{q^k} - x)$.

Ga netzolang door tot op een gegeven moment, na maximaal $\lfloor n/2 \rfloor$ stappen $f = \prod_{i=1}^{k} g_{(i)}$ voor zekere $k \leq n$.

OPMERKING 2.2. We kunnen deze constructie ook gebruiken om f kwadraatvrij te maken, door de ggd met $x^{q^i} - x$ zolang te nemen tot deze 1 is en de gevonden ggd steeds uit te delen.

OPMERKING 2.3. Bij implementatie van deze algoritme gaan we, wat betreft de opvolgende q-de machten van x^q, zeker niet zo eenvoudig te werk als hier geschetst is. Het eerste wat we opmerken is dan $x^{q^i} - x$ voor ons alleen interessant is modulo f. We zouden dus x^{q^i} kunnen berekenen door de q-de macht van $x^{q^{i-1}}$ modulo f te bepalen. In het slechtste geval vergt dit echter $\lfloor n/2 \rfloor$ q-de machtsverheffingen, en dat is geen aantrekkelijk vooruitzicht. De volgende methode geeft een aanzienlijke versnelling voor grote q of n.

Definieer een $n \times n$ matrix Q waarvan de i-de rij bestaat uit de coëfficiënten van $x^{i\cdot q}$ modulo f $(i = 0,\ldots,n-1)$. Dus als $g = x^{i\cdot q}$ modulo f, dan is $q_{ij} = g_j$. Stel nu $x^{q^{i-1}}$ modulo $f = \sum_{j=0}^{n-1} r_{i-1,j} x^j$, dan is

$$x^{q^i} \text{ modulo } f = (x^{q^{i-1}} \text{ modulo } f)^q \text{ modulo } f$$

$$= \left(\sum_{j=0}^{n-1} r_{i-1,j} x^j\right)^q \text{ modulo } f$$

$$= \sum_{j=0}^{n-1} r_{i-1,j}^q \cdot x^{j\cdot q} \text{ modulo } f$$

$$= \sum_{j=0}^{n-1} r_{i-1,j} \cdot x^{j\cdot q} \text{ modulo } f.$$

Hieruit volgt dat $(r_{i-1,0},\ldots,r_{i-1,n-1})\cdot Q = (r_{i,0},\ldots,r_{i,n-1})$. Hebben we nu Q eenmaal berekend (en dat kost 1 q-de machtsverheffing modulo f en n-2 vermenigvuldigingen modulo f in $\mathbb{F}_q[X]$), dan volgt x^{q^i} snel uit $x^{q^{i-1}}$. We zullen deze partiële factorisatiemethode en het gebruik van de matrix Q aan de hand van een voorbeeld illustreren.

VOORBEELD 2.1. Laat $f = x^6 + 3x^5 + 2x^4 + 3x^3 - 3x^2 + 3x + 2 \in \mathbb{F}_7[X]$, dus $q = p = 7$ en $\ell = 1$. f is kwadraatvrij wegens $f' \neq 0$ en $\gcd(f,f') = 1$. We bepalen eerst de matrix Q.

x^7 modulo f = $3x^4-2x^3+2x^2-1$,
x^{14} modulo f = $(x^7 \text{ modulo } f)^2$ modulo f = $-x^5+3x^4-2x^3-x^2-x+2$,
x^{21} modulo f = $(x^{14} \text{ modulo } f)\cdot(x^7 \text{ modulo } f)$ modulo f = $-x^5+3x^3+2x^2+3x+2$,
x^{28} modulo f = ... = $2x^5-2x^4+2x^3-x^2+x+2$ en
x^{35} modulo f = ... = $x^4-3x^3-x^2+2x-2$.

Dus

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & -2 & 3 & 0 \\ 2 & -1 & -1 & -2 & 3 & -1 \\ 2 & 3 & 2 & 3 & 0 & -1 \\ 2 & 1 & -1 & 2 & -2 & 2 \\ -2 & 2 & -1 & -3 & 1 & 0 \end{pmatrix}$$

Het produkt van de irreducibele factoren van graad 1 bepalen we nu door
$g_{(1)} = \gcd(x^7-x,f) = \gcd(3x^4-2x^3+2x^2-x-1,f) = x + 1$.
$g_{(1)}$ delen we weg uit f, en we houden als cofactor van $g_{(1)}$ over
$f/g_{(1)} = x^5 + 2x^4 + 3x^2 + x + 2$.
Om $g_{(2)}$ te berekenen moeten we x^{49} modulo f bepalen en dat doen we door Q van voren te vermenigvuldigen met het als rij-vector opgevatte polynoom x^7 modulo f.

$$(-1,0,2,-2,3,0)\cdot Q = (-2,2,-2,3,0,-1).$$

Dux x^{49} modulo $f = -x^5 + 3x^3 - 2x^2 + 2x - 2$ en
$g_{(2)} = \gcd(x^{49}-x,f/(g_{(1)}) = \gcd(-x^5+3x^3-2x^2+x-2,x^5+2x^4+3x^2+x+2) = x^2 + 2x - 2$.
$g_{(2)}$ delen we weg uit de cofactor van $g_{(1)}$, en we houden als cofactor van $g_{(1)}\cdot g_{(2)}$ over $f/(g_{(1)}\cdot g_{(2)}) = x^3 + 2x - 1$.

Het is duidelijk dat we nu niet verder hoeven te zoeken; we hebben de volledige factorisatie van f over $\mathbb{F}_7$ gevonden:

$$f = (x+1)\cdot(x^2+2x-2)(x^3+2x-1).$$

OPMERKING 2.4. Een verbeterde maar tevens veel gecompliceerdere versie van deze algoritme is te vinden in BERLEKAMP [2].

3. NULPUNTEN VAN POLYNOMEN IN $\mathbb{F}_q[X]$

We hebben in dit hoofdstuk dezelfde f als in Hoofdstuk 2. Door eventueel de ggd van f en x^q-x te nemen mogen we aannemen dat f uiteenvalt in n verschillende lineaire factoren (Lemma 2.4). Als q klein is kunnen we eenvoudig deze factoren vinden door iedere factor $x - s$, $s \in \mathbb{F}_q$, te proberen. Voor grote q werkt dit natuurlijk ook, maar dan kunnen we beter een van de methoden gebruiken die in dit hoofdstuk aan de orde komen.

De eerste nulpunt-bepalingsmethode werkt efficiënt als de karakteristiek p niet al te groot is. Is zelfs dat niet het geval dan zullen we onze toevlucht moeten nemen tot de probabilistische tweede methode, die in de praktijk snel tot resultaten zal leiden, maar die in het slechtste geval niet beter is dan het proberen van alle elementen van het lichaam.

We zullen eerst de deterministische methode, die dus geschikt is voor niet te grote karakteristiek, behandelen. Voor de aritmetiek in $\mathbb{F}_q$ hebben we een monisch modulo p irreducibel polynoom van graad ℓ nodig. Laat β een nulpunt van dit minimumpolynoom zijn, dan vormt de verzameling $\{1,\beta,\beta^2,\ldots,\beta^{\ell-1}\}$ een basis voor $\mathbb{F}_q$ over $\mathbb{Z}/p\mathbb{Z}$. Definieer het spoor-polynoom Tr door $\mathrm{Tr}(x) = \sum_{i=0}^{\ell-1} x^{p^i} = x + x^p + \ldots + x^{p^{\ell-1}}$.

LEMMA 3.1. *Laten f en* $v_{(i)}$ $(i = 1,\ldots,k)$ *monische polynomen zijn in* $\mathbb{F}_q[X]$, *zodat* $\gcd(v_{(i)},v_{(j)}) = 1$ $(i \neq j)$. *Als* $f \mid \prod_{i=1}^{k} v_{(i)}$, *dan* $f = \prod_{i=1}^{k} \gcd(f,v_{(i)})$.

LEMMA 3.2. $x^q - x = \prod_{s\in\mathbb{Z}/p\mathbb{Z}} (\mathrm{Tr}(x)-s)$.

Lemma 3.2 kunnen wij ook schrijven als $(\beta^j x)^q - \beta^j x = \prod_{s\in\mathbb{Z}/p\mathbb{Z}} (\mathrm{Tr}(\beta^j x)-s)$, $0 \le j < \ell$. Met $(\beta^j)^q = \beta^j$ volgt nu $x^q - x = \beta^{-j}\cdot \prod_{s\in\mathbb{Z}/p\mathbb{Z}} (\mathrm{Tr}(\beta^j x) - s)$.

Omdat f in n verschillende lineaire factoren uiteenvalt weten we dat $x^q - x \equiv 0$ modulo f (Lemma 2.4). Dus we vinden dat voor het spoor-polynoom geldt

$$\prod_{s\in\mathbb{Z}/p\mathbb{Z}} (\mathrm{Tr}(\beta^j x) - s) \equiv 0 \text{ modulo } f, \quad 0 \le j < \ell,$$

en omdat de polynomen $\mathrm{Tr}(\beta^j x) - s$, $s \in \mathbb{Z}/p\mathbb{Z}$, relatief priem zijn volgt met Lemma 3.1 dat $f = \prod_{s\in\mathbb{Z}/p\mathbb{Z}} \gcd(f,\mathrm{Tr}(\beta^j x) - s)$, $0 \le j < \ell$.

Dit levert een niet-triviale factorisatie van f op als er een j, $0 \le j < \ell$, bestaat zodat $\mathrm{Tr}(\beta^j x)$ modulo f geen constante is. Is dat het geval, dan ontaardt $f = \prod_{s\in\mathbb{Z}/p\mathbb{Z}} \gcd(f,\mathrm{Tr}(\beta^j x) - s)$ in $f = f\cdot\Pi 1$. Hebben we

een j gevonden zodat $Tr(\beta^j x)$ modulo f niet constant is, dan kan de methode recursief worden toegepast op de gevonden factoren van f, en alle nulpunten van f volgen.

We bewijzen nu dat zo'n j, $0 \le j < \ell$, altijd bestaat. Laten $s_1,\ldots,s_n \in \mathbb{F}_q$ de nulpunten van f zijn, en stel dat voor alle j, $0 \le j < \ell$, geldt dat $Tr(\beta^j x)$ modulo f een constante is. Dat wil zeggen dat er $t_0,\ldots,t_{\ell-1} \in \mathbb{F}_q$ zijn, zodat $Tr(\beta^j x) \equiv t_j$ modulo f, en dus $t_j = Tr(\beta^j s_1) = Tr(\beta^j s_2) = \ldots = Tr(\beta^j s_n)$, $0 \le j < \ell$. Voor ieder paar i,k, $1 \le i < k \le n$, geldt dus dat

$$Tr(\beta^j s_i) = Tr(\beta^j s_k) \Rightarrow \text{(Lemma 2.1) } Tr(\beta^j(s_i - s_k)) = 0, \quad 0 \le j < \ell.$$

Nogmaals gebruik makend van Lemma 2.1 vinden we voor een willekeurig ℓ-tal $p_0,\ldots,p_{\ell-1} \in \mathbb{Z}/p\mathbb{Z}$ dat

$$0 = \sum_{j=0}^{\ell-1} p_j \cdot Tr(\beta^j(s_i - s_k)) = Tr\Big(\Big(\sum_{j=0}^{\ell-1} p_j \beta^j\Big)(s_i - s_k)\Big).$$

$\{1,\beta,\beta^2,\ldots,\beta^{\ell-1}\}$ vormt een basis voor $\mathbb{F}_q$ over $\mathbb{Z}/p\mathbb{Z}$, dus we constateren dat $Tr(s(s_i - s_k)) = 0$, $\forall s \in \mathbb{F}_q$, en omdat $s_i \neq s_k$, dat $Tr(s) = 0$, $\forall s \in \mathbb{F}_q$. Dus Tr(x) heeft q nulpunten, in tegenspraak met Lemma 3.2 waaruit volgt dat Tr(x) $p^{\ell-1}$ nulpunten heeft.

OPMERKING 3.1. Bij implementatie van deze methode kunnen we er in vele gevallen voor zorgen dat we in één keer een j kiezen zodat $Tr(\beta^j x)$ modulo f niet constant is. We bepalen dan de ggd van f met $Tr(\beta^j x) - s$ voor alle $s \in \mathbb{Z}/p\mathbb{Z}$ en passen de methode recursief toe op iedere gevonden niet-triviale factor van f. Merk op dat we in de recursie niet nogmaals de al gebruikte j's hoeven te gebruiken.

Voor we overgaan tot de tweede, probabilistische methode eerst een voorbeeld.

VOORBEELD 3.1. $f = x^3 - 3x^2 + 3x + 5 \in \mathbb{F}_{121}[X]$, dus $q = 11^2$, $p = 11$, $\ell = 2$. Voor de aritmetiek in $\mathbb{F}_{121}$ kiezen we het modulo 11 irreducibele polynoom $G(T) = T^2 + 5$, en β een nulpunt van G. f is kwadraatvrij en valt in verschillende lineaire factoren uiteen omdat $gcd(x^{121} - x, f) = f$ (Lemma 2.4). $Tr(x) = x + x^{11} \equiv 4x^2 + 4x + 5$ modulo f, is niet constant. We gaan nu gcd(f,Tr(x) - s) berekenen voor $s \in \mathbb{Z}/11\mathbb{Z}$. Het blijkt dat deze gcd's alle 1 zijn, behalve voor s = -1 en s = -3. We krijgen

$$gcd(f, 4x^2+4x-5) = x^2 + x - 4 \text{ en}$$
$$gcd(f, 4x^2+4x-3) = x - 4.$$

Aan x - 4 hoeven we niets meer te doen, maar op $g = x^2 + x - 4$ moeten we de methode opnieuw toepassen, waarbij we het geval "j = 0" niet hoeven te beschouwen omdat wegens $gcd(f, 4x^2+4x-5) = x^2 + x - 4$ Tr(x) modulo g een constante is. $Tr(\beta x)$ moet nu wegens bovenstaand bewijs wel resultaten leveren omdat $0 \le j < 2$, en j = 0 hebben we al uitgesloten.

$Tr(\beta x) = \beta x + \beta^{11} x^{11} = 2\beta x + \beta$ modulo g, is inderdaad niet constant en we vinden

$$gcd(g, 2\beta x + \beta + 5) = x + 5\beta - 5 \text{ en}$$
$$gcd(g, 2\beta x + \beta - 5) = x - 5\beta - 5.$$

We hebben nu alle nulpunten van f gevonden: $f = (x-4)(x+5\beta-5)(x-5\beta-5)$.

OPMERKING 3.2. Merk op dat we de eventuele nulpunten van f in $\mathbb{Z}/p\mathbb{Z}$ direct kunnen bepalen door: $g = gcd(f, x^p - x)$ en vervolgens $gcd(g, x-s)$, $s \in \mathbb{Z}/p\mathbb{Z}$.

We zijn nu toegekomen aan de probabilistische nulpunt-bepalingsmethode, afkomstig van M.O. Rabin, een groot voorvechter van de probabilistische aanpak. De methode van Rabin is alleen bruikbaar als de karakteristiek oneven is, hetgeen geen grote beperking is, gezien het feit dat we voor kleine karakteristiek al een bevredigende methode kennen. We nemen daarom voor de rest van dit hoofdstuk aan dat p oneven is.

Omdat $s^q = s$ voor $s \in \mathbb{F}_q$, geldt voor $s \neq 0$ dat $s^{q-1} = 1$ en derhalve $s^{(q-1)/2} = 1$ of $s^{(q-1)/2} = -1$. We nemen $r = \frac{q-1}{2}$ en s met $s^r = 1$ noemen we van type 1 en s met $s^r = -1$ noemen we van type 2.

LEMMA 3.3. *Voor $s_1, s_2 \in \mathbb{F}_q$, ongelijk en niet nul, geldt dat*

$$\#\{s \in \mathbb{F}_q \mid 0 \neq (s_1+s)^r \neq (s_2+s)^r \neq 0\} = r.$$

BEWIJS. Voor $s_i + s \neq 0$ geldt $(s_i+s)^r = 1$ of -1 (i = 1,2) dus we vinden dat

$$0 \neq (s_1+s)^r \neq (s_2+s)^r \neq 0 \iff \left(\frac{s_1+s}{s_2+s}\right)^r = -1.$$

De vergelijking $x^r + 1 = 0$ heeft in $\mathbb{F}_q$ r oplossingen en iedere $t \in \mathbb{F}_q$ die voldoet geeft op unieke wijze een $s \in \mathbb{F}_q$ door $s = (s_1 - t \cdot s_2)/(t-1)$ omdat $t \neq 1$. □

De methode van Rabin berust op de volgende observatie. Alle wortels van $x^r - 1 = 0$ zijn van type 1. Als f nulpunten van verschillende types heeft kunnen we die scheiden door de ggd van $x^r - 1$ en f te berekenen. Als alle nulpunten van f van hetzelfde type zijn, dan kunnen we ze allemaal tegelijk random opschuiven waarbij we volgens Lemma 3.3 een kans van minstens een half hebben dat er twee verschoven nulpunten van verschillend type ontstaan. De verschoven nulpunten zijn dan te scheiden door de ggd met $x^r - 1$ te bepalen. De constructie is dus soortgelijk aan die van de eerste nulpunt-bepalingsmethode, alleen hier schuiven we aan de nulpunten van f, daar verschuiven we het spoor-polynoom en daarmee z'n nulpunten.

Een iets gedetailleerde beschrijving:

stap 1 - Bepaal $g = \gcd(x^r-1,f)$. Als $0 < \deg(g) < n$ ga dan verder met $f_{(1)} = g$ en vervolgens met $f_{(1)} = f/g$, anders met $f_{(1)} = f$.

stap 2 - Kies $s \in \mathbb{F}_q$ willekeurig, net zolang tot voor $h = \gcd(x^r-1, f_{(1)}(x-s))$ geldt dat $0 < \deg(h) < \deg(f_{(1)})$.
Wegens Lemma 3.3 hebben we steeds een kans van minstens een half om een niet triviale factorisatie van $f_{(1)}$ te vinden. Vinden we een nulpunt t van $f_{(1)}(x-s)$ dan is t-s een nulpunt van $f_{(1)}(x)$, immers $x-t \mid f_{(1)}(x-s) \Rightarrow x+s-t \mid f_{(1)}(x+s-s) = f_{(1)}(x)$.

stap 3 - Herhaal stap 2 met h resp. $f_{(1)}/h$ in plaats van $f_{(1)}$ net zolang tot alle nulpunten van f zijn gevonden.

VOORBEELD 3.2. Neem f uit Voorbeeld 3.1. We gaan weer de nulpunten van f in $\mathbb{F}_{121}$ bepalen, maar nu met de methode van Rabin. We kiezen voor de aritmetiek in $\mathbb{F}_{121}$ hetzelfde minimumpolynoom $G(T) = T^2 + 5$, en β met $G(\beta) = 0$.

stap 1 - $\gcd(x^{60}-1,f) = x-4$, we gaan dus verder met $f/(x-4) = x^2+x-4 = f_{(1)}$;

stap 2 - We kiezen $s = \beta+1$ en we vinden $\gcd(x^{60}-1, f_{(1)}(x-\beta-1)) = 1$, een slechte keuze voor s. Vervolgens $s = -2\beta-2$ levert $\gcd(x^{60}-1, f_{(1)}(x+2\beta+2)) = x-4\beta-3$, een goede keuze. $4\beta+3$ is een nulpunt van $f_{(1)}(x+2\beta+2)$, dus $4\beta+3+2\beta+2 = -5\beta+5$ is een nulpunt van $f_{(1)}(x)$. We vinden de factor $x+5\beta-5$ van $f_{(1)}$ met cofactor $x-5\beta-5$. Dus overeenkomstig Voorbeeld 3.1: $f = (x-4)(x+5\beta-5)(x-5\beta-5)$.

4. FACTORISATIE IN $\mathbb{F}_q[X]$

We gaan nu de resultaten van de beide vorige hoofdstukken gebruiken om de volledige factorisatie over $\mathbb{F}_q$ te bepalen van $f \in \mathbb{F}_q[X]$. Voor het gemak

nemen we aan dat f kwadraatvrij en monisch is, dat is gezien de resultaten uit Hoofdstuk 2 geen beperking. Er komen drie methoden ter sprake, één voor $\mathbb{F}_q$ met q klein, één voor $\mathbb{F}_q$ met q groot, waarbij de nulpunt-bepalingsmethoden uit Hoofdstuk 3 een belangrijke rol spelen, en een aparte methode voor $\mathbb{Z}/p\mathbb{Z}$ met p groot. De eerste methode is verreweg de bekendste methode voor factorisatie over $\mathbb{F}_q$ en staat bekend onder de naam *Berlekamp's factorisatie methode*. De tweede methode is een aanpassing van de eerste voor grote q en is afkomstig van H. Zassenhaus. De derde methode is van M.O. Rabin en berust op de door de heer Rabin gepropageerde probabilistische methoden. We zullen zien dat de matrix Q, die bij de partiële factorisatiemethode van Hoofdstuk 2 eigenlijk alleen maar een handigheidje bij het rekenen is, hier een heel wat belangrijker rol speelt.

Veronderstel dat we een polynoom $v \in \mathbb{F}_q[X]$ hebben zodat $f \mid \prod_{s\in\mathbb{F}_q}(v-s)$. De polynomen v-s, $s \in \mathbb{F}_q$, zijn relatief priem, dus volgens Lemma 3.1 geldt dat $f = \prod_{s\in\mathbb{F}_q} \gcd(f,v-s)$. Als bovendien geldt dat v niet triviaal is en graad kleiner dan n = deg(f) heeft, volgt hier een niet-triviale factorisatie uit van f. We gaan nu laten zien dat we met dergelijke polynomen v zelfs de volledige factorisatie van f over $\mathbb{F}_q$ kunnen krijgen.

Wegens Lemma 2.4 kunnen we de voorwaarde $f \mid \prod_{s\in\mathbb{F}_q}(v-s)$ ook schrijven als $f \mid v^q - v$, oftewel $v^q \equiv v$ modulo f. Met behulp van de matrix Q kunnen we nu alle $v \in \mathbb{F}_q[X]$ bepalen met $\deg(v) < n$ zodat $v^q \equiv v$ modulo f.

STELLING 4.1. *Zij* $v \in \mathbb{F}_q[X]$ *met* $\deg(v) < n$, *dan*

$$v^q \equiv v \text{ modulo } f \iff v\cdot Q = v,$$

met v *opgevat als rijvector, en* f, n *en* Q *als boven.*

BEWIJS. Laat q_{ik} het (i,k)-de element van Q zijn, $0 \le i,k < n$.

$$v^q = v(x)^q = v(x^q) = \sum_{i=0}^{n-1} v_i x^{iq} = \sum_{i=0}^{n-1} v_i \sum_{k=0}^{n-1} q_{ik} x^k$$

$$= \sum_{k=0}^{n-1} \Big(\sum_{i=0}^{n-1} v_i q_{ik} \Big) x^k = v = \sum_{k=0}^{n-1} v_k x^k$$

$$\iff \sum_{i=0}^{n-1} v_i q_{ik} = v_k,\ k = 0,\ldots,n-1$$

$$\iff (v_0,\ldots,v_{n-1})\cdot Q = (v_0,\ldots,v_{n-1}) \iff v\cdot Q = v. \qquad \square$$

Uit deze stelling volgt:
$v \in \mathbb{F}_q[X]$ met $\deg(v) < n$ en $v^q \equiv v$ modulo $f \iff v$ zit in de kern van $Q - I$ met I de $n \times n$ eenheidsmatrix.

We kunnen nu dus eenvoudig al deze v bepalen door de matrix $Q - I$ te diagonaliseren. Door $Q - I$ te diagonaliseren komen we automatisch nog meer te weten over f.

STELLING 4.2. *De rang van de kern van* $Q - I$ *is gelijk aan het aantal irreducibele factoren over* $\mathbb{F}_q$ *van* f.

BEWIJS. Veronderstel dat de kern van $Q - I$ wordt opgespannen door r onafhankelijke polynomen, d.w.z. rijvectoren opgevat als polynomen, $v_{(1)},\ldots,v_{(r)}$ met natuurlijk $r \leq n$. Dan volgt met Stelling 4.1 dat er q^r verschillende polynomen $v \in \mathbb{F}_q[X]$ bestaan met $\deg(v) < n$ en $v^q \equiv v$ modulo f, immers iedere lineaire combinatie van $v_{(1)}$ t/m $v_{(r)}$ van de vorm $\sum_{i=1}^{r} s_i \cdot v_{(i)}$, $s_i \in \mathbb{F}_q$ $(i = 1,\ldots,r)$, zit ook in de kern van $Q - I$.

Laat omgekeerd $f = \prod_{i=1}^{t} g_{(i)}$ de factorisatie van f over $\mathbb{F}_q$ zijn in irreducibele factoren, dus met $\gcd(g_{(i)},g_{(j)}) = 1$, $i \neq j$, omdat f kwadraatvrij is. Neem een willekeurige deelverzameling $\{s_1,\ldots,s_t\}$ van $\mathbb{F}_q$. De Chinese reststelling garandeert ons het bestaan van een uniek polynoom $w \in \mathbb{F}_q[X]$, zodat $w \equiv s_i$ modulo $g_{(i)}$, $i = 1,\ldots,t$, en $\deg(w) < \sum_{i=1}^{t} \deg(g_{(i)}) = n$.

$$\left.\begin{array}{l} s_i \in \mathbb{F}_q \Rightarrow s_i^q = s_i \Rightarrow w^q \equiv s_i^q = s_i \equiv w \text{ modulo } g_{(i)} \\ \text{de polynomen } g_{(i)} \text{ zijn relatief priem,} \\ i = 1,\ldots,t \end{array}\right\} \Rightarrow w^q \equiv w \text{ modulo } \prod_{i=1}^{t} g_{(i)},$$

oftewel $w^q \equiv w$ modulo f, en $\deg(w) < n$. $\{s_1,\ldots,w_t\} \subset \mathbb{F}_q$ was willekeurig gekozen, en het polynoom w volgt op unieke wijze uit de keuze van deze deelverzameling van $\mathbb{F}_q$, dus er zijn op deze manier q^t verschillende polynomen $w \in \mathbb{F}_q[X]$ te construeren met $w^q \equiv w$ modulo f en $\deg(w) < n$. Met het bovenstaande volgt nu $q^t \leq q^r$.

Om nu ook $t \geq r$ te bewijzen nemen we een willekeurig polynoom $v \in \mathbb{F}_q[X]$ uit de kern van $Q - I$. Dan $v^q - v \equiv 0$ modulo f, dus met Lemma 2.4 $\prod_{s \in \mathbb{F}_q} (v-s) \equiv 0$ modulo f, en omdat de $g_{(i)}$, $i = 1,\ldots,t$ factoren van f zijn

$$\prod_{s \in \mathbb{F}_q} (v-s) \equiv 0 \text{ modulo } g_{(i)}, \quad i = 1,\ldots,t.$$

De polynomen $v - s$, $s \in \mathbb{F}_q$, zijn relatief priem en $g_{(i)}$ is irreducibel, $i = 1,\ldots,t$; er moet dus voor $i = 1,\ldots,t$ een $s_i \in \mathbb{F}_q$ bestaan zodat $v - s_i \equiv 0$ modulo $g_{(i)}$. We hebben echter gezien dat, uitgaande van $v \equiv s_i$ modulo $g_{(i)}$ $(i = 1,\ldots,t)$, v uniek construeerbaar is met behulp van de Chinese reststelling, en dus is v één van de q^t op deze wijze te construeren polynomen. Conclusie: $q^r \leq q^t$. □

We hebben nu nog één stelling nodig voor we Berlekamp's factorisatiemethode kunnen geven.

STELLING 4.3. *Voor ieder paar polynomen* $(g_{(i)},g_{(j)})$, $1 \leq i < j \leq r$, *met* $g_{(i)}$ *en* $g_{(j)}$ *irreducibele factoren van* f *en* r *de rang van de kern van* $Q - I$, *is er een* $v_{(k)}$ *uit de basis van de kern van* $Q - I$ *zodat* $g_{(i)} \mid \gcd(f,v_{(k)}-s)$ *en* $g_{(j)} \nmid \gcd(f,v_{(k)}-s)$ *voor zekere* $s \in \mathbb{F}_q$.

BEWIJS. Zonder beperking van de algemeenheid mogen we $i = 1$ en $j = 2$ nemen. Neem $\{s_1,\ldots,s_r\} \subset \mathbb{F}_q$ met $s_1 \neq s_2$. Zoals in het bewijs van Stelling 4.2 kunnen we een polynoom $w \in \mathbb{F}_q[X]$ bepalen zodat $w \equiv s_i$ modulo $g_{(i)}$, $i = 1,2$. In het bijzonder $w \not\equiv s_1$ modulo $g_{(2)}$, en dus $g_{(1)} \mid \gcd(f,w-s_1)$ en $g_{(2)} \nmid \gcd(f,w-s_1)$. Volgens het bewijs van Stelling 4.2 is w een element van de kern van $Q - I$. De polynomen $v_{(1)},\ldots,v_{(r)}$ spannen deze kern op, dus is er een k, $1 \leq k \leq r$, zodat $g_{(1)} \mid \gcd(f,v_{(k)}-s)$ en $g_{(2)} \nmid \gcd(f,v_{(k)}-s)$, voor zekere $s \in \mathbb{F}_q$. □

Berlekamp's factorisatiemethode luidt nu als volgt:

- Bepaal een basis van de kern van $Q - I$, $\{v_{(1)},\ldots,v_{(r)}\}$, met $v_i \in \mathbb{F}_q[X]$, $i = 1,\ldots,r$, en r de rang van de kern. We mogen aannemen dat $v_{(1)} = 1$ en dat de overige $v_{(i)}$, $i = 2,\ldots,r$, niet triviaal en monisch zijn.
- Als $r = 1$, dan is f irreducibel (Stelling 4.2).
- $f = \prod_{s \in \mathbb{F}_q} \gcd(f,v_{(2)}-s)$ levert een niet-triviale factorisatie van f op omdat $v_{(2)}$ niet triviaal is en omdat $\deg(v_{(2)}) < n$. Als we op deze wijze direct r factoren van f vinden, dan zijn we volgens Stelling 4.2 klaar. Anders bepalen $\gcd(g,v_{(k)}-s)$, $s \in \mathbb{F}_q$, voor alle reeds gevonden factoren g van f en $k = 3,\ldots,r$. Stelling 4.3 garandeert nu dat we zo alle irreducibele factoren van f vinden.

VOORBEELD 4.1. $f = x^6 + 3x^5 + 2x^4 + 3x^3 - 3x^2 + 3x + 2 \in \mathbb{F}_7[X]$, het polynoom dat we al volledig gefactoriseerd hebben in Voorbeeld 2.1. We gaan Berlekamp's factorisatiemethode direct op f toepassen. Door de matrix $Q - I$ te

diagonaliseren vinden we de volgende drie polynomen die de kern van Q - I opspannen: $v_{(1)} = 1$, $v_{(2)} = x^4 + 2x^2 - x$, $v_{(3)} = x^5 - x^2 + 3x$. We zien dat $r = 3$, in overeenstemming met de factorisatie van f zoals die is verkregen in Voorbeeld 2.1. We gaan nu $\gcd(f,v_{(2)}-s)$, $s \in \mathbb{F}_7$, bepalen: $\gcd(f,v_{(2)}) = x^3 + 2x - 1$, $\gcd(f,v_{(2)}-1) = 1$, $\gcd(f,v_{(2)}-2) = x^2 + 2x - 2$, $\gcd(f,v_{(2)}-3) = 1$, $\gcd(f,v_{(2)}+3) = x + 1$. We hebben nu drie factoren gevonden, en omdat $r = 3$ concluderen we dat we klaar zijn.

OPMERKING 4.1. Het is duidelijk dat deze methode mogelijkerwijs niet meer efficiënt is als q groot is, want in de ggd berekening van f met $v_{(i)}$-s kunnen alle $s \in \mathbb{F}_q$ aan de beurt komen. Voor grote q is het daarom verstandig eerst te proberen zoveel mogelijk factoren te vinden met de partiële factorisatiemethoden. Het kwadraatvrij maken heeft nauwelijks last van de grootte van q, en bij de tweede partiële factorisatiemethode is de enige van q afhankelijke berekening de bepaling van x^q modulo f, en dat kan in O(log q) stappen gebeuren. Als door partiële factorisatie echter toch geen volledige factorisatie wordt gevonden en de boven beschreven methode wegens te grote q onbruikbaar is, zullen we onze toevlucht moeten nemen tot een van de volgende methoden.

We gaan Berlekamp's factorisatiemethode geschikt maken voor grote q. Als v een polynoom uit de kern van Q - I is, met v niet triviaal, dan weten we dat $f = \prod_{s \in \mathbb{F}_q} \gcd(f,v-s)$. Het vele rekenwerk bij Berlekamp's methode toegepast voor grote q wordt veroorzaakt door het feit dat we eventueel alle $s \in \mathbb{F}_q$ moeten proberen. Het merendeel van de te berekenen ggd's zal echter 1 zijn; we zouden ons liever beperken tot de deelverzameling S van $\mathbb{F}_q$ waarvoor geldt dat $\gcd(f,v-s) \neq 1$, $s \in S$. Deze $S \subset \mathbb{F}_q$ kennen we helaas niet, maar er is een manier om S te bepalen zonder alle lichaamselementen langs te moeten.

Uit $f = \prod_{s \in S} \gcd(f,v-s)$ volgt dat $f \mid \prod_{s \in S} (v-s)$. $\prod_{s \in S} (v-s)$ is een polynoom in v met graad #S. Geven we dit polynoom aan met H(v), dan geldt dus $H(v) \equiv 0$ modulo f. Omdat $H(v) = \prod_{s \in S} (v-s)$ volgt nu dat de nulpunten van H juist de elementen van S zijn. Om S te bepalen is het dus voldoende eerst H te berekenen en vervolgens de nulpunten van H. Als we H eenmaal hebben kunnen we met de methoden uit Hoofdstuk 3 de nulpunten bepalen; het enige wat we nu nog moeten doen is H bepalen.

Als r de rang van de kern van Q - I is, dan weten we dat $\#S \leq r$. De graad van H is #S en $H(v) \equiv 0$ modulo f, dus $1, v, v^2$ modulo $f, \ldots, v^r$ modulo f moeten lineair afhankelijk zijn. We kunnen #S en H dus bepalen door de

lineaire afhankelijkheidsrelatie van de kleinste graad van de residuen modulo f van de machten van v te bepalen: de coëfficiënten van H zijn de coëfficiënten van deze afhankelijkheidsrelatie.

VOORBEELD 4.2. Bij wijze van voorbeeld zullen we de methode toepassen om de $S \subset \mathbb{F}_7$ te bepalen behorende bij f en $v_{(2)}$ uit Voorbeeld 4.1. $v_{(2)} = x^4 + 2x^2 - x$, $v_{(2)}^2$ modulo $f = 3x^5 + x^4 - x^2 + x - 1$ en $v_{(2)}^3$ modulo $f = -3x^5 - 2x^4 - x^2 + 1$. We weten uit Voorbeeld 4.1 dat $r = 3$, daarom moeten $v_{(2)}^0$, $v_{(2)}^1$, $v_{(2)}^2$ en $v_{(2)}^3$ lineair afhankelijk zijn. Inderdaad geldt $v_{(2)}^3 + v_{(2)}^2 + v_{(2)} \equiv 0$ modulo f, dus $H(v) = v^3 + v^2 + v$. De nulpunten van H in $\mathbb{F}_7$ zijn 0, 2 en -3, in overeenstemming met wat we in Voorbeeld 4.1 gezien hebben.

OPMERKING 4.2. Welke nulpunt-bepalingsmethode we bij deze constructie gebruiken hangt af van de grootte van p. Voor p groot zullen we de probabilistische methode moeten kiezen, en daarmee is ook deze factorisatiemethode probabilistisch gevonden.

Tenslotte een methode speciaal geschikt voor factorisatie over $\mathbb{Z}/p\mathbb{Z}$ met p groot. Dankzij de resultaten van Hoofdstuk 2 mogen we aannemen dat het te factoriseren polynoom f in r verschillende irreducibele factoren van graad $\ell = n/r$ uiteenvalt. Bij deze methode hebben we een algoritme nodig die ons voor gegeven willekeurige ℓ een in $\mathbb{Z}/p\mathbb{Z}$ irreducibel monisch polynoom van graad ℓ levert. Omdat er tot nu toe nog geen snelle deterministische methode bekend is, waarmee we dit kunnen doen, zullen we een door Rabin voorgestelde probabilistische methode moeten gebruiken.

De methode werkt eenvoudigweg als volgt: kies willekeurig een monisch polynoom G van graad ℓ en bepaal met de partiële factorisatiemethoden uit Hoofdstuk 2 of G irreducibel is. Zo ja, dan zijn we klaar; zo nee, dan beginnen we opnieuw.

Niets garandeert ons dat we binnen een bepaald aantal stappen inderdaad een irreducibele G van graad ℓ hebben gevonden, maar het volgende lemma zegt ons dat het in de praktijk op een gegeven moment wel zal lukken.

LEMMA 4.1. *In $\mathbb{Z}/p\mathbb{Z}$ geldt:*

$$\frac{\textit{aantal irreducibele monische polynomen van graad } \ell}{\textit{aantal monische polynomen van graad } \ell} \approx \frac{1}{\ell}.$$

Dit betekent dat we bij iedere poging een kans van ongeveer $1/\ell$ op succes hebben. Voor ℓ groot lijkt dit dus nauwelijks aantrekkelijk.

Veronderstel $f = \prod_{i=1}^{r} g_{(i)}$ over $\mathbb{Z}/p\mathbb{Z}$, met $\deg(g_{(i)}) = \ell$, $i = 1,\ldots,r$. $g_{(i)}$ is een modulo p irreducibel polynoom van graad ℓ in $(\mathbb{Z}/p\mathbb{Z})[X]$, en heeft dus een nulpunt in $\mathbb{F}_q$, met $q = p^{\ell}$. Hieruit volgt dat ook f een nulpunt heeft in $\mathbb{F}_q$. Bepaal nu een nulpunt $\gamma \in \mathbb{F}_q$ van f met behulp van de probabilistische nulpunt-bepalingsmethode uit Hoofdstuk 3 (immers p is groot) en gebruik voor de aritmetiek in $\mathbb{F}_q$ een monisch modulo p irreducibel polynoom G, dat op de boven geschetste wijze verkregen is. Nu moet er een i, $1 \le i \le r$, zijn zodat $g_{(i)}(\gamma) = 0$, want $f(\gamma) = 0$. Dan geldt ook $g_{(i)}(\gamma)^p = g_{(i)}(\gamma^p) = 0$ en dus zijn $\gamma, \gamma^p, \ldots, \gamma^{p^{\ell-1}}$ alle nulpunten van $g_{(i)}$ in $\mathbb{F}_q$. Conclusie: $g_{(i)} = \prod_{i=0}^{\ell-1} (x - \gamma^{p^i}) \in (\mathbb{Z}/p\mathbb{Z})[X]$.

VOORBEELD 4.3. $f = x^4 - 8x^3 + 15x^2 - 2x - 1 \in \mathbb{Z}/127\mathbb{Z}$.
$\gcd(f, x^{127} - x) = 1$ en $\gcd(f, x^{127^2} - x) = f$, dus wegens Lemma 2.4 splitst f over $\mathbb{Z}/127\mathbb{Z}$ in twee irreducibele factoren van graad 2. Voor de aritmetiek in $\mathbb{F}_q$, $q = 127^2$, hebben we een monisch modulo 127 irreducibel polynoom G nodig. Het blijkt dat $G(T) = T^2 - T - 1$ voldoet, neem β met $G(\beta) = 0$.

We gaan nu een nulpunt van f in $\mathbb{F}_q$ zoeken met behulp van de probabilistische methode uit Hoofdstuk 3. In de eerste stap berekenen we $\gcd(x^{(q-1)/2} - 1, f) = x^2 - 3x + 1$ met cofactor $x^2 - 5x - 1$. Dit levert nog geen wortel van f, maar al wel de gevraagde factorisatie van f in $\mathbb{Z}/127\mathbb{Z}$. We zien dus dat het kan voorkomen dat we niet eens de wortels hoeven te bepalen, omdat de tussenresultaten bij de nulpuntbepaling al voldoende zijn.

Als we minder gelukkig waren geweest, hadden we een nulpunt van f in $\mathbb{F}_q$ moeten zoeken en daaruit een factor van f over $\mathbb{Z}/127\mathbb{Z}$ moeten reconstrueren. Laten we dit ten behoeve van het voorbeeld desondanks doen. Kies willekeurig een $s \in \mathbb{F}_q$ en bepaald $\gcd(x^{(q-1)/2} - 1, f(x-s))$. Voor $s = -33\beta + 43$ krijgen we als ggd $x - 2\beta - 28$. $x - 2\beta - 28$ is dus een factor van $f(x+33\beta-43)$ en dus is $x - 35\beta + 15$ een factor van $f(x)$ over $\mathbb{F}_q$. Hieruit reconstrueren we een factor van f over $\mathbb{Z}/127\mathbb{Z}$ door
$(x - (35\beta-15)) \cdot (x - (35\beta-15)^{127}) = (x - (35\beta-15)) \cdot (x + (35\beta-20)) = x^2 - 5x - 1$.

5. HET LIFTEN VAN EEN FACTORISATIE

In dit hoofdstuk behandelen we twee methoden om een factorisatie van f over $\mathbb{F}_q$ uit te breiden tot een factorisatie over $W_k(\mathbb{F}_q)$, $k \ge 1$ willekeurig. Dit proces staat bekend als het liften van een factorisatie. De eerste methode volgt uit het constructieve bewijs van het lemma van Hensel, waarin de existentie van een factorisatie in $W_k(\mathbb{F}_q)$ wordt aangetoond. Deze

constructie voert de factorisatie lineair omhoog, dat wil zeggen van $W_1(\mathbb{F}_q) \to W_2(\mathbb{F}_q) \to W_3(\mathbb{F}_q) \to \ldots \to W_k(\mathbb{F}_q)$. Het lemma van Zassenhaus, en in het bijzonder het bewijs ervan, levert een kwadratische uitbreiding van de constructie van Hensel. De Zassenhaus-methode gaat van $W_{2^0}(\mathbb{F}_q) \to W_{2^1}(\mathbb{F}_q) \to W_{2^2}(\mathbb{F}_q) \to \ldots \to W_{2^k}(\mathbb{F}_q)$, en is dus in aanzienlijk minder stappen klaar dan de eerste methode, als een factorisatie over $W_k(\mathbb{F}_q)$ wordt verlangd met k groot.

Onmisbaar bij het liften zijn de volgende twee algoritmen.

Algoritme 5.1. De uitgebreide Euclidische algoritme in $\mathbb{F}_q[X]$.
Gegeven polynomen g en h in $\mathbb{F}_q[X]$ berekent de algoritme op eenheden na unieke a, b en d in $\mathbb{F}_q[X]$ zodat $a \cdot g + b \cdot h = d$ in $\mathbb{F}_q[X]$, $\deg(a) < \deg(h) - \deg(d)$ en $\deg(b) < \deg(g) - \deg(d)$, waarbij d de ggd van g en h is.

Een gedetailleerde beschrijving van deze algoritme kan worden gevonden in KNUTH [12], blz. 302, door algoritme X toe te passen op de polynomen g en h in plaats van op de gehele getallen u en v, waarbij vooral de opmerking van G.H. Bradley ter harte moet worden genomen.

Algoritme 5.2.
Gegeven polynomen g, h, a, b, d en c in $(W_k(\mathbb{F}_q))[X]$, met $a \cdot g + b \cdot h = d$ en $d = \gcd(g,h)$ een deler van c over $W_k(\mathbb{F}_q)$. De algoritme berekent unieke a' en b' in $(W_k(\mathbb{F}_q))[X]$ zodat $a' \cdot g + b' \cdot h = c$ over $W_k(\mathbb{F}_q)$ en $\deg(a') < \deg(h) - \deg(d)$.

Bepaal q en r in $(W_k(\mathbb{F}_q))[X]$ zodat $a \cdot (c/d) = q \cdot (h/d) + r$, en neem $a' = r$ (dus $\deg(a') < \deg(h) - \deg(d)$) en $b' = b \cdot (c/d) + q \cdot (g/d)$. a' en b' voldoen want

$$\begin{aligned} a' \cdot g + b' \cdot h &= r \cdot g + b \cdot (c/d) \cdot h + q \cdot (g/d)\, h \\ &= (r + q \cdot (h/d)) \cdot g + b \cdot h \cdot (c/d) \\ &= a \cdot g \cdot (c/d) + b \cdot h \cdot (c/d) \\ &= d \cdot (c/d) = c \quad \text{over } W_k(\mathbb{F}_q). \end{aligned}$$

Uniciteit: stel a" en b" in $(W_k(\mathbb{F}_q))[X]$ voldoen ook, dan

$$a'\cdot g + b'\cdot h = a''\cdot g + b''\cdot h \Rightarrow (a'-a'')\cdot(g/d) = (b''-b')\cdot(h/d) \left.\begin{array}{l} \\ \gcd(g/d,h/d) = 1 \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} (h/d) \text{ is een deler van } (a'-a'') \\ \deg(a'-a'') < \deg(h) - \deg(d) \end{array}\right\} \Rightarrow \left.\begin{array}{l} a' = a'' \\ h/d \neq 0 \end{array}\right\} \Rightarrow b' = b''.$$

LEMMA 5.1 (HENSEL). *Laat* $f \in (\mathbb{Z}[\alpha])[X]$ *en* $\mathbb{F}_q$ *zodat* $lc(f) \neq 0$ *in* $\mathbb{F}_q$. *Als* $f = g_{(1)}\cdot h_{(1)}$ *over* $\mathbb{F}_q$ *met* $g_{(1)}$ *en* $h_{(1)}$ *in* $\mathbb{F}_q[X]$ *en* $(g_{(1)},h_{(1)}) = 1$, *dan bestaan er voor iedere* $k \geq 1$ $g_{(k)}$ *en* $h_{(k)}$ *in* $(W_k(\mathbb{F}_q))[X]$ *zodat* $f = g_{(k)}\cdot h_{(k)}$ *over* $W_k(\mathbb{F}_q)$, $g_{(k)} = g_{(1)}$ *in* $\mathbb{F}_q$ *en* $h_{(k)} = h_{(1)}$ *in* $\mathbb{F}_q$.

BEWIJS. Bepaal met behulp van algoritme 5.1 $a,b \in \mathbb{F}_q[X]$ zodanig dat $a\cdot g_{(1)} + b\cdot h_{(1)} = 1$, $\deg(a) < \deg(h_{(1)})$ en $\deg(b) < \deg(g_{(1)})$.
Stel nu dat er voor $j \geq 1$ polynomen $g_{(j)}$ en $h_{(j)}$ in $(W_j(\mathbb{F}_q))[X]$ bestaan zodat $f = g_{(j)}\cdot h_{(j)}$ over $W_j(\mathbb{F}_q)$, $g_{(j)} = g_{(1)}$ in $\mathbb{F}_q$ en $h_j = h_{(1)}$ in $\mathbb{F}_q$.

Dan bestaat er een polynoom $c_{(j)} \in \mathbb{F}_q[X]$ met $f - g_{(j)}\cdot h_{(j)} = p^j\cdot c_{(j)}$ over $W_{j+1}(\mathbb{F}_q)$. Bepaal met algoritme 5.2 $a_{(j)}$ en $b_{(j)}$ in $\mathbb{F}_q[X]$ zodanig dat $a_{(j)}\cdot g_{(1)} + b_{(j)}\cdot h_{(1)} = c_{(j)}$ over $\mathbb{F}_q$ met $\deg(a_{(j)}) < \deg(h_{(1)})$.
Definieer nu $g_{(j+1)} = g_{(j)} + p^j\cdot b_{(j)}$ en $h_{(j+1)} = h_{(j)} + p^j\cdot a_{(j)}$, dan inderdaad $g_{(j+1)} = g_{(1)}$ in $\mathbb{F}_q$, $h_{(j+1)} = h_{(1)}$ in $\mathbb{F}_q$, $g_{(j+1)}$ en $h_{(j+1)}$ in $(W_{j+1}(\mathbb{F}_q))[X]$ en bovendien

$$\begin{aligned} g_{(j+1)}\cdot h_{(j+1)} &= (g_{(j)} + p^j\cdot b_{(j)})\ (h_{(j)} + p^j\cdot a_{(j)}) \\ &= g_{(j)}\cdot h_{(j)} + p^j(a_{(j)}\cdot g_{(1)} + b_{(j)}\cdot h_{(1)}) \\ &= g_{(j)}\cdot h_{(j)} + p^j\cdot c_{(j)} \\ &= f \quad \text{over} \quad W_{j+1}(\mathbb{F}_q). \end{aligned}$$

□

Het is duidelijk dat uit dit bewijs een algoritme volgt om de factorisatie over $W_k(\mathbb{F}_q)$ ook daadwerkelijk te bepalen, uitgaande van een factorisatie over $\mathbb{F}_q$. Nog enkele opmerkingen over deze methode en een voorbeeld van een toepassing voordat we naar de kwadratische uitbreiding ervan gaan kijken.

OPMERKING 5.1. De methode zoals hier beschreven lift twee factoren tegelijk. Indien f over $\mathbb{F}_q$ nu in r ($r > 2$) factoren uiteenvalt, die relatief priem

zijn, kan de factorisatie van f over $W_k(\mathbb{F}_q)$ natuurlijk worden verkregen door r keer deze algoritme toe te passen op een factor en de bijbehorende cofactor. Door enkele eenvoudige wijzigingen in de algoritme aan te brengen is het voor f monisch ook mogelijk deze r factoren simultaan te liften, zonder gebruik te maken van de cofactoren.

OPMERKING 5.2. Bij implementatie van deze algoritme moet behalve het priemgetal p ook een monisch modulo p irreducibel polynoom $G \in (\mathbb{Z}/p\mathbb{Z})[X]$ bekend zijn, om de aritmetische bewerkingen in het lichaam $\mathbb{F}_q$, dan wel de afgeknotte Witt-ring $W_k(\mathbb{F}_q)$, te kunnen uitvoeren. Het kan voorkomen dat dit polynoom G een factor modulo p is van het minimumpolynoom F dat het algebraïsche getallenlichaam bepaalt. In dat geval moet bij toepassing van deze methode om f over $W_k(\mathbb{F}_q)$ te factoriseren eerst G worden gelift tot een factor over $\mathbb{Z}/p^k\mathbb{Z}$ van F (zie ook Hoofdstuk 9).

VOORBEELD 5.1. Laat $F(T) = T^2 + T + 1$ het minimumpolynoom zijn, met nulpunt α, en $f = x^3 + (11\alpha+1)x^2 - 25(\alpha+1)x + 30\alpha + 5 \in (\mathbb{Q}(\alpha))[X]$.
Veronderstel dat we de factorisatie van f over $\mathbb{F}_4$ hebben bepaald, waarbij de aritmetiek in $\mathbb{F}_4$ wordt bepaald door het modulo 2 irreducibele polynoom F. $f = (x+1)(x^2+\alpha x+1)$ over $\mathbb{F}_4$. Neem, als in het lemma van Hensel, $g_{(1)} = x+1$ en $h_{(1)} = x^2 + \alpha x + 1$.

We gaan de factorisatie van f over $W_4(\mathbb{F}_4)$ bepalen door drie stappen uit te voeren van de constructie van Hensel.

Voorbereiding: $a,b \in \mathbb{F}_q[X]$ met $a\cdot g_{(1)} + b\cdot h_{(1)} = 1$ geeft:
$a = (\alpha+1)x+\alpha$, $b = \alpha+1$.

$W_1(\mathbb{F}_4) \to W_2(\mathbb{F}_4)$: $g_{(1)}\cdot h_{(1)} = (x+1)\cdot(x^2+\alpha x+1) = x^3+(\alpha+1)x^2+(\alpha+1)x+1$, dus
$f - g_{(1)}\cdot h_{(1)} = 10\alpha x^2 - 26(\alpha+1)x + 30\alpha + 4$
$= 2\alpha x^2 + 2(\alpha+1)x + 2\alpha$ in $W_2(\mathbb{F}_4)$
$= 2(\alpha x^2 + (\alpha+1)x+\alpha)$.
Dus $c_{(1)} = \alpha x^2 + (\alpha+1)x + \alpha \in \mathbb{F}_4[X]$. We passen algoritme 5.2 toe om $a_{(1)},b_{(1)} \in \mathbb{F}_4[X]$ te bepalen met
$a_{(1)}\cdot g_{(1)} + b_{(1)}\cdot h_{(1)} = c_{(1)}$:
$a\cdot c_{(1)} = q\cdot h_{(1)} + r$ voor $q = x+\alpha+1$ en $r = 0$ en dus
$a_{(1)} = 0$ en $b_{(1)} = b\cdot c_{(1)} + q\cdot g_{(1)} = \alpha$.
We krijgen $g_{(2)} = g_{(1)} + 2\cdot b_{(1)} = x+2\alpha+1 \in (W_2(\mathbb{F}_4))[X]$ en
$h_{(2)} = h_{(1)} + 2\cdot a_{(1)} = x^2+\alpha x+1 \in (W_2(\mathbb{F}_4))[X]$.

$W_2(\mathbb{F}_4) \to W_3(\mathbb{F}_4)$: $f - g_{(2)}\cdot h_{(2)} = 8\alpha x^2 - 24(\alpha+1)x + 28\alpha + 4$
$= 4\alpha + 4$ in $W_3(\mathbb{F}_4)$, dus $c_{(2)} = \alpha+1 \in \mathbb{F}_4[X]$.

Toepassing van algoritme 5.2 geeft $a_{(2)} = \alpha x+1$ en $b_{(2)} = \alpha$ met $a_{(2)}\cdot g_{(1)} + b_{(2)}\cdot h_{(1)} = c_{(2)}$.
We krijgen $g_{(3)} = g_{(2)}+2^2\cdot b_{(2)} = x-2\alpha+1 \in (W_3(\mathbb{F}_4))[X]$ en
$h_{(3)} = h_{(2)}+2^2\cdot a_{(2)} = x^2-3\alpha x-3 \in (W_3(\mathbb{F}_4))[X]$.

$W_3(\mathbb{F}_4) \to W_4(\mathbb{F}_4)$: $f - g_{(3)}\cdot h_{(3)} = 16\alpha x^2 - (16\alpha+16)x + 24\alpha + 8$
$= 8\alpha+8$ in $W_4(\mathbb{F}_4)$, dus $c_{(3)} = \alpha+1 \in \mathbb{F}_4[X]$.
$c_{(3)} = c_{(2)}$, dus ook $a_{(3)} = a_{(2)}$ en $b_{(3)} = b_{(2)}$.
We krijgen $g_{(4)} = g_{(3)}+2^3\cdot b_{(3)} = x+6\alpha+1 \in (W_4(\mathbb{F}_4))[X]$ en
$h_{(4)} = h_{(3)}+2^3\cdot a_{(3)} = x^2+5\alpha x+5 \in (W_4(\mathbb{F}_4))[X]$.

We hebben nu de factorisatie van f over $W_4(\mathbb{F}_4)$ gevonden. Merk op dat $f = g_{(4)}\cdot h_{(4)}$ over $\mathbb{Q}(\alpha)$, oftewel we hebben de factorisatie van f over $\mathbb{Q}(\alpha)$ bepaald.

LEMMA 5.2 (ZASSENHAUS). *Laat* $f \in (\mathbb{Z}[\alpha])[X]$ *en* $\mathbb{F}_q$ *zodat* $lc(f) \neq 0$ *in* $\mathbb{F}_q$. *Als er* $g_{(j)}$, $h_{(j)}$, $a_{(j)}$ *en* $b_{(j)}$ *in* $(W_{2^j}(\mathbb{F}_q))[X]$ *bestaan zodat* $f = g_{(j)}\cdot h_{(j)}$ *over* $W_{2^j}(\mathbb{F}_q)$, $a_{(j)}\cdot g_{(j)} + b_{(j)}\cdot h_{(j)} = 1$ *over* $W_{2^j}(\mathbb{F}_q)$ *en* $lc(h_{(j)})$ *een eenheid in* $\mathbb{F}_q$, *dan kunnen we "in één stap"* $g_{(j+1)}, h_{(j+1)}, a_{(j+1)}, b_{(j+1)} \in (W_{2^{j+1}}(\mathbb{F}_q))[X]$ *bepalen zodat* $f = g_{(j+1)}\cdot h_{(j+1)}$ *over* $W_{2^{j+1}}(\mathbb{F}_q)$, $a_{(j+1)}\cdot g_{(j+1)} + b_{(j+1)}\cdot h_{(j+1)} = 1$ *over* $W_{2^{j+1}}(\mathbb{F}_q)$, $lc(h_{(j+1)})$ *een eenheid in* $\mathbb{F}_q$, $g_{(j+1)} = g_{(j)}$ *in* $W_{2^j}(\mathbb{F}_q)$ *en* $h_{(j+1)} = h_{(j)}$ *in* $W_{2^j}(\mathbb{F}_q)$.

BEWIJS. $f = g_{(j)}\cdot h_{(j)}$ over $W_{2^j}(\mathbb{F}_q)$, dus er bestaat een polynoom $c_{(j)} \in (W_{2^j}(\mathbb{F}_q))[X]$ met $f - g_{(j)}\cdot h_{(j)} = p^{2^j}\cdot c_j$ over $W_{2^{j+1}}(\mathbb{F}_q)$. Bepaal met algoritme 5.2 $h'_{(j)}$ en $g'_{(j)}$ in $(W_{2^j}(\mathbb{F}_q))[X]$ zodanig dat $h'_{(j)}\cdot g_{(j)} + g'_{(j)}\cdot h_{(j)} = c_{(j)}$ over $W_{2^j}(\mathbb{F}_q)$, met $\deg(h'_{(j)}) < \deg(h_{(j)})$.

Definieer nu $g_{(j+1)} = g_{(j)} + p^{2^j}g'_{(j)}$ en $h_{(j+1)} = h_{(j)} + p^{2^j}h'_{(j)}$, dan inderdaad $g_{(j+1)}, h_{(j+1)} \in (W_{2^{j+1}}(\mathbb{F}_q))[X]$, $g_{(j+1)} = g_{(j)}$ in $W_{2^j}(\mathbb{F}_q)$, $h_{(j+1)} = h_{(j)}$ in $W_{2^j}(\mathbb{F}_q)$, $lc(h_{(j+1)}) = lc(h_{(j)})$ is een eenheid in $\mathbb{F}_q$ wegens $\deg(h'_{(j)}) < \deg(h_{(j)})$ en bovendien

$$\begin{aligned} g_{(j+1)}\cdot h_{(j+1)} &= (g_{(j)} + p^{2^j}\cdot g'_{(j)})\cdot(h_{(j)} + p^{2^j}\cdot h'_{(j)}) \\ &= g_{(j)}\cdot h_{(j)} + p^{2^j}(h'_{(j)}\cdot g_{(j)} + g'_{(j)}\cdot h_{(j)}) \\ &= g_{(j)}\cdot h_{(j)} + p^{2^j}c_{(j)} \\ &= f \quad \text{over} \quad W_{2^{j+1}}(\mathbb{F}_q). \end{aligned}$$

Er bestaat nu een polynoom $r_{(j)} \in (W_{2^j}(\mathbb{F}_q))[X]$ met

$a_{(j)}\cdot g_{(j+1)} + b_{(j)}\cdot h_{(j+1)} = 1 + p^{2^j}r_{(j)}$ over $W_{2^{j+1}}(\mathbb{F}_q)$. Bepaal met algoritme 5.2 $a'_{(j)}$ en $b'_{(j)}$ in $(W_{2^j}(\mathbb{F}_q))[X]$ zodanig dat $a'_{(j)}\cdot g_{(j)} + b'_{(j)}\cdot h_{(j)} = r_{(j)}$ over $W_{2^{j+1}}(\mathbb{F}_q)$.

Definieer nu $a_{(j+1)} = a_{(j)} - p^{2^j}a'_{(j)}$ en $b_{(j+1)} = b_{(j)} - p^{2^j}b'_{(j)}$, dan $a_{(j+1)}, b_{(j+1)} \in (W_{2^{j+1}}(\mathbb{F}_q))[X]$ en

$$\begin{aligned} &a_{(j+1)}\cdot g_{(j+1)} + b_{(j+1)}\cdot h_{(j+1)} \\ &= (a_{(j)} - p^{2^j}a'_{(j)})\cdot g_{(j+1)} + (b_{(j)} - p^{2^j}b'_{(j)})\cdot h_{(j+1)} \\ &= a_{(j)}\cdot g_{(j+1)} + b_{(j)}\cdot h_{(j+1)} - p^{2^j}(a'_{(j)}\cdot g_{(j+1)} + b'_{(j)}\cdot h_{(j+1)}) \\ &= a_{(j)}\cdot g_{(j+1)} + b_{(j)}\cdot h_{(j+1)} - p^{2^j}\cdot r_{(j)} \\ &= 1 \quad \text{over} \quad W_{2^{j+1}}(\mathbb{F}_q). \end{aligned}$$

□

Net zoals uit het bewijs van het lemma van Hensel direct een algoritme volgt voor het lineaire liften, zo volgt uit dit bewijs een algoritme voor het kwadratische liften. Immers de in stap 1 benodigde $a_{(0)}$ en $b_{(0)}$ zijn weer te bepalen met algoritme 5.1.

OPMERKING 5.3. Net als bij de constructie van Hensel is het bij deze methode mogelijk de algoritme zodanig aan te passen voor monische f dat r factoren tegelijk worden gelift. Het is mij echter niet gelukt dit te doen zonder de cofactoren nodig te hebben. Deze wijziging heeft daarom in dit geval nauwelijks praktisch nut.

OPMERKING 5.4. Het hangt sterk van de toepassing af welk van beide methodes de voorkeur verdient. Als er echt ver moet worden gelift kan Zassenhaus het best worden gebruikt, maar als dat niet nodig is, en vooral als er veel factoren van een monische f moeten worden gelift, is Hensel te prefereren.

VOORBEELD 5.2. We nemen hetzelfde voorbeeld als bij de constructie van Hensel. $F(T) = T^2 + T + 1$, $F(\alpha) = 0$, $f = x^3 + (11\alpha+1)x^2 - 25(\alpha+1)x + 30\alpha + 5 \in (\mathbb{Q}(\alpha))[X]$. $f = (x+1)\cdot(x^2+\alpha x+1)$ over $\mathbb{F}_4$.
$g_{(0)} = x+1$, $h_{(0)} = x^2 + \alpha x + 1$, $a_{(0)} = (\alpha+1)x + \alpha$, $b_{(0)} = \alpha + 1$.

$W_{20}(\mathbb{F}_4) \to W_{21}(\mathbb{F}_4)$: zie Voorbeeld 5.1:

$c_{(0)} = \alpha x^2 + (\alpha+1)x + \alpha$, $h'_{(0)} = 0$, $g'_{(0)} = \alpha$,

$g_{(1)} = g_{(0)} + 2^{2^0} \cdot g'_{(0)} = x + 2\alpha + 1$,

$h_{(1)} = h_{(0)} + 2^{2^0} \cdot h'_{(0)} = x^2 + \alpha x + 1$, beide in $(W_{21}(\mathbb{F}_4))[X]$.

$a_{(0)} \cdot g_{(1)} + b_{(0)} \cdot h_{(1)} = 1 + 2((\alpha+1)x^2 + (\alpha+1)x - 1)$, dus

$r_{(0)} = (\alpha+1)x^2 + (\alpha+1)x - 1 \in (W_{20}(\mathbb{F}_4))[X]$.

We passen algoritme 5.2 toe en we krijgen

$a'_{(0)} = \alpha$, $b'_{(0)} = \alpha+1$ zodat

$a'_{(0)} \cdot g_{(0)} + b'_{(0)} \cdot h_{(0)} = r_{(0)}$ over $W_{20}(\mathbb{F}_4)$.

We krijgen nu

$a_{(1)} = a_{(0)} - 2^{2^0} \cdot a'_{(0)} = (\alpha+1)x - \alpha \in (W_{21}(\mathbb{F}_4))[X]$ en

$b_{(1)} = b_{(0)} - 2^{2^0} \cdot b'_{(0)} = -(\alpha+1) \in (W_{21}(\mathbb{F}_4))[X]$.

$W_{21}(\mathbb{F}_4) \to W_{22}(\mathbb{F}_4)$: $f - g_{(1)} \cdot h_{(1)} = 8\alpha x^2 - 24(\alpha+1)x + 28\alpha + 4$

$= 8\alpha x^2 + 8(\alpha+1)x - 4\alpha + 4$ in $W_{22}(\mathbb{F}_4)$

$= 4(2\alpha x^2 + 2(\alpha+1)x - \alpha + 1$, dus

$c_{(1)} = 2\alpha x^2 + 2(\alpha+1)x - \alpha + 1 \in (W_{21}(\mathbb{F}_4))[X]$.

We bepalen met algoritme 5.2 $h'_{(1)}$ en $g'_{(1)}$ met

$h'_{(1)} \cdot g_{(1)} + g'_{(1)} \cdot h_{(1)} = c_{(1)}$ over $W_{21}(\mathbb{F}_4)$:

$h'_{(1)} = \alpha x + 1$, $g'_{(1)} = \alpha$. We krijgen

$g_{(2)} = g_{(1)} + 2^{2^1} g'_{(1)} = x + 6\alpha + 1 \in (W_{22}(\mathbb{F}_4))[X]$ en

$h_{(2)} = h_{(1)} + 2^{2^1} h'_{(1)} = x^2 + 5\alpha x + 5 \in (W_{22}(\mathbb{F}_4))[X]$,

waarmee we de gevraagde factorisatie over $W_4(\mathbb{F}_4)$ hebben bepaald.

6. GRENZEN VOOR DE COËFFICIËNTEN VAN DE FACTOREN VAN POLYNOMEN IN $\mathbb{Z}[X]$

Bij het factoriseren van $f \in \mathbb{Z}[X]$ is het bijzonder nuttig te weten hoe groot de coëfficiënten van de factoren van f kunnen zijn. In Hoofdstuk 7 en 8 zullen we zien hoe we zo'n bovengrens kunnen gebruiken bij de factorisatie van polynomen over $\mathbb{Z}$. In dit hoofdstuk zullen we met betrekkelijk eenvoudige middelen een alleen van $f \in \mathbb{Z}[X]$ afhankelijke bovengrens voor de coëfficiënten van de factoren van f afleiden. De te behandelen lemmata en stellingen zijn afkomstig uit MIGNOTTE [16].

We gebruiken de volgende afkortingen: als $f \in \mathbb{C}[X]$, deg(f) = n, dan

$$\|f\| = \left(\sum_{i=0}^{n} |f_i|^2 \right)^{\frac{1}{2}}, \quad L(f) = \sum_{i=0}^{n} |f_i|.$$

LEMMA 6.1. $f \in \mathbb{C}[X]$, deg(f) = n, $c \in \mathbb{C}$, $c \neq 0$, *dan* $\|(x+c)f\| = |c| \|(x+\bar{c}^{-1}) \cdot f\|$.

BEWIJS. $g = (x+c) \cdot f = \sum_{i=0}^{n+1} (f_{i-1} + c \cdot f_i) \cdot x^i$ en

$$h = (x+\bar{c}^{-1}) \cdot f = \sum_{i=0}^{n+1} (f_{i-1} + \bar{c}^{-1} f_i) \cdot x^i.$$

$$\|g\|^2 = \sum_{i=0}^{n+1} |f_{i-1} + c \cdot f_i|^2 = \sum_{i=0}^{n+1} (f_{i-1} + c \cdot f_i) \cdot \overline{(f_{i-1} + c \cdot f_i)}$$

$$= \sum_{i=0}^{n+1} (f_{i-1} + c \cdot f_i) \cdot (\bar{f}_{i-1} + \bar{c} \cdot \bar{f}_i)$$

$$= \sum_{i=0}^{n+1} (|f_{i-1}|^2 + f_{i-1} \cdot \bar{c} \cdot \bar{f}_i + \bar{f}_{i-1} \cdot c \cdot f_i + |c|^2 \cdot |f_i|^2).$$

$$\|h\|^2 = \sum_{i=0}^{n+1} |f_{i-1} + \bar{c}^{-1} f_i|^2 = \sum_{i=0}^{n+1} (f_{i-1} + \bar{c}^{-1} f_i) \overline{(f_{i-1} + \bar{c}^{-1} f_i)}$$

$$= \sum_{i=0}^{n+1} (f_{i-1} + \bar{c}^{-1} f_i)(\bar{f}_{i-1} + c^{-1} \bar{f}_i)$$

$$= \sum_{i=0}^{n+1} (|f_{i-1}|^2 + f_{i-1} \cdot c^{-1} \cdot \bar{f}_i + \bar{f}_{i-1} \cdot \bar{c}^{-1} \cdot f_i + |c|^{-2} \cdot |f_i|^2).$$

Dus $|c|^2 \cdot \|h\|^2 = \sum_{i=0}^{n+1} (|c|^2 |f_{i-1}|^2 + f_{i-1} \cdot \bar{c} \cdot \bar{f}_i + \bar{f}_{i-1} \cdot c \cdot f_i + |f_i|^2) = \|g\|^2$

omdat $f_{-1} = f_{n+1} = 0$. □

LEMMA 6.2. $x_i \in \mathbb{C}$, i = 1,...,n, *met* $0 < |x_1| \leq \ldots \leq |x_t| < 1 \leq |x_{t+1}| \leq \ldots \leq |x_n|$, $t \geq 0$. *Als* $f = \prod_{i=1}^{n} (x - x_i)$ *en* $g = \prod_{i=1}^{t} (x - \bar{x}_i^{-1}) \cdot \prod_{i=t+1}^{n} (x - x_i)$, *dan*

$$\|f\| = \left| \prod_{i=1}^{t} x_i \right| \cdot \|g\|.$$

BEWIJS. Met volledige inductie naar t. t = 0 triviaal. Stel t > 0. Zij $\tilde{f} = f/(x-x_1)$, $\tilde{g} = g/(x-\bar{x}_1^{-1})$.

$$\|f\| = \|(x-x_1) \cdot \tilde{f}\| = |x_1| \cdot \|(x-\bar{x}_1^{-1}) \cdot \tilde{f}\| \quad \text{(Lemma 6.1)}$$

$$= |x_1| \cdot \left| \prod_{i=2}^{t} x_i \right| \cdot \|(x-\bar{x}_1^{-1}) \cdot \tilde{g}\| \quad \text{(inductie hypothese, want } \bar{x}_1^{-1} \geq 1)$$

$$= \left| \prod_{i=1}^{t} x_i \right| \cdot \|g\|.$$

□

STELLING 6.1. $f \in \mathbb{C}[X]$, $\deg(f) = n$, *dan zijn er* $x_i \in \mathbb{C}$, $i = 1,\ldots,n$, *zodat* $f = f_n \cdot \prod_{i=1}^{n} (x-x_i)$ *en* $|x_1| \le \ldots \le |x_t| < 1 \le |x_{t+1}| \le \ldots \le |x_n|$ *voor zekere* $t \ge 0$. *Bovendien* $\|f\|^2 \ge |f_n|^2 \cdot |\prod_{i=t+1}^{n} x_i|^2 + |f_0|^2 \cdot |\prod_{i=t+1}^{n} x_i|^{-2}$.

BEWIJS. De eerste bewering mogen we bekend veronderstellen, we beperken ons tot het bewijs van de tweede bewering.

Laat $x_1 \neq 0$. Definieer $g = f_n \cdot \prod_{i=1}^{t} (x-\bar{x}_i^{-1}) \cdot \prod_{i=t+1}^{n} (x-x_i)$. Pas Lemma 6.2 toe op f en g: $\|f\| = |\prod_{i=1}^{t} x_i| \cdot \|g\|$, en dus

$$(1) \qquad \|f\|^2 = |\prod_{i=1}^{t} x_i|^2 \cdot \left(\sum_{i=0}^{n} |g_i|^2 \right) \ge |\prod_{i=1}^{t} x_i|^2 \cdot (|g_n|^2 + |g_0|^2).$$

$|f_n| \cdot |\prod_{i=1}^{n} x_i| = |f_0|$ en $|g_n| = |f_n|$, dus

$$(2) \qquad |g_n| \cdot |\prod_{i=1}^{t} x_i| = |f_0| \cdot |\prod_{i=t+1}^{n} x_i|^{-1}.$$

$|g_0| = |f_n| \cdot |\prod_{i=1}^{t} \bar{x}_i^{-1} \cdot \prod_{i=t+1}^{n} x_i|$, dus

$$(3) \qquad |g_0| \cdot |\prod_{i=1}^{t} x_i| = |f_n| \cdot |\prod_{i=t+1}^{n} x_i|.$$

Uit (1), (2) en (3) volgt nu

$$\|f\|^2 \ge |f_0|^2 \cdot |\prod_{i=t+1}^{n} x_i|^{-2} + |f_n|^2 \cdot |\prod_{i=t+1}^{n} x_i|^2.$$

Laat nu $x_1 = \ldots = x_s = 0$ $(s \le t)$. Hieruit volgt onmiddellijk $f_0 = 0$, dus is het voldoende te bewijzen dat $\|f\|^2 \ge |f_n|^2 \cdot |\prod_{i=t+1}^{n} x_i|^2$. Door f te vervangen door f/x^s krijgen we

$$\|f\|^2 = \|f/x^s\|^2 \ge |f_n|^2 \cdot |\prod_{i=t+1}^{n} x_i|^2 + |f_s|^2 |\prod_{i=t+1}^{n} x_i|^{-2} \ge |f_n|^2 \cdot |\prod_{i=t+1}^{n} x_i|^2. \quad \square$$

GEVOLG 6.1. f als in Stelling 6.1, dan $\|f\| \ge |f_n| \cdot \prod_{i=t+1}^{n} |x_i|$. $\square$

LEMMA 6.3. f *als in Stelling* 6.1, *dan* $|f_i| \le \binom{n}{i} \cdot |f_n| \cdot \prod_{i=t+1}^{n} |x_i|$ *en* $L(f) \le 2^n \cdot |f_n| \cdot \prod_{i=t+1}^{n} |x_i|$.

BEWIJS. De eerste bewering eenvoudig met volledige inductie naar n, de tweede bewering volgt daar direct uit. $\square$

STELLING 6.2. $f, g_{(1)}, \ldots, g_{(r)}, h \in \mathbb{Z}[X]$ *met* $f = \prod_{i=1}^{r} g_{(i)} \cdot h$, *dan*

$$\prod_{i=1}^{r} L(g_{(i)}) \le 2^{\sum_{i=1}^{r} \deg(g_{(i)})} \cdot \|f\|$$

en

$$|g_{(i)j}| \le \binom{\deg(g_{(i)})}{j} \cdot \|f\|, \quad i = 1,\ldots,r.$$

BEWIJS. De nulpunten van de $g_{(i)}$, $i = 1,\ldots,r$, met absolute waarde ≥ 1 zijn ook nulpunten van f met absolute waarde ≥ 1. Pas de tweede bewering van Lemma 6.3 toe op iedere $g_{(i)}$, $i = 1,\ldots,r$, vermenigvuldig de resultaten en pas op de ontstane ongelijkheid Gevolg 6.1 toe, dan krijgen we met $|lc(h)| \ge 1$

$$\prod_{i=1}^{r} L(g_{(i)}) \le 2^{\sum_{i=1}^{r} \deg(g_{(i)})} \cdot \|f\|.$$

De tweede bewering volgt eenvoudig door toepassing van Lemma 6.3 en Gevolg 6.1. $\square$

OPMERKING 6.1. Merk op dat de tweede bewering van Stelling 6.2 ons de gevraagde bovengrens voor de coëfficiënten van de factoren van een polynoom in $\mathbb{Z}[X]$ geeft.

7. GGD BEREKENINGEN IN $\mathbb{Z}[X]$

Voordat we de resultaten van de vorige hoofdstukken kunnen toepassen bij de factorisatie van polynomen over $\mathbb{Z}$, moeten we ons bezig houden met het probleem van de ggd berekeningen in $\mathbb{Z}[X]$. Laten $f_{(1)}$ en $f_{(2)}$, met $\deg(f_{(1)}) \ge \deg(f_{(2)})$, twee polynomen zijn in $\mathbb{Z}[X]$ waarvan we de ggd willen bepalen. We mogen aannemen dat $\mathrm{cont}(f_{(1)}) = \mathrm{cont}(f_{(2)}) = 1$. Proberen we de gewone Euclidische algoritme op $f_{(1)}$ en $f_{(2)}$ toe te passen dan zullen we in het algemeen te maken krijgen met niet-monische polynomen, en het is vooralsnog niet duidelijk hoe we niet-monische polynomen op elkaar kunnen delen zonder niet-gehele coëfficiënten te krijgen. Staan we het gebruik van rationale getallen toe, dan hebben we verder geen moeilijkheden met het toepassen van de Euclidische algoritme, alleen blijven we zitten met de in de praktijk onaantrekkelijke, want veel rekentijd vergende, rationale aritmetiek. We krijgen de volgende algoritme.

Algoritme 7.1. Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$. Bereken $f_{(i+1)} = f_{(i-1)}$ modulo $f_{(i)}$ in $\mathbb{Q}[X]$, $i = 2,3,\ldots$, tot voor zekere $k \geq 1$, $f_{(k+1)} = 0$.
De ggd van $f_{(1)}$ en $f_{(2)}$ is $c \cdot f_{(k)} \in \mathbb{Z}[X]$ voor zekere $c \in \mathbb{Q}$.

Een op het eerste gezicht iets aantrekkelijker versie van deze algoritme wordt verkregen door alleen met monische polynomen te werken.

Algoritme 7.1'. Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$. $f'_{(i)} = f_{(i)}/lc(f_{(i)}) \in \mathbb{Q}[X]$, $i = 1,2$. Bereken $f_{(i+1)} = f'_{(i-1)}$ modulo $f'_{(i)}$ in $\mathbb{Q}[X]$, $f'_{(i+1)} = f_{(i+1)}/lc(f_{(i+1)})$, $i = 2,3,\ldots$, tot voor zekere $k \geq 1$, $f'_{(k+1)} = 0$.
De ggd van $f_{(1)}$ en $f_{(2)}$ is $c \cdot f'_{(k)} \in \mathbb{Z}[X]$ voor zekere $c \in \mathbb{Z}$.

VOORBEELD 7.1.

Alg. 7.1:

$$f_{(1)} = 2x^4 + x^3 + 4x^2 + 8x + 3,$$
$$f_{(2)} = 6x^3 + 5x^2 + 5x + 2,$$
$$f_{(3)} = \frac{26}{9}x^2 + \frac{71}{9}x + \frac{29}{9},$$
$$f_{(4)} = \frac{4968}{169}x + \frac{2484}{169},$$
$$f_{(5)} = 0 \Rightarrow \text{ggd} = 2x + 1.$$

Alg. 7.1':

$$f'_{(1)} = x^4 + \tfrac{1}{2}x^3 + 2x^2 + 4x + \tfrac{3}{2},$$
$$f'_{(2)} = x^3 + \frac{5}{6}x^2 + \frac{5}{6}x + \frac{1}{3},$$
$$f'_{(3)} = x^2 + \frac{71}{26}x + \frac{29}{26},$$
$$f'_{(4)} = x + \tfrac{1}{2},$$
$$f'_{(5)} = 0 \Rightarrow \text{ggd} = 2x + 1.$$

Wegens de gebleken praktische onbruikbaarheid zullen we deze algoritmen verder niet beschouwen. Een andere bekende manier om de Euclidische algoritme toepasbaar te maken voor polynomen in $\mathbb{Z}[X]$ is door gebruik te maken van pseudodeling in $\mathbb{Z}[X]$. Pseudodeling is een manier om twee polynomen in $\mathbb{Z}[X]$ op elkaar te delen zonder niet-gehele coëfficiënten te krijgen. Als $f,g \in \mathbb{Z}[X]$, $\deg(f) \geq \deg(g)$, waarbij we f willen delen door g, dan gaan we als volgt te werk: vermenigvuldig f eerst met $lc(g)^{\deg(f)-\deg(g)+1}$, en voer vervolgens de deling uit. Het is duidelijk dat alle coëfficiënten gedurende het deelproces nu geheel blijven. We noemen de resulterende q en r, met $lc(g)^{\deg(f)-\deg(g)+1} \cdot f = q \cdot g + r$ en $\deg(r) < \deg(g)$, het pseudoquotiënt en de pseudorest. Omdat we vaak alleen geïnteresseerd zijn in de pseudorest geven we dit ook wel aan met $r = f \text{ psmod } g$. We kunnen nu direct onze tweede ggd-algoritme formuleren.

Algoritme 7.2. EPRS (Euclidean Polynomial Remainder Sequence):
Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$. Bereken $f_{(i+1)} = f_{(i-1)}$ psmod $f_{(i)}$ in $\mathbb{Z}[X]$, $i = 2,3,\ldots$, tot voor zekere $k \geq 1$, $f_{(k+1)} = 0$.
De ggd van $f_{(1)}$ en $f_{(2)}$ is $pp(f_{(k)})$.

VOORBEELD 7.2.

EPRS:

$$f_{(1)} = 2x^4 + x^3 + 4x^2 + 8x + 3,$$
$$f_{(2)} = 6x^3 + 5x^2 + 5x + 2,$$
$$f_{(3)} = 104x^2 + 284x + 116,$$
$$f_{(4)} = 317952x + 158976,$$
$$f_{(5)} = 0 \Rightarrow \text{ggd} = 2x + 1.$$

Uit dit voorbeeld blijkt dat het gebruik van deze algoritme, wegens de door de pseudodeling veroorzaakte exponentiële coëfficiëntengroei, sterk te ontraden is. Een verbetering van de EPRS-algoritme krijgen we als we iedere pseudorest eerst primitief maken voor we verder gaan.

Algoritme 7.3. PPRS (Primitive PRS):
Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$. Bereken $f_{(i+1)} = pp(f_{(i-1)} \text{ psmod } f_{(i)})$ in $\mathbb{Z}[X]$, $i = 2,3,\ldots$, tot voor zekere $k \geq 1$, $f_{(k+1)} = 0$.
De ggd van $f_{(1)}$ en $f_{(2)}$ is $f_{(k)}$.

VOORBEELD 7.3.

PPRS:

$$f_{(1)} = 2x^4 + x^3 + 4x^2 + 8x + 3,$$
$$f_{(2)} = 6x^3 + 5x^2 + 5x + 2,$$
$$f_{(3)} = 26x^2 + 71x + 29,$$
$$f_{(4)} = 2x + 1,$$
$$f_{(5)} = 0 \Rightarrow \text{ggd} = 2x + 1.$$

Hier is de coëfficiëntengroei lineair (BROWN [3]), wat dat betreft is de PPRS-algoritme dus heel bruikbaar. Het in iedere stap primitief maken van de pseudorest gaat echter ten koste van vele ggd berekeningen en dat maakt de PPRS-algoritme in de praktijk inefficiënt. Het blijkt dat er goedkopere manieren zijn om aan een gehele factor van de pseudorest te komen waardoor de coëfficiëntengroei in de meeste gevallen lineair blijft. In chronologische volgorde staan deze verbeteringen van de EPRS-algoritme bekend onder de namen Reduced PRS (RPRS), Subresultant PRS (SPRS), Improved SPRS (ISPRS) en zelfs nog een verbeterde versie van deze laatste methode

die tenslotte ook maar ISPRS is genoemd. Beschrijvingen van deze gecompliceerde algoritmen moeten echter achterwege blijven; we zullen ze geen van alle gaan gebruiken. Geïnteresseerden kunnen we verwijzen naar de niet geringe hoeveelheid literatuur die er op dit gebied is (o.a. [3],[4],[5],[7], [8],[9],[10],[12],[19] en [24]).

We zullen nu een methode gaan bekijken die in de praktijk uitstekende resultaten aflevert en die in het bijzonder zeer efficiënt werkt als de gezochte ggd 1 is. In het kort komt deze methode hierop neer: bepaal de ggd van $f_{(1)}$ en $f_{(2)}$ modulo enkele priemgetallen en combineer de gevonden ggd's van gelijke en minimale graad met de Chinese reststelling.

STELLING 7.1. $f_{(1)}, f_{(2)}, g \in \mathbb{Z}[X]$ *met* $g = \gcd(f_{(1)}, f_{(2)})$. *Voor alle priemgetallen* p *met* $p \nmid lc(f_{(1)}) \cdot lc(f_{(2)})$ *geldt dat* $\deg(g_{(p)}) \geq \deg(g)$, *met* $g_{(p)}$ *de ggd van* $f_{(1)}$ *en* $f_{(2)}$ *uitgerekend modulo* p, *en voor slechts eindig veel* p *geldt dat* $\deg(g_{(p)}) > \deg(g)$.

BEWIJS. De eerste bewering van de stelling is triviaal, de tweede bewering volgt door te kijken naar de rij kopcoëfficiënten van $f_{(3)}, f_{(4)}, \ldots, f_{(k)}$ bij de EPRS-algoritme. Hier zullen modulo p alleen andere graden optreden als p een van de kopcoëfficiënten deelt. Dit kan maar voor eindig veel p het geval zijn. □

Algoritme 7.4. De modulaire ggd algoritme:

Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$.

Stap 1 - Kies p, $p \nmid lc(f_{(1)}) \cdot lc(f_{(2)})$, $g := \gcd(f_{(1)}, f_{(2)})$ modulo p, $\Pi p := p$.

Stap 2 - Test of g al aanleiding geeft tot de ggd van $f_{(1)}$ en $f_{(2)}$ over $\mathbb{Z}$:
neem $h = (lc(f_{(1)}) \cdot g)$ modulo Πp,
als $h \mid lc(f_{(1)}) \cdot f_{(1)}$ en $pp(h) \mid f_{(2)}$ dan is pp(h) de gezochte ggd,
ga anders verder met Stap 3

Stap 3 - Kies een nieuwe p, $p \nmid lc(f_{(1)}) \cdot lc(f_{(2)})$, $g_{(p)} = \gcd(f_{(1)}, f_{(2)})$ modulo p. We onderscheiden de volgende drie gevallen:
- $\deg(g_{(p)}) < \deg(g)$: $g := g_{(p)}$, $\Pi p := p$, ga naar Stap 2.
- $\deg(g_{(p)}) = \deg(g)$: combineer g en $g_{(p)}$ met de Chinese reststelling tot de unieke h met $h \equiv g$ modulo Πp, $h \equiv g_{(p)}$ modulo p en alle coëfficiënten van h zijn in absolute waarde $\leq \lfloor \frac{\Pi p \cdot p}{2} \rfloor$.
  $g := h$, $\Pi p := \Pi p \cdot p$, ga naar Stap 2.
- $\deg(g_{(p)}) > \deg(g)$: herhaal Stap 3.

Toelichting bij Stap 2: alle ggd's van $f_{(1)}$ en $f_{(2)}$ modulo een priemgetal zijn monisch en dus ook het hieruit met de Chinese reststelling geconstrueerde polynoom g. De ggd van $f_{(1)}$ en $f_{(2)}$ over $\mathbb{Z}$ hoeft allerminst monisch te zijn, dus we moeten op de een of andere manier uit deze g een niet-monische ggd zien te halen. Dit gaat op de in Stap 2 beschreven wijze, waarvan de verificatie verder triviaal is.

Het zal met Stelling 7.1 verder duidelijk zijn dat de modulaire ggd algoritme werkt: Πp is op een gegeven moment zo groot dat zeker alle coëfficiënten van de ggd van $f_{(1)}$ en $f_{(2)}$ over $\mathbb{Z}$ in absolute waarde kleiner of gelijk zijn aan $\lfloor \Pi p/2 \rfloor$.

OPMERKING 7.1. De priemgetallen in de modulaire ggd algoritme moeten om begrijpelijke redenen groot worden gekozen. Hoe eerder Πp groot is, des te eerder zijn we klaar.

OPMERKING 7.2. Het aantal keren dat Stap 2 wordt uitgevoerd kan in vele gevallen tot 1 worden teruggebracht door eerst de maximale grootte van de coëfficiënten van de ggd te bepalen, en Stap 3 zolang uit te voeren tot Πp groot genoeg is (Hoofdstuk 6).

VOORBEELD 7.4. Hoewel 127 geen groot priemgetal is, is het toch groot genoeg voor dit voorbeeld. $f_{(1)} = 2x^4 + x^3 + 4x^2 + 8x + 3$, $f_{(2)} = 6x^3 + 5x^2 + 5x + 2$,
$g = \gcd(f_{(1)} \cdot f_{(2)})$ modulo $127 = x - 63$.
$(lc(f_{(1)}) \cdot g)$ modulo $127 = (2x - 126)$ modulo $127 = 2x + 1$.
$2x+1 \mid 2 \cdot f_{(1)}$ en $pp(2x+1) = 2x+1 \mid f_{(2)} \Rightarrow$ ggd $= 2x + 1$.

Als laatste methode noemen we nog de EZGCD algoritme ([20],[33]), die gebaseerd is op de liftalgoritmen uit Hoofdstuk 5. De methode is voor polynomen in één variabele over $\mathbb{Z}$ minder efficiënt dan de modulaire ggd algoritme, en is in het algemeen minder geschikt voor niet-kwadraatvrije polynomen. We zullen ons daarom beperken tot een zeer globale schets.

Algoritme 7.5. EZGCD (Extended Zassenhaus GCD):

Gegeven $f_{(1)}$ en $f_{(2)}$ in $\mathbb{Z}[X]$.

- Bepaal de ggd $g_{(p)}$ van $f_{(1)}$ en $f_{(2)}$ modulo enkele priemgetallen p tot er voldoende zekerheid bestaat omtrent de graad van de ggd over $\mathbb{Z}$ (Stelling 7.1).
- Lift een $g_{(p)}$ van de goede graad tot een factor modulo p^k van $f_{(2)}$ voor k voldoende groot (zie Hoofdstuk 6), waarbij $g_{(p)}$ de rol speelt van de

$h_{(1)}$ resp. de $h_{(0)}$ in de formuleringen van de Lemmata 5.1 en 5.2.

- Dankzij de opmerkingen over de graad van $h_{(i)}$ in Lemmata 5.1 en 5.2 weten we dat de gelifte ggd monisch is. We kunnen dus op dezelfde wijze als bij de modulaire ggd algoritme de echte ggd reconstrueren.

OPMERKING 7.3. Merk op dat de liftalgoritmen alleen mogen worden toegepast als $\gcd(g_{(p)}, f_{(2)}/g_{(p)}) = 1$ over $\mathbb{Z}/p\mathbb{Z}$. Als dat niet het geval is kunnen we altijd nog proberen of $\gcd(g_{(p)}, f_{(1)}/g_{(p)}) = 1$ over $\mathbb{Z}/p\mathbb{Z}$ en als dat het geval is $g_{(p)}$ liften tot een factor modulo p^k van $f_{(1)}$. Indien echter beide ggd's ongelijk 1 zijn moeten we onze toevlucht nemen tot een "speciaal geval"-algoritme, die we hier verder niet zullen beschrijven.

8. FACTORISATIE VAN POLYNOMEN IN $\mathbb{Z}[X]$

We zijn nu zover dat we polynomen kunnen gaan factoriseren over $\mathbb{Z}$. We presenteren eerst het factorisatieschema, daarna zullen we de afzonderlijke stappen toelichten.

Laat $f \in \mathbb{Z}[X]$ het te factoriseren polynoom van graad n zijn.

Stap 1: We mogen veronderstellen dat f primitief en kwadraatvrij is.

Stap 2: Factoriseer $\tilde{f} = f$ modulo p over $\mathbb{Z}/p\mathbb{Z}$ voor een geschikt gekozen priemgetal p: $\tilde{f} = lc(\tilde{f}) \cdot \prod_{i=1}^{r} g_{(i)}$ modulo p.

Stap 3: Lift deze factorisatie tot een factorisatie over $\mathbb{Z}/p^k\mathbb{Z}$ voor een geschikt gekozen $k \geq 1$: $f = lc(f) \cdot \prod_{i=1}^{r} h_{(i)}$ modulo p^k.

Stap 4: Bepaal de factoren van f over $\mathbb{Z}$ door combinaties van factoren over $\mathbb{Z}/p^k\mathbb{Z}$ te proberen.

Toelichting

Stap 1: Door de ggd van de coëfficiënten van f weg te delen uit f kunnen we bereiken dat f primitief is. De in Hoofdstuk 7 beschreven modulaire ggd algoritme voor polynomen in $\mathbb{Z}[X]$ maakt het mogelijk f kwadraatvrij te maken door toepassing van Lemma 2.2, dat immers ook voor polynomen in $\mathbb{Z}[X]$ geldt. De situatie voor het kwadraatvrij maken is hier wat eenvoudiger dan in Hoofdstuk 2 omdat we hier niet met een karakteristiek $p \neq 0$ te maken hebben. Het is dan ook eenvoudig mogelijk door herhaalde ggd berekeningen een factorisatie van f in kwadraatvrije polynomen te verkrijgen, dat wil zeggen $f = \prod_{i=1}^{t} f_{(i)}^{i}$.

Stap 2: We beschouwen alleen priemgetallen p zodat $p \nmid lc(f)$ en zodat

f modulo p kwadraatvrij is, opdat wij in stap 3 de liftalgoritmen kunnen toepassen. Dergelijke priemen zijn altijd te vinden wegens het volgende resultaat uit VAN DER WAERDEN [25] (zie ook Opm. 8.1 aan het eind van dit hoofdstuk):

De resultante van f en g is nul $\Leftrightarrow$ f en g hebben een niet-constante factor gemeen. f is kwadraatvrij $\Rightarrow$ f en f' hebben geen niet-constante factor gemeen $\Rightarrow$ de discriminant van f is niet nul $\Rightarrow$ de discriminant van f is nul modulo slechts een eindig aantal priemen $\Rightarrow$ f en f' hebben modulo slechts een eindig aantal priemen een factor gemeen $\Rightarrow$ f is niet kwadraatvrij modulo slechts eindig veel priemen.

Het hangt nu sterk van $\deg(f) = n$ en de gekozen p af hoe we f over $\mathbb{Z}/p\mathbb{Z}$ gaan factoriseren. We kunnen de volgende algemene richtlijnen geven:

- n en p beide klein: dit is het eenvoudigste geval, pas direct Berlekamp's factorisatiemethode toe.
- n groot en p klein: het nadeel van Berlekamp's factorisatiemethode is nu dat we de kern van de $n \times n$ matrix $Q - I$ moeten bepalen, en dat kost $O(n^3)$ operaties. Het is daarom in dit geval aan te raden eerst de partiële factorisatiemethode uit Hoofdstuk 2 toe te passen, waarbij we eventueel n, en dus Q, direct kleiner kunnen maken door de ggd van f met $x^p - x$ te berekenen. We weten dankzij deze partiële factorisatie in ieder geval het aantal irreducibele factoren waarin f modulo p uiteenvalt en daarom is het vaak niet nodig dat we een volledige basis van de kern van $Q - I$ bepalen: we bepalen alleen een basisvector als we hem nodig hebben.
- p groot: onafhankelijk van n is het in dit geval nuttig om eerst de partiële factorisatiemethode toe te passen om het gebruik van de nu haast onvermijdelijke probabilistische algoritmen tot een minimum te beperken.

Stap 3: Omdat we p zo gekozen hebben dat de factorisatie van f modulo p kwadraatvrij is, mogen we de liftalgoritmen uit Hoofdstuk 5 toepassen. We bepalen k door k minimaal te nemen zodat

$$p^k \geq 2|lc(f)| \binom{\lfloor n/2 \rfloor}{\lfloor n/4 \rfloor} \left(\sum_{i=0}^{n} f_i^2 \right)^{\frac{1}{2}}$$

(zie Stelling 6.3). Zie voor de keuze van de liftalgoritme Opm.5.4.

Stap 4: We staan in deze stap voor het probleem de factoren modulo p^k zo bij elkaar te nemen dat ze samen juist een factor van f over $\mathbb{Z}$ vormen. Hierbij moet natuurlijk voor niet-monische f dezelfde deeltest worden gebruikt als in Stap 2 voor algoritme 7.4: als we willen proberen of $h_{(i_1)},\ldots,h_{(i_s)}$ samen een factor van f over $\mathbb{Z}$ vormen, dan testen we of $g = lc(f)\cdot\prod_{j=1}^{s} h_{(i_j)}$ modulo p^k deelt op $lc(f)\cdot f$ over $\mathbb{Z}$. Als de deling opgaat is pp(g) een factor van f over $\mathbb{Z}$.

Omdat we in het slechtste geval (bijv. irreducibiliteit) alle combinaties van factoren modulo p^k, met de graad van zo'n combinatie $\leq \lfloor n/2 \rfloor$, moeten proberen, vergt deze stap mogelijk een in r exponentieel aantal deelpogingen. Als r groot is kan dat er de oorzaak van zijn dat deze stap de bottleneck van de hele berekening wordt. Hier is niet altijd wat aan te doen, maar in het algemeen kunnen we de volgende voorzorgsmaatregelen nemen:

- bereken de factorisatie modulo p van f voor enkele priemgetallen p. Gebruik de p met het kleinste aantal factoren om te liften, en haal zoveel mogelijk informatie uit de verschillende factorisaties omtrent de mogelijke graden van de factoren van f over $\mathbb{Z}$. Voorbeeld: Stel dat f modulo p_1 in drie factoren van graad 2 factoriseert en dat f modulo p_2 in twee factoren van graad 3 factoriseert, dan zien we onmiddellijk dat f over $\mathbb{Z}$ irreducibel is;
- lift de factorisatie een slag verder dan nodig en probeer combinaties met te hoge coëfficiëntwaarden niet;
- probeer bij de testdeling eerst de constante coëfficiënten.

VOORBEELD 8.1. $f = 9x^5 + 9x^4 + 15x^3 + 6x^2 + 7x + 4$.

Stap 1: f is primitief en kwadraatvrij.

Stap 2: $p = 2$ voldoet aan $p \nmid lc(f) = 9$, en $\tilde{f} = f$ modulo $2 = x^5 + x^4 + x^3 + x$ is kwadraatvrij. $\tilde{f} = x\cdot(x+1)\cdot(x^3+x+1)$ modulo 2 is het resultaat van een directe toepassing van Berlekamp's factorisatiemethode.

Stap 3: $n = 5 \Rightarrow 2^k \geq 2\cdot 9\cdot\binom{2}{1}\cdot\sqrt{81+81+225+36+49+16} \approx 707 \Rightarrow k = 10$.
Door toepassing van de lineaire liftalgoritme vinden we
$f = 9(x+837)(x+188)(x^3+683x+683)$ modulo 1024.

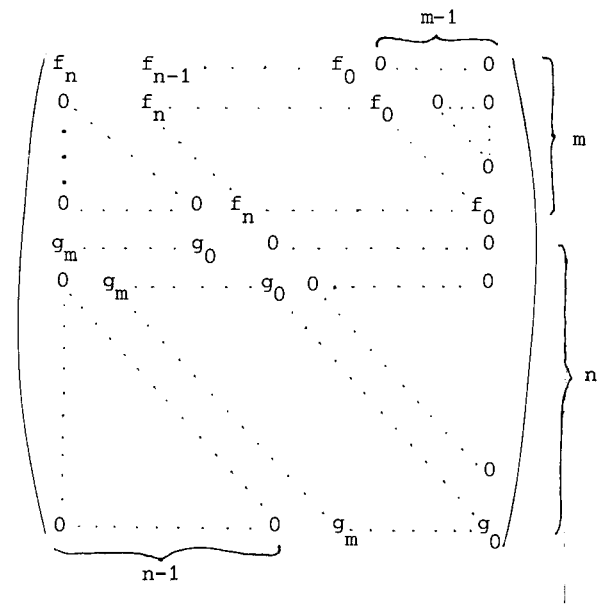
Stap 4: We proberen alleen combinaties van de factoren met graad ≤ 2:
$x + 837$: $9\cdot 837 = 7533 = 365$ modulo 1024 en $365 \nmid 9\cdot 4 = 36 \Rightarrow$ geen factor.
$x + 188$: $9\cdot 188 = 1692 = -356$ modulo 1024 en $356 \nmid 9\cdot 4 = 36 \Rightarrow$ geen factor.
$(x+837)(x+188)$: $9\cdot 837\cdot 188 = 1416204 = 12$ modulo 1024 en $12 \mid 9\cdot 4 = 36 \Rightarrow$

➡ dit levert mogelijkerwijs een factor op.
$9(x+837)(x+188) = 9x^2+9x+12$ modulo 1024 en dit deelt inderdaad op $9\cdot f$, dus $pp(9x^2+9x+12) = 3x^2 + 3x + 4$ is een factor over $\mathbb{Z}$ van f.

De cofactor van deze factor blijkt te zijn $3x^3 + x + 1$, en die moeten we ook kunnen krijgen door $pp(9(x^3+683x+683)$ modulo 1024).
$9(x^3+683x+683) = 9x^3+3x+3$ modulo 1024, dus dat klopt.

Hadden we ons gehouden aan de aanwijzing bij Stap 4 om wat meer priemen te proberen, dan hadden we in Stap 2 een factorisatie van $\tilde{f} = f$ modulo 7 in slechts twee polynomen gevonden: $\tilde{f} = 2\cdot(x^2+x-1)\cdot(x^3+5x+5)$ modulo 7.

OPMERKING 8.1. De resultante van f en g met $deg(f) = n$ en $deg(g) = m$ is gedefinieerd als de determinant van de volgende matrix:



De discriminant van f (discr(f)) is gedefinieerd als de resultante van f en f'.

9. FACTORISATIE VAN POLYNOMEN IN $(\mathbb{Q}(\alpha))[X]$

Laat f een polynoom zijn van de graad n in $(\mathbb{Q}(\alpha))[X]$. We mogen, omdat $\mathbb{Q}(\alpha)$ een lichaam is, aannemen dat f monisch is; er hoeft echter geenszins te gelden dat de coëfficiënten van f in $\mathbb{Z}[\alpha]$ liggen. Tot nu toe hebben we

alleen polynomen met gehele coëfficiënten beschouwd. Bij de factorisatie over $\mathbb{Q}(\alpha)$ wordt het echter in vele gevallen onvermijdelijk dat de coëfficiënten niet-triviale noemers krijgen, bijvoorbeeld al door het monisch maken van het te factoriseren polynoom f. We kunnen echter wel altijd een $d \in \mathbb{Z}$ vinden zodat de coëfficiënten van f alle in $\frac{1}{d}\mathbb{Z}[\alpha]$ liggen. Dankzij de volgende lemmata, die we zonder bewijs vermelden, kunnen we een $D \in \mathbb{Z}$ bepalen zodat alle coëfficiënten van alle monische factoren van f in $\frac{1}{D}\mathbb{Z}[\alpha]$ liggen.

LEMMA 9.1. *Zij* $b = \max\{d \in \mathbb{N}$ *zodat* $d^2 | \text{discr}(F)\}$, *dan* $\text{defect}(\alpha) | b$ *en dus* $R \subset \frac{1}{b}\mathbb{Z}[\alpha]$, *waarbij* $\text{defect}(\alpha) = \min\{d \in \mathbb{N} \mid R \subset \frac{1}{d}\mathbb{Z}[\alpha]\}$.

LEMMA 9.2. *Zij* $f \in (\frac{1}{d}R)[X]$, f *monisch, en laat* $f = g \cdot h$ *over* $\mathbb{Q}(\alpha)$ *met* g *en* h *monisch, dan* $g,h \in (\frac{1}{d}R)[X]$.

Veronderstel nu dat $f \in (\frac{1}{d}\mathbb{Z}[\alpha])[X]$, f monisch. Dan zeker $f \in (\frac{1}{d}R)[X]$, en dus wegens Lemma 9.2 zal iedere monische factor van f een element zijn van $(\frac{1}{d}R)[X]$. We kunnen ook een $c \in \mathbb{N}$ bepalen zodat $R \subset \frac{1}{c}\mathbb{Z}[\alpha]$; we nemen $\text{defect}(\alpha)$ als die bekend is en anders de b uit Lemma 9.1. Combineren we dit, dan is iedere monische factor van f een element van $\frac{1}{d \cdot c}\mathbb{Z}[\alpha]$, en we nemen $D = d \cdot c$.

VOORBEELD 9.1. $F(T) = T^2 + 7$, dat wil zeggen $\alpha = \sqrt{-7}$, en $f = x^2 + \frac{x}{2} + \frac{1}{2}$. De discriminant van F berekenen we door de resultante van F en F' te berekenen: $\begin{vmatrix} 1 & 0 & 7 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{vmatrix} = 28$, dus $b = 2$.

We krijgen $D = 2 \cdot 2 = 4$, en inderdaad $f = (x + \frac{1}{4} + \frac{\alpha}{4})(x + \frac{1}{4} - \frac{\alpha}{4})$.

Voordat we net als in Hoofdstuk 8 het schema voor de factorisatie van f over $\mathbb{Q}(\alpha)$ geven, eerst nog wat over ggd berekeningen in $(\mathbb{Q}(\alpha))[X]$.

Bij de berekening van de ggd van twee monische polynomen $f_{(1)}$ en $f_{(2)}$ in $(\mathbb{Q}(\alpha))[X]$ kunnen we natuurlijk gewoon de Euclidische algoritme gebruiken. We krijgen dan wel, zoals in Hoofdstuk 7 beschreven voor $\mathbb{Z}[X]$, te maken met coëfficiëntengroei of introductie van nieuwe noemers. Toch zal dit in enkele gevallen de aantrekkelijkste methode zijn, want de modulaire ggd algoritme heeft hier te kampen met een moeilijkheid die ook optreedt bij de factorisatie in $(\mathbb{Q}(\alpha))[X]$. Dit probleem komt voort uit het feit dat het minimumpolynoom F modulo een willekeurig gekozen priemgetal p helemaal niet

irreducibel hoeft te zijn, en dat er zelfs minimumpolynomen zijn die modulo iedere priem reducibel zijn. Bij de modulaire ggd algoritme is dit des te onaangenamer omdat we daar geïnteresseerd zijn in grote priemgetallen, en we hebben gezien dat we voor de factorisatie modulo grote priemgetallen alleen probabilistische methoden tot onze beschikking hebben. We zullen desondanks een mogelijke beschrijving van de modulaire ggd algoritme in $(\mathbb{Q}(\alpha))[X]$ geven, opdat het fenomeen van het factoriserende minimumpolynoom straks bij het factorisatieschema wat vertrouwder is.

Algoritme 9.1. Modulaire ggd algoritme in $(\mathbb{Q}(\alpha))[X]$.
Gegeven $f_{(1)}$ en $f_{(2)}$ in $(\frac{1}{D}\mathbb{Z}[\alpha])[X]$, monisch, met D zo gekozen dat ook alle monische factoren van $f_{(1)}$ en $f_{(2)}$ in $(\frac{1}{D}\mathbb{Z}[\alpha])[X]$ liggen.

Stap 1 - Kies p, $p \nmid D$, zodat F modulo p in zo weinig mogelijk factoren uiteen valt, $F = \prod_{i=1}^{t} F_{(i)}$ modulo p, en zodat de graden van $\gcd(D^{-1}$ modulo $p \cdot (f_{(1)} \cdot D)$, D^{-1} modulo $p \cdot (f_{(2)} \cdot D))$ in $\mathbb{F}_{q_i}[X]$, $i = 1,\ldots,t$, overeenstemmen, waarbij de aritmetiek in $\mathbb{F}_{q_i}$ bepaald wordt door p en $F_{(i)}$. Combineer deze t ggd's met de Chinese reststelling tot een ggd g van D^{-1} modulo $p \cdot (f_{(1)} \cdot D)$ en D^{-1} modulo $p \cdot (f_{(2)} \cdot D)$ modulo p en F. $\Pi p := p$.

Stap 2 - Test of g al aanleiding geeft tot de ggd van $f_{(1)}$ en $f_{(2)}$ over $\mathbb{Q}(\alpha)$. Construeer hiertoe een polynoom $h \in (\frac{1}{D}\mathbb{Z}[\alpha])[X]$ door $h_i = ((g_i \cdot D) \text{ modulo } \Pi p)/D$. Als h deelt op $f_{(1)}$ en $f_{(2)}$ over $\mathbb{Q}(\alpha)$, dan is h de gezochte ggd, anders gaan we verder met Stap 3.

Stap 3 - Kies een nieuwe p en beschouw dezelfde drie gevallen als in algoritme 7.4.

Merk op dat we in Stap 1 en Stap 2 meteen zien hoe we de noemer D door modulo p te werken kwijtraken en terugkrijgen.

Ook de EZGCD-algoritme kunnen wij gebruiken voor ggd berekeningen in $(\mathbb{Q}(\alpha))[X]$. We geven daar geen beschrijving van, met behulp van algoritme 7.5 en algoritme 9.1 is die eenvoudig te bedenken.

Nu dan eindelijk het schema voor de factorisatie in $(\mathbb{Q}(\alpha))[X]$ van een monisch polynoom $f \in (\frac{1}{D}\mathbb{Z}[\alpha])[X]$ met D als boven.

Stap 1 - We mogen aannemen dat f kwadraatvrij is.

Stap 2 - Factoriseer F over $\mathbb{Z}/p\mathbb{Z}$ voor een geschikt gekozen priemgetal p: $F = \prod_{i=1}^{t} F_{(i)}$ modulo p.

Stap 3 - Factoriseer f over $\mathbb{F}_{q_i}$ met $q_i = p^{\deg(F_{(i)})}$, $i = 1,\ldots,t$, waarbij

de aritmetiek in $\mathbb{F}_{q_i}$ bepaald wordt door p en $F_{(i)}$.

Stap 4 - Lift de factorisatie van F tot een factorisatie over $\mathbb{Z}/p^k\mathbb{Z}$ voor een geschikt gekozen $k \geq 1$: $F = \prod_{i=1}^{t} F_{(i)}^k$ modulo p^k.

Stap 5 - Lift de factorisaties van f tot factorisaties over $W_k(\mathbb{F}_{q_{ik}})$, $i = 1,\ldots,t$, waarbij de aritmetiek in $\mathbb{F}_{q_{ik}}$ bepaald wordt door p en $F_{(i)}^k$.

Stap 6 - Bepaal uit deze t factorisaties van f de factoren van f over $\mathbb{Q}(\alpha)$.

Toelichting

Stap 1 - Lemma 2.2 geldt ook voor polynomen in $(\mathbb{Q}(\alpha))[X]$, dus deel de, met een geschikt gekozen ggd algoritme berekende, ggd van f en f' weg uit f.

Stap 2 - Kies p priem, $p \nmid D$, p liefst klein, en zodat F modulo p zo weinig mogelijk factoren heeft en kwadraatvrij is. Bovendien moet p zo worden gekozen dat f kwadraatvrij blijft in $\mathbb{F}_{q_i}$, $i = 1,\ldots,t$. Dat een dergelijke p altijd te vinden is volgt uit eenzelfde soort redenering als in Hoofdstuk 8, Stap 2.

Stap 3 - Dit kan worden gedaan met de methoden uit Hoofdstuk 4. De noemer D van f kunnen we kwijtraken door f eerst met D en vervolgens met D^{-1} modulo p te vermenigvuldigen.

Stap 4 - Het enige probleem dat we hier nog moeten oplossen is het bepalen van k. Net als in Hoofdstuk 6 gedaan is voor polynomen in $\mathbb{Z}[X]$ kunnen we een bovengrens aangeven voor de coëfficiënten van de monische factoren van f. Zonder bewijs vermelden we het volgende resultaat:

Zonder bewijs vermelden we het volgende resultaat:
Laat $f_i = \sum_{j=0}^{m-1} \frac{c_{ij}}{D} \alpha^j$, $c_{ij} \in \mathbb{Z}$, $D \in \mathbb{Z}$, $i = 0,\ldots,n-1$, $f_n = 1$; kies dan k zodat

$$p^k \geq \left(2Dm!\binom{\lfloor n/2 \rfloor}{\lfloor n/4 \rfloor}\left(\sum_{\ell=0}^{m-1} |F_\ell|\right)^{m(m-1)/2} \cdot \left(\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} |c_{ij}|\left(\sum_{\ell=0}^{m-1} |F_\ell|\right)^j / D\right)^{\lfloor n/4 \rfloor}\right) \Big/ \sqrt{\text{Discr}(F)}.$$

Zie voor een bewijs WEINBERGER [31] of WANG [27].

Stap 5 - We mogen de liftalgoritmen toepassen omdat f kwadraatvrij is over alle lichamen $\mathbb{F}_{q_{ik}}$. De noemer van f raken we in dit geval kwijt door f eerst met D en vervolgens met D^{-1} modulo p^k te vermenigvuldigen.

Stap 6 - Dit is net als bij de factorisatie in $\mathbb{Z}[X]$ de bottleneck van de algoritme. De situatie is zelfs nog aanmerkelijk gecompliceerder, want we hebben nu t factorisaties van f waaruit we de factoren van f over $\mathbb{Q}(\alpha)$ moeten samenstellen. We zullen bij wijze van voorbeeld schetsen wat we moeten doen om de lineaire factoren van f te bepalen:

- Als er een i is, $1 \leq i \leq t$, zodat f over $W_k(\mathbb{F}_{q_{ik}})$ geen lineaire factoren heeft, dan heeft f over $\mathbb{Q}(\alpha)$ ook geen lineaire factoren.
- Neem anders voor iedere i, $i = 1,\ldots,t$, een lineaire factor van f over $W_k(\mathbb{F}_{q_{ik}})$, combineer dit t-tal tot een factor g van f modulo p^k en F met behulp van de Chinese reststelling. Reconstrueer de noemer D van g door g te vermenigvuldigen met D modulo p^k, en door g vervolgens de noemer D te geven, en test of het resultaat een deler van f is over $\mathbb{Q}(\alpha)$. Dit moet voor ieder t-tal lineaire factoren worden gedaan.

Het moge nu duidelijk zijn wat we moeten doen om de factoren van f over $\mathbb{Q}(\alpha)$ van graad d te bepalen: bepaal alle mogelijke factoren van graad d van f over $W_k(\mathbb{F}_{q_{ik}})$, $i = 1,\ldots,t$, die zelf dus weer combinaties kunnen zijn van factoren van f over $W_k(\mathbb{F}_{q_{ik}})$. Combineer ieder t-tal van factoren van graad d met de Chinese reststelling tot een factor van f modulo p^k en F, reconstrueer hiervoor de noemer D en test of het een deler van f over $\mathbb{Q}(\alpha)$ is.
Ook hier geldt: probeer in Stap 1 t/m 5 zoveel mogelijk informatie te krijgen over de mogelijke graden van de factoren van f over $\mathbb{Q}(\alpha)$ en probeer t en het aantal factoren van f over $\mathbb{F}_{q_i}$ door geschikte p keuze zo laag mogelijk te krijgen.

LITERATUUR

[1] BERLEKAMP, E.R., *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[2] BERLEKAMP, E.R., *Factoring polynomials over large finite fields*, Math. Comp. 24 (1970), 713-735.

[3] BROWN, W.S., *On Euclid's algorithm and the computation of polynomial*

greatest common divisors, J. ACM 18 (1971), 478-504.

[4] BROWN, W.S. & J.F. TRAUB, *On Euclid's algorithm and the theory of subresultants*, J.ACM. 18 (1971), 505-514.

[5] BROWN, W.S., *The subresultant PRS algorithm*, ACM Transactions on Mathematical Software 4 (1978), 237-249.

[6] CLAYBROOK, B.G., *A new approach to the symbolic factorization of multivariate polynomials*, Artificial Intelligence 7 (1976), 203-241.

[7] COLLINS, G.E., *Polynomial remainder sequences and determinants*, Amer. Math. Montly 73 (1966), 708-712.

[8] COLLINS, G.E., *Subresultants and reduced polynomial remainder sequences*, J.ACM 14 (1967), 128-142.

[9] COLLINS, G.E., *The calculation of multivariate polynomial resultants*, J. ACM 18 (1971), 515-532.

[10] COLLINS, G.E., *Computer algebra of polynomials and rational functions*, Amer. Math. Monthly 80 (1973), 725-755.

[11] HOROWITZ, E., *Modular arithmetic and finite field theory: a tutorial*, Proceedings of the 1971 ACM Symposium on symbolic and algebraic manipulation 188-194.

[12] KNUTH, D.E., *The art of computer programming*, Vol. 2, Addison-Wesley, Reading, Mass. 1969.

[13] LIPTON, J.D., *Chinese remainder and interpolation algorithms*, Proceedings of the 1971 ACM Symposium on symbolic and algebraic manipulation, 372-391.

[14] MAHLER, K., *On application of Jensen's formula to polynomials*, Mathematika 7 (1960), 98-100.

[15] McELIECE, R.J., *Factorization of polynomials over finite fields*, Math. Comp. 23 (1969), 861-867.

[16] MIGNOTTE, M., *An inequality about factors of polynomials*, Math. Comp. 28 (1974), 1153-1157.

[17] MIGNOTTE, M., *Some problems about polynomials*, Proceedings of the 1976 ACM Symposium on symbolic and algebraic computation, 227-228.

[18] MOENCK, R.T., *Practical fast polynomial multiplication*, Proceedings of the 1976 ACM Symposium on symbolic and algebraic computation, 136-148.

[19] MOSES, J., *The evolution of algebraic manipulation algorithms*, Information Processing 1974, 483-488.

[20] MOSES, J. & D.Y.Y. YUN, *The EZGCD algorithm*, Proceedings of the ACM (1973), 159-166.

[21] MUSSER, D.R., *Multivariate polynomial factorization*, J. ACM 22 (1975), 291-308.

[22] MUSSER, D.R., *On the efficiency of a polynomial irreducibility test*, J. ACM 25 (1978), 271-282.

[23] RABIN, M.O., *Probabilistic algorithms*, 18 425 theory of computers, fall 1977.

[24] TEER, F., *Simplificatie en het berekenen van grootste gemene delers van polynomen*, Doctoraalscriptie, 1974.

[25] VAN DER WAERDEN, B.L., *Moderne algebra*, Springer, Berlin, 1931.

[26] WANG, P.S. & L.P. ROTHSCHILD, *Factoring multivariate polynomials over the integers*, Math. Comp. 29 (1975), 935-950.

[27] WANG, P.S., *Factoring multivariate polynomials over algebraic number fields*, Math. Comp. 30 (1976), 324-336.

[28] WANG, P.S., *An efficient squarefree decomposition algorithm*, SIGSAM (1977), 4-6.

[29] WANG, P.S., *An improved multivariate polynomial factoring algorithm*, Math. Comp. 32 (1978), 1215-1231.

[30] WANG, P.S. & B.M. TRAGER, *New algorithms for polynomial squarefree decomposition over the integers*, SIAM J. Comput. 8 (1979), 300-305.

[31] WEINBERGER, P.J. & L.P. ROTHSCHILD, *Factoring polynomials over algebraic number fields*, ACM Transactions on Mathematical Software 2 (1976), 335-350.

[32] WILLETT, M., *Factoring polynomials over a finite field*, SIAM J.Appl. Math. 35 (1978), 333-337.

[33] YUN, D.Y.Y., *The Hensel lemma in algebraic manipulation,* Ph.D. Thesis, MIT, 1974.

[34] YUN, D.Y.Y., *On squarefree decomposition algorithms,* Proceedings of the 1976 ACM Symposium on symbolic and algebraic computation, 26-35.

[35] YUN, D.Y.Y., *Algebraic algorithms using p-adic constructions,* Proceedings of the 1976 ACM Symposium on symbolic and algebraic computation, 248-259.

[36] ZASSENHAUS, H., *On Hensel factorization,* I. Journal of number theory 1 (1969), 291-311.

[37] ZIMMER, H.G., *Computational problems, methods, and results in algebraic number theory,* Springer, Berlin, 1972.