Modellen voor eindige lichamen

Eggermont, R.H.

R.H. Eggermont
Korteweg 4A
3223 BK Hellevoetsluis
reggermo@math.leidenuniv.nl

Modellen voor eindige lichamen

Bachelorscriptie, 10 juni 2009

Scriptiebegeleider: prof.dr. H.W. Lenstra

Mathematisch Instituut, Universiteit Leiden

Inhoudsopgave

1 Inleiding

Eindige lichamen zijn veel bestudeerd in theoretische context. Als gevolg hiervan kunnen we bijvoorbeeld in polynomiale tijd testen of grote getallen priem zijn en of een polynoom over een eindig lichaam reducibel is of niet (waarbij gedeeltelijke factorisatie vaak direct mogelijk is). Om deze theorie toe te passen op een gegeven eindig lichaam, is echter een model nodig voor het eindige lichaam, dat bij voorkeur zo efficiënt mogelijk werkt. En wellicht willen we ook in polynomiale tijd kunnen testen of een gegeven model inderdaad een lichaam geeft.

Wat is er nu nodig voor een model? Aangezien we weten dat ieder eindig lichaam van de vorm $\mathbb{F}_{p^n}$ is voor zeker priemgetal $p \in \mathbb{Z}_{>0}$ en geheel getal $n \in \mathbb{Z}_{>0}$, lijkt het handig om uit te gaan van een additieve groep isomorf met $(\mathbb{Z}/p\mathbb{Z})^n$ voor zekere $p, n \in \mathbb{Z}_{>0}$, waarop een vermenigvuldiging is gedefinieerd op de basiselementen van de standaardbasis (wat met behulp van distributiviteit een vermenigvuldiging definieert tussen elk tweetal elementen van $(\mathbb{Z}/p\mathbb{Z})^n$). Er lijken hier al vrij sterke aannames gemaakt te worden; bijvoorbeeld het feit dat al een basis gegeven is, lijkt relatief sterk. Maar om bewerkingen snel uit te kunnen voeren, blijkt dit vrijwel altijd vereist te zijn; als voor ieder element de optelling en vermenigvuldiging met ieder ander element gegeven moeten zijn, is de hoeveelheid informatie veel te groot om in polynomiale tijd te testen.

In deze scriptie zal onder andere uitgelegd worden wat precies wordt bedoeld met "polynomiale tijd". Er zal enige achtergrond over algoritmen gegeven worden, en er zullen verschillende modellen van eindige lichamen besproken worden, met de nadruk op het eerder genoemde model. En de volgende stelling zal bewezen worden:

Stelling 1.1. *Er bestaat een algoritme dat voor $p, n \in \mathbb{Z}_{>0}$ in polynomiale tijd bepaalt of de abelse groep $(\mathbb{Z}/p\mathbb{Z})^n$, met gegeven vermenigvuldiging op de basiselementen, een lichaam is.*

Het bewijs zal gegeven worden door een expliciet algoritme te geven, en daarvan de snelheid te bepalen.

2 Artin-ringen

Onder een ring wordt een abelse groep verstaan met daarop een associatieve, *commutatieve*, distributieve vermenigvuldiging met een eenheidselement voor deze vermenigvuldiging gedefinieerd.

Lemma 2.1. *Zij R een ring, zij $\mathfrak{p} \subset R$ een priemideaal en laat $I_1, \ldots, I_n \subset R$ idealen zijn ($n \in \mathbb{Z}_{>0}$). Er geldt: $\mathfrak{p} \supseteq I_1 \cap I_2 \cap \ldots \cap I_n$ dan en slechts dan als $\mathfrak{p} \supseteq I_k$ voor zekere $k \in \{1, 2, \ldots, n\}$.*

Bewijs. $\Leftarrow$: Triviaal.

$\Rightarrow$: Stel $\mathfrak{p} \not\supseteq I_k$ voor alle $k \in \{1, 2, \ldots, n\}$. Kies $i_k \in I_k$ zodat $i_k \notin \mathfrak{p}$ voor $k \in \{1, 2, \ldots, n\}$. Er geldt $i_1 \cdot i_2 \cdot \ldots \cdot i_n \in (I_1 \cap I_2 \cap \ldots \cap I_n) \subseteq \mathfrak{p}$ omdat $I_1, \ldots, I_n$ idealen zijn. Maar dit is in tegenspraak met het feit dat $\mathfrak{p}$ priem is. □

Definitie 2.2. *Zij R een ring. Het* nilradicaal *van R is $\{x \in R : \exists n \in \mathbb{Z}_{>0}$ zodat $x^n = 0\}$. Notatie:* $\mathrm{nil}(R)$.

De ring R heet gereduceerd *indien* $\mathrm{nil}(R) = \{0\}$.

Lemma 2.3. *Zij R een ring. Er geldt:*

$$\bigcap_{\mathfrak{p} \subset R \,:\, \mathfrak{p} \text{ priem}} \mathfrak{p} = \mathrm{nil}(R)$$

Bewijs. Als $R = \{0\}$, is de genoemde doorsnede de lege doorsnede, ofwel $\bigcap_{\mathfrak{p} \subset R \,:\, \mathfrak{p}\ priem} \mathfrak{p} = R = \mathrm{nil}(R)$. Stel $R \neq \{0\}$.

$\supseteq$: Stel $x \in \mathrm{nil}(R)$. Laat $n \in \mathbb{Z}_{>0}$ zodat $x^n = 0$. Zij $\mathfrak{p} \subset R$ een priemideaal. Er geldt $x^n = 0 \in \mathfrak{p}$, dus ofwel $x \in \mathfrak{p}$, ofwel $x^{n-1} \in \mathfrak{p}$. Inductief volgt $x \in \mathfrak{p}$.

$\subseteq$: Stel $x \in R \setminus \mathrm{nil}(R)$. Merk op dat $R \setminus \mathrm{nil}(R) \neq \emptyset$ omdat $1 \notin \mathrm{nil}(R)$. Beschouw $S = \{I \subset R$ ideaal: $x^n \notin I$ voor alle $n \in \mathbb{Z}_{>0}\}$. Omdat voor alle $n \in \mathbb{Z}_{>0}$ geldt $x^n \neq 0$, is $\{0\} \in S$, dus $S \neq \emptyset$.

We tonen aan dat S een maximaal element heeft met betrekking tot inclusie: Zij $K \subset S$ een keten. Als $K = \emptyset$ heeft K een bovengrens in S omdat $S \neq \emptyset$. Veronderstel daarom zonder verlies van algemeenheid dat $K \neq \emptyset$.

Zij $I = \{y \in R | \exists J \in K : y \in J\}$. Omdat $K \neq \emptyset$, geldt $0 \in I$, dus $I \neq \emptyset$. Merk op dat I een ideaal is: Voor $r \in R$ en $y \in I$ geldt $y \in J$ voor zeker ideaal $J \in K$, dus $ry \in J$, dus $ry \in I$. In het bijzonder geldt voor alle $y \in I$ dat $-y \in I$. Voor $y_1, y_2 \in I$ geldt $y_1 \in J_1$, $y_2 \in J_2$ voor zekere idealen $J_1, J_2 \in K$. Omdat K volledig geordend is, geldt $J_1 \subseteq J_2$ of $J_2 \subseteq J_1$. In het bijzonder geldt $y_1, y_2 \in J_2$ of $y_1, y_2 \in J_1$, dus $y_1 + y_2 \in J_2$ of $y_1 + y_2 \in J_1$, dus $y_1 + y_2 \in I$.

Omdat voor alle $n \in \mathbb{Z}_{>0}$ en voor alle $J \in K$ geldt $x^n \notin J$, geldt $x^n \notin I$ voor alle $n \in \mathbb{Z}_{>0}$, dus $I \in S$, en voor iedere $J \in K$ geldt $J \subseteq I$. Dus iedere keten van S heeft een bovengrens in S. Met het lemma van Zorn volgt dat S een maximaal element heeft, zeg P.

We tonen aan dat P priem is. Stel $y, z \in R$ zodanig dat $yz \in P$, $y \notin P$ en $z \notin P$. Dan is $P + (y) \supsetneq P$ en $P + (z) \supsetneq P$, dus wegens de maximaliteit van P bevatten $P+(y)$ en $P+(z)$ een macht van x. Maar dan bevat $(P+(y))(P+(z)) \subseteq P + (yz) = P$ een macht van x, wat in tegenspraak is met $P \in S$. Dus P is priem en $x \notin P$. Dus

$$x \notin \bigcap_{\mathfrak{p} \subset R \,:\, \mathfrak{p} \text{ priem}} \mathfrak{p}$$

□

Gevolg 2.4. *Voor iedere ring R is* nil(R) *een ideaal van R.*

Definitie 2.5. *Een ring R heet* Artin *als iedere dalende keten $I_1 \supseteq I_2 \supseteq \ldots$ van idealen $I_1, I_2, \ldots \subset R$ stabiliseert, ofwel als er een $N \in \mathbb{Z}$ bestaat zodanig dat voor iedere $n \geq N$ geldt: $I_n = I_N$.*

Voorbeelden: Lichamen en eindige ringen zijn Artin. Bijvoorbeeld $\mathbb{Z}$ is niet Artin: $(1) \supsetneq (2) \supsetneq (2^2) \supsetneq (2^3) \supsetneq \ldots$

Lemma 2.6. *Zij R een Artin-ring en $I \subset R$ een ideaal. Dan is R/I ook een Artin-ring.*

Bewijs. Het inverse beeld van een ideaal onder een homomorfisme is weer een ideaal. Een dalende keten van idealen in R/I geeft wegens het kanonieke (surjectieve) homomorfisme $\psi : R \to R/I$ een dalende keten in R, omdat $A \subseteq B \subseteq R/I \Rightarrow \psi^{-1}(A) \subseteq \psi^{-1}(B)$ en $A \subsetneq B \subseteq R/I \Rightarrow \psi^{-1}(A) \subsetneq \psi^{-1}(B)$ wegens de surjectiviteit van ψ. Deze keten in R stabiliseert omdat R Artin is, dus stabiliseert de keten in R/I. □

Lemma 2.7. *Ieder priemideaal van een Artin-ring R is maximaal.*

Bewijs. Zij $\mathfrak{p} \subset R$ een priemideaal. Zij $x \in (R/\mathfrak{p}) \setminus \{0\}$ willekeurig gegeven. We beschouwen de dalende keten $(x) \supseteq (x^2) \supseteq (x^3) \supseteq \ldots$. Omdat $R/\mathfrak{p}$ Artin is, stabiliseert deze keten. In het bijzonder is er $n \in \mathbb{Z}_{>0}$ zodat $(x^{n+1}) = (x^n)$, dus er is $y \in R/\mathfrak{p}$ zodat $x^n = y \cdot x^{n+1}$, ofwel $x^n(1 - y \cdot x) = 0$. Omdat $x \neq 0$ en omdat $R/\mathfrak{p}$ een domein is, is $x^n \neq 0$, dus $y \cdot x = 1$, dus x is inverteerbaar. Dus $R/\mathfrak{p}$ is een lichaam, dus $\mathfrak{p}$ is maximaal. □

Lemma 2.8. *Een Artin-ring R heeft slechts eindig veel maximale idealen.*

Bewijs. Laat $S = \{\bigcap_{i=1}^{n} \mathfrak{m}_i | n \in \mathbb{Z}_{>0}, \mathfrak{m}_i \subset R$ een maximaal ideaal voor $i \in \{1, 2, \ldots, n\}\}$ zijn. Als $S = \emptyset$, volgt direct dat R geen maximale idealen heeft, wat er uiteraard slechts eindig veel zijn.

Veronderstel $S \neq \emptyset$. Stel S bevat geen minimaal element. Dan kunnen we een oneindig lange strikt dalende keten van idealen maken, wat in tegenspraak is met het feit dat R Artin is. Dus S bevat een minimaal element, zeg $M = \bigcap_{i=1}^{n} \mathfrak{m}_i$, waarbij $\mathfrak{m}_i = \mathfrak{m}_j$ dan en slechts dan als $i = j$.

Stel $\mathfrak{m} \subset R$ is een maximaal ideaal. Er geldt: $\mathfrak{m} \cap M \in S$, dus $\mathfrak{m} \cap M = M$ omdat M minimaal is. Dus $\mathfrak{m} \supseteq M$. Met lemma 2.1 volgt dat $\mathfrak{m} \supseteq \mathfrak{m}_k$ voor zekere $k \in \{1, 2, \ldots, n\}$. Omdat $\mathfrak{m}_k$ maximaal is en $\mathfrak{m} \neq R$, volgt $\mathfrak{m} = \mathfrak{m}_k$. Dus R heeft slechts eindig veel maximale idealen. □

Stelling 2.9. *Een gereduceerde Artin-ring R is isomorf met een product van eindig veel lichamen.*

Bewijs. Zij R een gereduceerde Artin-ring.

Als $R = \{0\}$, is R isomorf met het lege product. Veronderstel $R \neq \{0\}$.

Veronderstel R heeft maximale idealen $\mathfrak{m}_1, \mathfrak{m}_2, \ldots, \mathfrak{m}_n$. Merk op dat R minstens één maximaal ideaal heeft omdat $R \neq \{0\}$. Uit lemma 2.3 en lemma 2.7 volgt dat $\mathfrak{m}_1 \cap \mathfrak{m}_2 \cap \ldots \cap \mathfrak{m}_n = \text{nil}(R) = \{0\}$. Omdat $\mathfrak{m}_i, \mathfrak{m}_j$ paarsgewijs copriem zijn vanwege hun maximaliteit, geldt met de Chinese reststelling: $R = R/\{0\} \cong \prod_{i=1}^{n} R/\mathfrak{m}_i$.

Omdat $\mathfrak{m}_i$ maximaal is, is $R/\mathfrak{m}_i$ een lichaam voor elke $i \in \{1, 2, \ldots, n\}$. □

Definitie 2.10. *Zij R een ring. De* karakteristiek *van R, genoteerd als* kar*(R), is de niet-negatieve voortbrenger van de kern van het unieke ringhomomorfisme $\mathbb{Z} \to R$.*

Merk op dat voor iedere eindige ring R geldt dat $\text{kar}(R) \neq 0$.

Stelling 2.11. *Zij R een eindige ring met $p = \text{kar}(R)$. Er geldt: R is een lichaam dan en slechts dan als p priem is, p-de machtsverheffing op R een permutatie van R vormt en $|\{x \in R : x^p = x\}| = p$.*

Bewijs. $\Rightarrow$: De karakteristiek van ieder eindig lichaam is priem, dus p is priem. De afbeelding $x \mapsto x^p$ is het bekende Frobenius-automorfisme en dus in het bijzonder een permutatie van R. Omdat R een eindig lichaam is, geldt $R \cong \mathbb{F}_{p^n}$ voor zekere $n \in \mathbb{Z}_{>0}$, en geldt $|\{x \in R : x^p = x\}| = |\mathbb{F}_p| = p$.

$\Leftarrow$: Omdat p-de machtsverheffing een permutatie van R vormt, geldt dat er geen niet-triviaal element is met p-de macht gelijk aan 0. Dit geldt dan en slechts als er geen niet-triviaal element x is waarvoor er $n \in \mathbb{Z}_{>0}$ bestaat met $x^n = 0$: Stel er zijn $x \in R$ en $n \in \mathbb{Z}_{>1}$ zodanig dat $x^n = 0$ en $x^{n-1} \neq 0$. Dan geldt $(x^{n-1})^p = x^{p(n-1)} = 0$ omdat $p(n-1) \geq 2(n-1) = 2n - 2 \geq n$ omdat $n \geq 2$.

Dus $\text{nil}(R) = \{0\}$, dus R is een gereduceerde Artin-ring. Omdat $\text{kar}(R)$ priem is, geldt $R \neq \{0\}$, want $\text{kar}(\{0\}) = 1$. Dus R is isomorf met een eindig, niet-leeg product van lichamen, zeg $R \cong F_1 \times F_2 \times \ldots \times F_m$ voor zekere $m \in \mathbb{Z}_{>0}$ en lichamen $F_1, \ldots, F_m$.

Het volgt direct dat al deze lichamen een karakteristiek hebben die p deelt, ofwel al deze lichamen hebben karakteristiek p. Nu geldt $p = |\{x \in R : x^p = x\}| = |\{(x_1, x_2, \ldots, x_m) \in F_1 \times F_2 \times \ldots \times F_m : (x_1, x_2, \ldots, x_m)^p = (x_1, x_2, \ldots, x_m)\}| = |\{x_1 \in F_1 : x_1^p = x_1\}| \cdot |\{x_2 \in F_2 : x_2^p = x_2\}| \cdot \ldots \cdot |\{x_m \in F_m : x_m^p = x_m\}| = p^m$. Dus $m = 1$ en R is isomorf met een lichaam, dus R is een lichaam.

□

3 Voorkennis over algoritmen

Een zeer informele definitie is dat een algoritme een eindige reeks instructies is waaruit, gegeven bepaalde invoergegevens, een uitvoer wordt gegeven. Een

algoritme dat twee getallen optelt, heeft bijvoorbeeld als invoer de twee getallen die bij elkaar moeten worden opgeteld, en als uitvoer de som van deze getallen.

Onder de lengte van de invoer wordt het aantal bits verstaan waaruit de invoer bestaat. Bijvoorbeeld: een element $x \in \mathbb{Z}/p\mathbb{Z}$ kan gerepresenteerd worden door $\lceil \frac{\log p}{\log 2} \rceil$ bits. Dit is in te zien doordat $\mathbb{Z}/p\mathbb{Z}$ precies p elementen heeft, en met $\lceil \frac{\log p}{\log 2} \rceil$ bits is het mogelijk om $2^{\lceil \frac{\log p}{\log 2} \rceil} \geq p$ verschillende elementen te representeren. Voor grote p is het verschil tussen $\lceil \frac{\log p}{\log 2} \rceil$ en $\frac{\log p}{\log 2}$ relatief klein, dus in het algemeen zullen we zeggen dat x lengte $\frac{\log p}{\log 2}$ heeft.

De bitcomplexiteit van een algoritme is het aantal binaire operaties dat gebruikt wordt om een algoritme uit te voeren (afhankelijk van de invoer). In het algemeen geldt: Hoe lager de bitcomplexiteit, hoe sneller het algoritme. Onder binaire operaties verstaan we het optellen van twee bits, het vermenigvuldigen van twee bits, het vergelijken van twee bits en het complementeren van een bit.

Definitie 3.1. *Laten f, g reëelwaardige functies zijn gedefinieerd op een deelverzameling D van $\mathbb{R}$. We zeggen $f = O(g)$ als er een $C \in \mathbb{R}$ bestaat zodanig dat voor alle $x \in D$ geldt: $|f(x)| \leq C \cdot |g(x)|$.*

Definitie 3.2. *Zij A een algoritme waarvan de invoer I lengte $L(I)$ heeft; laat $B(I)$ de bitcomplexiteit van A zijn. We zeggen dat A* in polynomiale tijd werkt *als er $k \in \mathbb{R}_{\geq 0}$ is zodanig dat $B(I) = O((L(I))^k)$.*

We zeggen dat A in lineaire tijd werkt *als $B(I) = O(L(I))$ en dat A* in bijna lineaire tijd werkt *als voor iedere $\epsilon > 0$ geldt: $B(I) = O((L(I))^{1+\epsilon})$.*

Merk op dat een algoritme A in polynomiale tijd werkt in de lengte van de invoer dan en slechts dan als we een polynoom kunnen vinden in $\mathbb{R}[X]$ waardoor de bitcomplexiteit begrensd wordt in termen van de lengte van de invoer.

Bijvoorbeeld het vergelijken en het optellen van twee gehele getallen n en m werkt in lineaire tijd in termen van $\log n + \log m$, en het vermenigvuldigen van twee gehele getallen n en m werkt in bijna lineaire tijd in termen van $\log n + \log m$ (zie ook [13]).

Het vinden van de inverse van een element van $\mathbb{F}_p$ voor $p \in \mathbb{Z}_{>0}$ priem kan uitgevoerd worden in bitcomplexiteit $O((\log p)^{1+\epsilon})$ (gevolg 11.10 uit [3]).

We hebben nu voldoende kennis om een algoritme op te stellen dat test of een simpel model een model is voor een eindig lichaam, en om te controleren of dit algoritme binnen polynomiale tijd werkt in de lengte van de invoer.

4 Algoritme

Notatie: In $(\mathbb{Z}/p\mathbb{Z})^n$ is $\mathbf{e}_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ de i-de basisvector van de standaard basis voor $i \in \{1, 2, \ldots, n\}$, waarbij $n, p \in \mathbb{Z}_{>0}$.

Voor positieve, gehele getallen p en n en een n bij n bij n systeem A met gehele getallen $a_{i,j,k}$ met $0 \leq a_{i,j,k} < p$ voor $i, j, k \in \{1, 2, \ldots, n\}$ noteren we

$R = (\mathbb{Z}/p\mathbb{Z})^n$. We definiëren een vermenigvuldiging op R door

$$(x_1, x_2, \ldots, x_n) \cdot (y_1, \ldots, y_n) = \sum_{i,j=1}^{n} x_i y_j \mathbf{e}_i \cdot \mathbf{e}_j$$

met $\mathbf{e}_i \cdot \mathbf{e}_j = (a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,n})$ voor $i, j \in \{1, 2, \ldots, n\}$.

Dit is de unieke distributieve vermenigvuldiging op R met vermenigvuldiging op de basiselementen gegeven door $\mathbf{e}_i \cdot \mathbf{e}_j = (a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,n})$ voor $i, j \in \{1, 2, \ldots, n\}$.

Algoritme 1. Invoer: Positieve, gehele getallen p en n, en een n bij n bij n systeem A met gehele getallen $a_{i,j,k}$ met $0 \leq a_{i,j,k} < p$ voor $i, j, k \in \{1, 2, \ldots, n\}$.

Uitvoer: "Ja" of "nee". [Dit is het antwoord op de vraag of R met daarop een vermenigvuldiging gegeven door $\mathbf{e}_i \cdot \mathbf{e}_j = (a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,n})$ voor $i, j \in \{1, 2, \ldots, n\}$ een lichaam is; zie stelling 4.1.]

1 Test of p priem is.

Indien dit waar is, ga door naar 2, zo niet, geef uitvoer "nee".

2 Test of $a_{i,j,k} = a_{j,i,k}$ voor $1 \leq i \leq n$; $1 \leq j < i$, $1 \leq k \leq n$.

Indien dit waar is, ga door naar 3, zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, betekent dit dat de vermenigvuldiging niet commutatief is (zie lemma 4.2).]

3 Test of $\sum_{m=1}^{n} a_{i,j,m} \cdot a_{m,k,l} = \sum_{m=1}^{n} a_{j,k,m} \cdot a_{i,m,l}$ voor $1 \leq i, j, k, l \leq n$.

Indien dit waar is, ga door naar 4, zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, betekent dit dat de vermenigvuldiging niet associatief is (zie lemma 4.3).]

4 Test of er een $\mathbf{1} \in R$ is zodanig dat voor $1 \leq j \leq n$ geldt: $\mathbf{1} \cdot \mathbf{e}_j = \mathbf{e}_j$.

Indien dit waar is, ga door naar 5a, zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, betekent dit dat de vermenigvuldiging op R geen eenheidselement heeft (zie lemma 4.4). Mogelijkerwijs is het nuttig om deze $\mathbf{1}$ op te slaan voor verder gebruik.]

5a Stel de matrix F op voor de lineaire afbeelding van R naar R (voor de standaard basis), gegeven door p-de machtsverheffing. [Omdat p priem is en R commutatief en associatief is, is p-de machtsverheffing van R naar R lineair.]

Sla F op en ga door naar 5b.

5b Bepaal of de rang van F gelijk is aan n. [Als hier als uitvoer "nee" wordt gegeven, betekent dit dat p-de machtsverheffing op R geen permutatie van R vormt (zie lemma 4.7).]

Indien dit waar is, ga door naar 5c, zo niet, geef uitvoer "nee".

5c Bepaal of de rang van $F - \mathrm{Id}$ gelijk is aan $n-1$. [Met Id wordt de n bij n identiteitsmatrix bedoeld.]

Indien dit waar is, geef uitvoer "ja". Zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, betekent dit dat $|\{\mathbf{x} \in R : \mathbf{x}^p = \mathbf{x}\}| \neq p$ (zie lemma 4.8).]

Notatie: Als A het systeem is uit algoritme 1, noteer $A_{i,\cdot,\cdot} = \{a_{i,j,k}\}_{j,k=1}^{n}$, $A_{\cdot,j,\cdot} = \{a_{i,j,k}\}_{i,k=1}^{n}$ en $A_{\cdot,\cdot,k} = \{a_{i,j,k}\}_{i,j=1}^{n}$ voor $i,j,k \in \{1,2,\ldots,n\}$.

We beschouwen iedere stap van algoritme 1 als deelalgoritme met uitvoer "ja" of "nee", waarbij ieder van de stappen uitgevoerd kan worden met invoer I, waarbij $I = (p,n,A)$ of $I = (p,n,A,F)$, in het geval van stappen 5*b* en 5*c*. In dit geval zijn p,n elementen van $\mathbb{Z}_{>0}$, is A een systeem als in algoritme 1 en is F een n bij n matrix met coëfficiënten in $\mathbb{Z}/p\mathbb{Z}$. Bij ieder van de stappen wordt in het volledige algoritme uitvoer "nee" gegeven na het uitvoeren van de betreffende stap als de stap zelf met corresponderende invoer als uitvoer "nee" zou geven, en andersom wordt doorgegaan met de volgende stap, of uitvoer "ja" gegeven in het geval van stap 5*c*, als de stap zelf met corresponderende invoer als uitvoer "ja" zou geven.

Stelling 4.1. ***i*** *Algoritme 1 geeft een correct antwoord op de vraag of de abelse groep $R = (\mathbb{Z}/p\mathbb{Z})^n$ met distributieve vermenigvuldiging gegeven door $\mathbf{e}_i \cdot \mathbf{e}_j = (a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,n})$ voor $i,j \in \{1,2,\ldots,n\}$ een lichaam is.*

ii *Algoritme 1 werkt (mits geschikt geïmplementeerd) in polynomiale tijd in termen van de lengte van de invoer, namelijk, voor elke $\epsilon > 0$, in bitcomplexiteit $O(n^5(\log p)^{1+\epsilon} + n^4(\log p)^{2+\epsilon} + (\log p)^{6+\epsilon})$.*

Bewijs. Zij $\epsilon > 0$.

We tonen aan dat algoritme 1 uitvoer "ja" geeft dan en slechts dan als R een lichaam is.

Stap 1 van het algoritme kan in bitcomplexiteit $O((\log p)^{6+\epsilon})$ uitgevoerd worden; zie [7].

Merk op dat het om te bewijzen dat R een ring is, het voldoende is om aan de tonen dat de vermenigvuldiging commutatief en associatief is en dat er een eenheidselement voor deze vermenigvuldiging is, omdat de vermenigvuldiging automatisch distributief is.

Lemma 4.2. *Stap 2 van algoritme 1 heeft bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$. De vermenigvuldiging is commutatief dan en slechts dan als stap 2 met corresponderende invoer als uitvoer "ja" geeft.*

Bewijs. Iedere vergelijking van twee elementen van A kost hoogstens $O(\log p)$ binaire operaties. Het is voldoende om $\frac{1}{2}(n^3 - n)$ vergelijkingen te maken, dus stap 2 heeft bitcomplexiteit $O(n^3 \log p)$.

Er geldt $\mathbf{e}_i \cdot \mathbf{e}_j = \mathbf{e}_j \cdot \mathbf{e}_i$ voor $1 \leq i \leq n$ en $1 \leq j \leq n$ dan en slechts dan als $(a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,n}) = (a_{j,i,1}, a_{j,i,2}, \ldots, a_{j,i,n})$ voor $1 \leq i \leq n$ en $1 \leq j \leq n$. Dit geldt dan en slechts dan als $a_{i,j,k} = a_{j,i,k}$ voor $1 \leq i,j \leq n$. Aangezien

$a_{i,j,k} = a_{j,i,k}$ als $i = j$ en omdat gelijkheden symmetrisch zijn, is dit waar dan en slechts dan als $a_{i,j,k} = a_{j,i,k}$ voor $1 \leq i \leq n$; $1 \leq j < i$. □

Lemma 4.3. *Stap 3 van algoritme 1 heeft bitcomplexiteit $O(n^5(\log p)^{1+\epsilon})$. De vermenigvuldiging is associatief dan en slechts dan als stap 3 met corresponderende invoer als uitvoer "ja" geeft.*

Bewijs. Voor iedere $1 \leq i, j, k, l \leq n$ is het voldoende om $2n$ vermenigvuldigingen en $2(n-1)$ optellingen uit te voeren. Dit geeft dat de bitcomplexiteit $n^4 \cdot O(n(\log p)^{1+\epsilon}) = O(n^5(\log p)^{1+\epsilon})$ is.

De vermenigvuldiging is associatief dan en slechts dan als $(\mathbf{e}_i \cdot \mathbf{e}_j) \cdot \mathbf{e}_k = \mathbf{e}_i \cdot (\mathbf{e}_j \cdot \mathbf{e}_k)$ voor $1 \leq i, j, k \leq n$. Uitschrijven van deze vermenigvuldigingen geeft dat dit waar is dan en slechts dan als

$$\begin{aligned}&(\sum_{m=1}^{n} a_{i,j,m} \cdot a_{m,k,1}, \sum_{m=1}^{n} a_{i,j,m} \cdot a_{m,k,2}, \ldots, \sum_{m=1}^{n} a_{i,j,m} \cdot a_{m,k,n}) \\ = &(\sum_{m=1}^{n} a_{j,k,m} \cdot a_{i,m,1}, \sum_{m=1}^{n} a_{j,k,m} \cdot a_{i,m,2}, \ldots, \sum_{m=1}^{n} a_{j,k,m} \cdot a_{i,m,n})\end{aligned}$$

voor $1 \leq i, j, k \leq n$. Dit is waar dan en slechts dan als
$\sum_{m=1}^{n} a_{i,j,m} \cdot a_{m,k,l} = \sum_{m=1}^{n} a_{j,k,m} \cdot a_{i,m,l}$ voor $1 \leq i, j, k, l \leq n$. □

Lemma 4.4. *Stap 4 van algoritme 1 heeft bitcomplexiteit $O(n^4(\log p)^{1+\epsilon})$. Als stap 1 en stap 2 met corresponderende invoer als uitvoer "ja" geven, heeft R een eenheidselement voor de vermenigvuldiging dan en slechts dan als stap 4 met corresponderende invoer als uitvoer "ja" geeft.*

Bewijs. Rechtsvermenigvuldiging van een element $\mathbf{x} \in R$ met $\mathbf{e}_j$ wordt gegeven door $\mathbf{x} \cdot \mathbf{e}_j = A^T_{\cdot,j,\cdot} \cdot \mathbf{x}$. Het stelsel gegeven door $A^T_{\cdot,j,\cdot} \cdot \mathbf{x} = \mathbf{e}_j$ met $1 \leq j \leq n$ kan opgelost worden met Gauss-eliminatie (zie [11]); dit heeft bitcomplexiteit $O(n^4(\log p)^{1+\epsilon})$. Omdat p priem is, is het altijd mogelijk de inverse van een element van $\mathbb{Z}/p\mathbb{Z}$ te vinden (wat nodig kan zijn om de oplossing van een stelsel vergelijkingen te vinden met Gauss-eliminatie).

Wegens de distributiviteit en commutativiteit van de vermenigvuldiging, is er $\mathbf{1} \in R$ zodanig dat voor $1 \leq j \leq n$ geldt $\mathbf{1} \cdot \mathbf{e}_j = \mathbf{e}_j$ dan en slechts dan als er $\mathbf{1} \in R$ is zodanig dat voor iedere $\mathbf{x} \in R$ geldt $\mathbf{1} \cdot \mathbf{x} = \mathbf{x} = \mathbf{x} \cdot \mathbf{1}$. □

Gevolg 4.5. *Stap 2, 3 en 4 van algoritme 1 geven met corresponderende invoer als uitvoer "ja" dan en slechts dan als R een ring is.*

Lemma 4.6. *Stap 5a van algoritme 1 heeft bitcomplexiteit $O(n^4(\log p)^{2+\epsilon})$.*

Bewijs. Om p-de machtsverheffing uit te voeren, is het voldoende om $O(\log p)$ vermenigvuldigingen uit te voeren (door herhaaldelijk te kwadrateren). Het is voldoende om dit voor alle n basisvectoren uit te voeren. Aangezien vermenigvuldiging uitgevoerd kan worden in bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$, geeft dit totale bitcomplexiteit $O(n^4(\log p)^{2+\epsilon})$. □

Lemma 4.7. *Stap 5b van algoritme 1 heeft bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$. Gegeven dat R een ring is en p priem is, vormt p-de machtsverheffing op R een permutatie van R dan en slechts dan als stap 5b met corresponderende invoer als uitvoer "ja" geeft.*

Bewijs. De rang van een matrix kan bepaald worden door Gauss-eliminatie uit te voeren en vervolgens te kijken hoeveel niet-nul rijen de matrix heeft. Het laatste heeft bitcomplexiteit $O(n^2 \log p)$, terwijl het eerste bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$ heeft. De totale bitcomplexiteit is dus $O(n^3(\log p)^{1+\epsilon})$.

De rang van F is gelijk aan n dan en slechts dan als F inverteerbaar is. Aangezien F de matrix is die de p-de machtsverheffing op R representeert, is F inverteerbaar dan en slechts dan als p-de machtsverheffing op R een permutatie van R vormt. □

Lemma 4.8. *Stap 5c van algoritme 1 heeft bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$. Gegeven dat R een ring is en p priem is, geldt dat $|\{\mathbf{x} \in R : \mathbf{x}^p = \mathbf{x}\}| = p$ dan en slechts dan als stap 5c met corresponderende invoer als uitvoer "ja" geeft.*

Bewijs. De bitcomplexiteit is dezelfde als bij 5b, ofwel $O(n^3 \log(p)^{1+\epsilon})$.

Omdat p priem is, is R een vectorruimte over $\mathbb{F}_p$. Er geldt dat $F - \mathrm{Id}$ rang $n-1$ heeft dan en slechts dan als $\dim(\mathrm{Ker}(F-\mathrm{Id})) = 1$, wat waar is dan en slechts dan als $\mathrm{Ker}(F - \mathrm{Id}) = \mathrm{span}_{\mathbb{F}_p}(\mathbf{v})$ voor zekere $\mathbf{v} \in R$. Dit is waar dan en slechts dan als $|\mathrm{Ker}(F - \mathrm{Id})| = p$. Omdat $F - \mathrm{Id}$ de lineaire afbeelding representeert gedefinieerd door $\mathbf{x} \mapsto \mathbf{x}^p - \mathbf{x}$, geldt $\mathrm{Ker}(F - \mathrm{Id}) = \{\mathbf{x} \in R : \mathbf{x}^p - \mathbf{x}\}$, dus $F - \mathrm{Id}$ heeft rang $n - 1$ dan en slechts dan als $|\{\mathbf{x} \in R : \mathbf{x}^p - \mathbf{x}\}| = p$. □

Uit de lemma's kunnen we concluderen dat het algoritme "ja" als uitvoer geeft dan en slechts dan als R een eindige ring is waarvoor $\mathrm{kar}(R) = p$ priem is, p-de machtsverheffing op R een permutatie van R vormt en $|\{\mathbf{x} \in R : \mathbf{x}^p = \mathbf{x}\}| = p$. Met stelling 2.11 volgt dat dit geldt dan en slechts dan als R een lichaam is.

□

5 Optimalisatie

De delen van het algoritme die de rekentijdschatting bepalen zijn de stappen 1, 3 en 5*a*. Hiervan is stap 1 wellicht te verbeteren door een betere priemtest.

Stap 5*a* lijkt lastig te verbeteren; voor p-de machtsverheffing lijken inherent $\frac{\log p}{\log 2}$ vermenigvuldigingen nodig te zijn. We zullen later zien dat we, gegeven een bepaalde basistransformatiematrix en diens inverse, vermenigvuldiging in R in $O(n^2(\log p)^{1+\epsilon})$ binaire operaties kunnen uitvoeren (onder bepaalde omstandigheden). Het opstellen van deze matrix vereist wellicht echter te grote bitcomplexiteit.

Er is wel ruimte voor verbetering bij stap 3. We kunnen deze stap herformuleren in termen van matrixvermenigvuldiging: Test of $(A_{\cdot,j,\cdot} \cdot A_{\cdot,k,\cdot})_{i,l} = (A_{j,\cdot,\cdot} \cdot A_{i,\cdot,\cdot})_{k,l}$ voor $i, j, k, l \in \{1, 2, \ldots, n\}$.

Het is mogelijk om matrixvermenigvuldiging van twee n bij n matrices uit te voeren in $O(n^{2,376})$ lichaamsoperaties van $\mathbb{F}_p$ in plaats van in $O(n^3)$ lichaamsoperaties; zie ook [12]. Dit betekent dat we stap 3 kunnen verkorten tot bitcomplexiteit $O(n^{4,376}(\log p)^{1+\epsilon})$.

In totaal kan algoritme 1 dus met geschikte implementatie voor elke $\epsilon > 0$ werken in bitcomplexiteit $O(n^{4,376}(\log p)^{1+\epsilon} + n^4 \log(p)^{2+\epsilon} + (\log p)^{6+\epsilon})$.

Verder is het mogelijk om stap 4 te verkorten tot bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$ door hem te vervangen door de volgende stappen:

4a' Test of er een unieke $\mathbf{1} \in A$ is zodanig dat $\mathbf{1} \cdot \mathbf{e}_1 = \mathbf{e}_1$.

Indien dit waar is, sla $\mathbf{1}$ op en ga door naar 4b, zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, betekent dit ofwel dat er geen eenheidselement is voor de vermenigvuldiging, ofwel dat $\mathbf{e}_1$ een nuldeler is; zie de tekst hierna.]

4b' Test of $\mathbf{1} \cdot \mathbf{e}_i = \mathbf{e}_i$ voor $2 \leq i \leq n$ (existentie van een eenheidselement).

Indien dit waar is, ga door naar 5a, zo niet, geef uitvoer "nee". [Als hier als uitvoer "nee" wordt gegeven, is $\mathbf{1}$ geen eenheidselement voor de vermenigvuldiging; zie de tekst hierna.]

Als het stelsel (p, n, A) een model moet geven voor een eindig lichaam, is het noodzakelijk dat er precies één $\mathbf{1} \in A$ is zodanig dat $\mathbf{1} \cdot \mathbf{e}_1 = \mathbf{e}_1$; als dit niet zo is, is $\mathbf{e}_1$ een nuldeler, en in het bijzonder geen eenheid.

Het stelsel $A^T_{\cdot,1,\cdot} \cdot \mathbf{x} = \mathbf{e}_1$ kan opgelost worden met Gauss-eliminatie, gegeven dat stap 1 en stap 2 met corresponderende invoer als uitvoer "ja" zouden geven; dit heeft bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$.

Indien er een unieke oplossing $\mathbf{1} \in R$ bestaat voor dit stelsel, voldoet het nu om te controleren of $\mathbf{1} \cdot \mathbf{e}_j = \mathbf{e}_j$ voor $2 \leq j \leq n$ als we willen weten of $\mathbf{1}$ een eenheidselement is voor de vermenigvuldiging. Te controleren: $A^T_{\cdot,j,\cdot} \cdot \mathbf{1} = \mathbf{e}_j$ voor $2 \leq j \leq n$; dit kan gedaan worden in $O(n^3(\log p)^{1+\epsilon})$ bewerkingen. Als niet doorgegaan wordt na de aangepaste stap 4, betekent dit ofwel dat R geen eenheidselement heeft voor de vermenigvuldiging, of dat $\mathbf{e}_1$ niet inverteerbaar is. In beide gevallen is het correct om af te breken. Als wordt doorgegaan, verloopt het algoritme verder zoals gewenst; alleen lemma 4.4 en gevolg 4.5 zijn niet meer correct; R zou een ring kunnen zijn waarin $\mathbf{e}_1$ niet inverteerbaar is.

We kunnen tevens een (triviale) ondergrens geven voor de bitcomplexiteit van een algoritme dat test of een stelsel als in algoritme 1 een lichaam geeft, namelijk $O(n^3 \log p)$. Dit is de benodigde tijd om de invoer te lezen: Er zijn n^3 elementen die lengte $\frac{\log p}{\log 2}$ hebben, er is één element dat lengte $\frac{\log p}{\log 2} + 1$ heeft en er is één element dat lengte $\frac{\log n}{\log 2} + 1$ heeft; aangezien $\frac{\log n}{\log 2} + 1 < n^3$ voor $n \in \mathbb{Z}_{>0}$ is de totale benodigde tijd om alles af te lezen $O(n^3 \log p)$.

Dit is feitelijk vrij langzaam. Daarom zal in de rest van deze scriptie een ander model behandeld worden, en de mogelijkheid om ons oorspronkelijke model in dit snellere model om te schrijven.

6 Overige algebra

Zij $p \in \mathbb{Z}_{>0}$ priem. Uit de algebra (zie paragraaf 22 uit [6]) is bekend dat in $\mathbb{F}_p[X]$ geldt dat

$$X^{p^n} - X = \prod_{\substack{f \in \mathbb{F}_p[X]: \\ f \text{ monisch irreducibel} \\ \deg(f) | n}} f$$

voor alle $n \in \mathbb{Z}_{>0}$.

Lemma 6.1. *Zij $p \in \mathbb{Z}_{>0}$ priem, zij $f \in \mathbb{F}_p[X]$ met $\deg(f) > 0$. Dan is f irreducibel in $\mathbb{F}_p[X]$ dan en slechts dan als $\gcd(f, X^{p^i} - X) = 1$ voor $i \in \{1, 2, \ldots, \lfloor \frac{1}{2} \deg(f) \rfloor\}$.*

Bewijs. $\Rightarrow$: Als f irreducibel is, heeft f geen irreducibele deler van graad ten hoogste $\frac{1}{2}\deg(f)$, dus heeft f geen deler gemeenschappelijk met $X^{p^i} - X$ voor $i \in \{1, 2, \ldots, \lfloor \frac{1}{2} \deg(f) \rfloor\}$.

$\Leftarrow$: Als f niet irreducibel is, heeft f een irreducibele deler g van graad ten hoogste $\frac{1}{2}\deg(f)$. Er geldt dat g tevens $X^{p^{\deg(g)}} - X$ deelt, dus $g | \gcd(f, X^{p^{\deg(g)}} - X)$, dus $\gcd(f, X^{p^{\deg(g)}} - X) \neq 1$. □

Lemma 6.2. *Zij K een lichaam met algebraïsche afsluiting $\overline{K}$. Laat L, M lichamen zijn met $K \subseteq L \subseteq \overline{K}$, $K \subseteq M \subseteq \overline{K}$ zodanig dat voor ieder tussenlichaam $K \subseteq L' \subseteq L$ en voor ieder tussenlichaam $K \subseteq M' \subseteq M$ waarvoor $[L' : K]$ en $[M' : K]$ eindig zijn geldt dat $\gcd([L' : K], [M' : K]) = 1$. Dan geldt voor ieder tweetal K-bases B_L van L en B_M van M dat $B_L \cdot B_M = \{ef : e \in B_L, f \in B_M\}$ een basis vormt voor LM, het compositum van L en M binnen $\overline{K}$. Verder geldt dat als voor $e, e' \in B_L$, $f, f' \in B_M$ geldt $ef = e'f'$, dan $e = e'$, $f = f'$.*

Bewijs. Laat B_L en B_M bases zijn van $K \subseteq L$ respectievelijk $K \subseteq M$. Beschouw $K[B_L \cdot B_M]$. Merk op dat ieder element hierin te schrijven is als een polynomiale uitdrukking in producten van elementen van $B_L \cdot B_M$. Omdat een macht van een element van $B_L \cdot B_M$ te schrijven is als een product van een element $l \in L$ en een element $m \in M$, en omdat l te schrijven is als K-lineaire combinatie van elementen van B_L en m te schrijven is als K-lineaire combinatie van elementen van B_M, is ieder element uit $K[B_L \cdot B_M]$ te schrijven als K-lineaire combinatie van elementen van $B_L \cdot B_M$.

Stel $x \in K[B_L \cdot B_M]$ met $x \neq 0$. Dan geldt dat x algebraïsch over K is en f_K^x heeft constante term ongelijk aan 0. Ofwel, $\sum_{i=1}^n k_i x^i = -k_0$ voor zekere $n \in \mathbb{Z}_{>0}$, $k_0, \ldots, k_n \in K$, $k_0 \neq 0$. Dan $x \cdot -k_0^{-1} \sum_{i=1}^n k_i x^{i-1} = 1$, dus $K[B_L \cdot B_M]$ is een lichaam. In het bijzonder bevat het L en M, dus $LM \subseteq K[B_L \cdot B_M]$, dus ieder element van LM is te schrijven als K-lineaire combinatie van elementen van $B_L \cdot B_M$.

Stel $\sum_{i=1}^n \sum_{j=1}^m k_{i,j} e_i f_j = 0$ voor zekere $n, m \in \mathbb{Z}_{>0}$, $k_{i,j} \in K$, $e_i \in B_L$, $f_j \in B_M$ voor $1 \leq i \leq n$, $1 \leq j \leq m$. Beschouw $L' = K(\{e_1, \ldots, e_n\})$ en $M' = K(\{f_1, \ldots, f_m\})$. Omdat $e_1, \ldots, e_n$ lineair onafhankelijk zijn over K,

kunnen we $\{e_1, \ldots, e_n\}$ aanvullen tot een basis voor L', die eindig is omdat $e_1, \ldots, e_n$ algebraïsch over K zijn. Evenzo kunnen we $\{f_1, \ldots, f_m\}$ aanvullen tot een basis voor M'.

Omdat $\text{ggd}([L' : K], [M' : K]) = 1$ en $L' \subseteq L'M'$, $M' \subseteq L'M'$, geldt $[L' : K] \cdot [M' : K] | [L'M' : K]$. Verder zien we als hiervoor dat $L'M'$ voortgebracht wordt door het product van deze bases van L' en M'. Aangezien dit product hoogstens $[L' : K] \cdot [M' : K]$ elementen bevat, moet gelden $[L'M' : K] \leq [L' : K] \cdot [M' : K]$, dus volgt $[L' : K] \cdot [M' : K] = [L'M' : K]$. Nu geldt dat het product van deze bases een basis is voor $L'M'$. Hieruit volgt dat dit product precies $[L' : K] \cdot [M' : K]$ elementen moet bevatten, dus $e_i f_j = e_{i'} f_{j'} \Rightarrow e_i = e_{i'}$ en $f_j = f_{j'}$ voor $1 \leq i, i' \leq n$ en $1 \leq j, j' \leq m$.

In het bijzonder is $\{e_i f_j\}_{i=1,\ldots,n, j=1,\ldots,m}$ lineair onafhankelijk over K, dus $k_{i,j} = 0$ voor $1 \leq i \leq n$ en $1 \leq j \leq m$. □

Merk op dat we dit lemma kunnen samenvatten in termen van tensorproducten door $L \otimes_K M \cong L \cdot M$ als K-algebras.

Zij $q \in \mathbb{Z}_{>0}$ een priemmacht.

Definitie 6.3. *Voor $x \in \overline{\mathbb{F}}_q$ is de* q-graad *van x de graad van het minimumpolynoom van x over $\mathbb{F}_q$.*

Stelling 6.4. *Zij $q \in \mathbb{Z}_{>0}$ een priemmacht. Er is een $\mathbb{F}_q$-basis B voor $\overline{\mathbb{F}}_q$ zodanig dat B een $\mathbb{F}_q$-basis bevat voor $\mathbb{F}_{q^n}$ voor iedere $n \in \mathbb{Z}_{>0}$.*

Bewijs. Voor $r, m \in \mathbb{Z}_{>0}$ met r priem, laat $a_{r,m} \in \mathbb{F}_{q^{r^m}} \setminus \mathbb{F}_{q^{r^{m-1}}}$. Dan heeft $a_{r,m}$ als $q^{r^{m-1}}$-graad r en dus zijn $1, a_{r,m}, a_{r,m}^2, \ldots, a_{r,m}^{r-1}$ lineair onafhankelijk over $\mathbb{F}_{q^{r^{m-1}}}$ en vormen in het bijzonder een basis voor $\mathbb{F}_{q^{r^{m-1}}} \subset \mathbb{F}_{q^{r^m}}$.

Uit stelling 21.3 uit [6] volgt inductief dat voor $r \in \mathbb{Z}_{>0}$ priem, de collectie van elementen $\prod_{m=1}^{\infty} a_{r,m}^{c_{r,m}}$, met $0 \leq c_{r,m} < r$ en $c_{r,m} \neq 0$ voor slechts eindig veel $m \in \mathbb{Z}_{>0}$, een basis vormt voor $\mathbb{F}_{q^{r^\infty}} = \bigcup_{m=1}^{\infty} \mathbb{F}_{q^{r^m}}$ over $\mathbb{F}_q$.

Door lemma 6.2 herhaald toe te passen, volgt dat de collectie van elementen $\prod_{r \text{ priem}} \prod_{m=1}^{\infty} a_{r,m}^{c_{r,m}}$, met $0 \leq c_{r,m} < r$ en $c_{r,m} \neq 0$ voor slechts eindig veel $m \in \mathbb{Z}_{>0}$ en $r \in \mathbb{Z}_{>0}$ priem, een basis vormt voor $\overline{\mathbb{F}}_q$ over $\mathbb{F}_q$. We noteren deze collectie als B.

Merk op dat dit in het bijzonder betekent dat voor iedere $n \in \mathbb{Z}_{>0}$, we n verschillende lineair onafhankelijke elementen van $\mathbb{F}_{q^n}$ in B hebben, namelijk de elementen van de vorm $\prod_{r|n, r \text{ priem}} \prod_{m=1}^{\text{ord}_r(n)} a_{r,m}^{c_{r,m}}$ waarbij $0 \leq c_{r,m} < r$ voor iedere r, m. Oftewel, B bevat een $\mathbb{F}_q$-basis voor $\mathbb{F}_{q^n}$ voor iedere $n \in \mathbb{Z}_{>0}$.

□

Lemma 6.5. **1** *Voor iedere $q \in \mathbb{Z}_{>0}$ geldt $\dim_{\mathbb{F}_q}(\mathbb{F}_{q^n} / \sum_{d|n,\, d \neq n} \mathbb{F}_{q^d}) = \phi(n)$, waarbij $\phi : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ de Euler phi functie is.*

2 *Iedere $\mathbb{F}_q$-basis voor $\mathbb{F}_{q^n}$ bevat minstens $\phi(n)$ elementen van q-graad n.*

Bewijs. **1** Zij $B \subset \overline{\mathbb{F}}_q$ een basis als in stelling 6.4. Zij $B_n \subset B$ een $\mathbb{F}_q$-basis van $\mathbb{F}_{q^n}$ voor $n \in \mathbb{Z}_{>0}$. Omdat $\mathbb{F}_{q^d} \subset \mathbb{F}_{q^n}$ voor iedere $d|n$, moet gelden $B_d \subset B_n$ voor iedere $d|n$.

De verzameling $\bigcup_{d|n, d<n} B_d$ spant $\sum_{d|n,\ d<n} \mathbb{F}_{q^d}$ op, dus $B_n \setminus \bigcup_{d|n,\ d<n} B_d$ vormt een basis voor $\mathbb{F}_{q^n} / \sum_{d|n,\ d<n} \mathbb{F}_{q^d}$.

Met inductie naar n volgt dat het aantal elementen van q-graad n in B_n gelijk is aan $n - \sum_{d|n,\ d<n} \phi(d) = \phi(n)$, dus $|B_n \setminus \bigcup_{d|n,\ d<n} B_d| = \phi(n)$, dus $\dim_{\mathbb{F}_q}(\mathbb{F}_{q^n} / \sum_{d|n,\ d\neq n} \mathbb{F}_{q^d}) = \phi(n)$.

2 Zij C een $\mathbb{F}_q$-basis van $\mathbb{F}_{q^n}$. Er geldt dat $C \setminus \{x \in C : x \text{ heeft } q\text{-graad kleiner dan } n\}$ de vectorruimte $\mathbb{F}_{q^n} / \sum_{d|n, d\neq n} \mathbb{F}_{q^d}$ opspant, dus $|\{x \in C : x \text{ heeft } q\text{-graad } n\}| \geq \dim_{\mathbb{F}_q}(\mathbb{F}_{q^n} / \sum_{d|n,\ d\neq n} \mathbb{F}_{q^d}) = \phi(n)$. Dus C bevat tenminste $\phi(n)$ elementen van q-graad n. □

7 Efficiënter model

Zoals gezegd is het grootste probleem met het oorspronkelijke model dat er zoveel gegevens nodig zijn. Een efficiëntere manier om een model te geven voor een eindig lichaam is het volgende: Beschouw $\mathbb{F}_p[X]/(f)$, waarbij f een monisch polynoom is en $p \in \mathbb{Z}_{>0}$. Laat $n = \deg(f)$. Schrijf $f = a_0 + a_1X + a_2X^2 + \ldots + a_{n-1}X^{n-1} + X^n$.

Nu dient, om te controleren of $\mathbb{F}_p[X]/(f)$ een lichaam is, slechts gecontroleerd te worden of p priem is en of f irreducibel is. Als we $\{1, X, X^2, \ldots, X^{n-1}\}$ als basis nemen voor $\mathbb{F}_p[X]/(f)$, zijn de enige benodigde gegevens $a_0, a_1, \ldots, a_{n-1}$, ofwel n elementen van $\mathbb{F}_p$. De totale informatie die benodigd is, is nu $O(n \log p)$.

Vermenigvuldiging op de basiselementen wordt gegeven door $X^i \cdot X^j = X^{i+j}$, waarbij X^k wordt bepaald door deling met rest door f indien $k \geq n$. Totale vermenigvuldiging van twee elementen wordt gegeven door de polynoomvermenigvuldiging van de twee, gevolgd door deling met rest door f. Aangezien het product van twee polynomen van graad hoogstens $n-1$ als graad maximaal $2n-2$ heeft, kan deling met rest volgens paragraaf 17 van [2] in bijna lineaire tijd uitgevoerd worden. De lengte van de invoer is in dit geval $(3n-1)\frac{\log p}{\log 2}$, dus deling met rest kan voor iedere $\epsilon > 0$ in $O((n \log p)^{1+\epsilon})$ binaire operaties uitgevoerd worden. Het vermenigvuldigen van twee polynomen van graad minder dan n kan ook in bitcomplexiteit $O((n \log p)^{1+\epsilon})$ uitgevoerd worden, dus vermenigvuldiging in $\mathbb{F}_p[X]/(f)$ kan in bitcomplexiteit $O((n \log p)^{1+\epsilon})$ uitgevoerd worden voor iedere $\epsilon > 0$.

Een simpel algoritme om te testen of $f \in \mathbb{F}_p[X]$ irreducibel is, waarbij p priem is en $n = \deg(f)$, gaat nu als volgt:

Voor $1 \leq i \leq \lfloor \frac{1}{2}n \rfloor$, schrijf $\overline{X}^{p^i} - \overline{X} \in \mathbb{F}_p[X]/(f)$ op de basis $1, X, \ldots, X^{n-1}$. Bepaal de ggd van f en het resterende polynoom (dat graad kleiner dan n heeft). Als dit in alle gevallen $\overline{1}$ is, geef uitvoer "ja"; zo niet geef uitvoer "nee".

Het bepalen van $\overline{X}^{p^i} - \overline{X}$ wordt gedaan door $\overline{X}$ herhaald te kwadrateren, wat in $O(i \log p)$ bewerkingen gedaan kan worden. Elk van deze bewerkingen kan gedaan worden in bijna lineaire tijd, dus de uiteindelijke bitcomplexiteit is

$O((n \log p)^{2+\epsilon})$, voor iedere $\epsilon > 0$. Merk op dat deze stap van het algoritme feitelijk de rest berekent van de deling van $X^{p^i} - X$ door f. Deze stap kan met een truc ingekort worden tot bitcomplexiteit $O((n + \log p) \cdot (n \log p)^{1+\epsilon})$ voor alle $\epsilon > 0$, door op te merken dat als je X^{p^i} wilt berekenen voor $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$, je dit in feite kunt doen door bepaalde polynomen mod f te evalueren in ten hoogste $\frac{1}{2}n$ punten, waarbij dit aantal polynomen ten hoogste $\log n$ is. Hierbij kun je gevolg 10.8 uit [3] gebruiken.

De grootste gemeenschappelijke deler van twee polynomen in $\mathbb{F}_p[X]$ bepalen kan in bitcomplexiteit $O((n \log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$ volgens paragraaf 22 in [2]. Dit uitvoeren voor $\lfloor \frac{n}{2} \rfloor$ polynomen geeft bitcomplexiteit $O(n(n \log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$. Als eerst getest moet worden of p priem is, is de totale bitcomplexiteit $O((n + \log p)(n \log p)^{1+\epsilon} + (\log p)^{6+\epsilon})$ voor iedere $\epsilon > 0$, wat een verbetering is wat betreft de macht van n die nodig is.

Gevolg 7.1. *Er is een algoritme dat in polynomiale tijd test of $\mathbb{F}_p[X]/(f)$ een lichaam vormt voor gegeven $p \in \mathbb{Z}_{>0}$ en monische $f \in \mathbb{F}_p[X]$. Met geschikte implementatie kan de bitcomplexiteit van dit algoritme afgeschat worden met $O((n + \log p)(n \log p)^{1+\epsilon} + (\log p)^{6+\epsilon})$.*

Het algoritme in kwestie is het volgende algoritme:

Algoritme 2. Invoer: Positieve, gehele getallen p en n en een systeem A met gehele getallen a_i met $0 \leq a_i < p$ voor $i \in \{0, 1, \ldots, n-1\}$. [Noteer $f(X) = a_0 + a_1 X + \ldots + a_{n-1} X^{n-1} + X^n$.]

Uitvoer: "Ja" of "nee". [Dit is het antwoord op de vraag of $\mathbb{F}_p[X]/(f)$ een lichaam is.]

1 Test of p priem is.

Indien dit waar is, ga door naar 2, zo niet, geef uitvoer "nee".

2a Bereken $f_i = X^{p^i} - X \bmod f$ voor $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$.

Sla $f_1, f_2, \ldots, f_{\lfloor \frac{n}{2} \rfloor}$ op en ga door naar 2b.

2b Test of $\gcd(f_i, f) = 1$ voor $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$.

Indien dit waar is, geef uitvoer "ja". Zo niet, geef uitvoer "nee". [De uitvoer is hier "ja" dan en slechts dan als f irreducibel is, zoals uit de voorgaande tekst en uit lemma 6.1 op te maken is.]

Stel er is nu getest of een model (p, n, A) als in paragraaf 4 een lichaam is. Hoe gemakkelijk is het nu om dit model om te schrijven naar een model van de vorm $(p, a_0, a_1, \ldots, a_{n-1})$ als in deze paragraaf? Met lemma 6.5 weten we dat tenminste 1 van de basiselementen $\mathbf{e}_1, \ldots, \mathbf{e}_n$ als p-graad n heeft. Het minimumpolynoom van $\mathbf{e}_i$ opstellen is een kwestie van lineaire algebra. We stellen de machten op van $\mathbf{e}_i$ van 0 tot en met $n-1$, en zoeken naar een lineaire afhankelijkheid. Het vinden van deze machten kan in $n-2$ vermenigvuldigingen gedaan worden, als we ervan uitgaan dat het eenheidselement gegeven is (wat gemakkelijk aan

te passen is in stap 4 van algoritme 1). Met Gauss-eliminatie vinden we het minimumpolynoom van $\mathbf{e}_i$. Totaal heeft dit bitcomplexiteit $O(n^4(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$. Als we dit doen voor $1 \leq i \leq n$ (waarbij we stoppen zodra we een basiselement vinden van p-graad n), hebben we bitcomplexiteit $O(n^5(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$.

Veronderstel dat $\mathbf{e}_i$ een minimumpolynoom heeft van graad n over $\mathbb{F}_p$. We willen nu een element snel kunnen omschrijven op de basis $\{1, \mathbf{e}_i, \mathbf{e}_i^2, \ldots, \mathbf{e}_i^{n-1}\}$. Hiervoor dienen we een basistransformatiematrix op te stellen. Uitschrijven van $\mathbf{e}_j$ op de nieuwe basis kan gedaan worden door eerst de machten van $\mathbf{e}_i$ te bekijken op de oude basis, en vervolgens $\mathbf{e}_j$ als lineaire combinatie van deze machten te vinden, wat met Gauss-eliminatie in bitcomplexiteit $O(n^3(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$ gedaan kan worden. Dit moet $n-1$ maal gebeuren, dus de transformatiematrix kan opgesteld worden in $O(n^4(\log p)^{1+\epsilon})$ binaire operaties. Het omschrijven van de representatie van een element $\mathbf{x}$ op de oude basis naar een representatie op de nieuwe basis, is nu een kwestie van het vermenigvuldigen van $\mathbf{x}$ met deze matrix, wat in bitcomplexiteit $O(n^2(\log p)^{1+\epsilon})$ gedaan kan worden voor iedere $\epsilon > 0$. Vermenigvuldigen van twee elementen op de nieuwe basis lukt in bitcomplexiteit $O((n \log p)^{1+\epsilon})$, en terugschrijven naar de oude basis kan in bitcomplexiteit $O(n^2(\log p)^{1+\epsilon})$ (als de inverse van de basistransformatiematrix eerst gevonden is), dus de vermenigvuldiging van twee elementen kan in $O(n^2(\log p)^{1+\epsilon})$ binaire operaties uitgevoerd worden voor iedere $\epsilon > 0$, wat uiteraard sneller is dan de oorspronkelijke benodigde bitcomplexiteit van $O(n^3(\log p)^{1+\epsilon})$.

Gevolg 7.2. *Het is mogelijk om, met enig voorwerk, vermenigvuldiging van twee elementen in een model (p, n, A) als in paragraaf 4 uit te voeren in bitcomplexiteit $O(n^2(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$. Het voorwerk kan gedaan worden in $O(n^5(\log p)^{1+\epsilon})$ voor iedere $\epsilon > 0$.*

Referenties

[1] M.F. Atiyah en I.G. Macdonald, *Introduction to Commutative Algebra*, 1969, Addison-Wesley Publishing Company

[2] D.J. Bernstein, "Fast multiplication and its applications", in: *Surveys in algorithmic number theory*. Link: http://www.math.leidenuniv.nl/∼psh/ANTproc/10djb.pdf

[3] J. Von zur Gathen en J. Gerhard, *Modern Computer Algebra*, 2003, Cambridge University Press

[4] S. Lang, *Algebra*, 2002, Springer

[5] P. Stevenhagen, *Algebra II*, dictaat tweedejaarsvak wiskunde Universiteit Leiden, 2007

[6] P. Stevenhagen, *Algebra III*, dictaat tweedejaarsvak wiskunde Universiteit Leiden, 2008

[7] http://mathworld.wolfram.com/AKSPrimalityTest.html

[8] http://nl.wikipedia.org/wiki/Algoritme

[9] http://en.wikipedia.org/wiki/Big-O_notation

[10] http://en.wikipedia.org/wiki/Bit_complexity

[11] http://en.wikipedia.org/wiki/Gaussian_elimination

[12] http://en.wikipedia.org/wiki/Matrix_multiplication

[13] http://en.wikipedia.org/wiki/Multiplication_algorithm