

Inhoudsopgave

1	Inleiding	3
2	Algebraïsche structuren	4
2.1	voorbeelden	4
2.2	ringen en lichamen: definities	5
2.3	ringen en lichamen: constructies	9
2.4	(eenheden)groepen	15
2.5	machtsverheffen	21
2.6	het euclidische algoritme	23
2.7	de chinese reststelling	31
2.8	Opgaven	35
2.9	een diagnostische toets	37
3	Cryptografie	38
3.1	eenheden modulo N	38
3.2	worteltrekken modulo N	40
3.3	RSA	41
3.4	Opgaven	45
4	Irreducibiliteit	46
4.1	polynomen over een eindig lichaam	46
4.2	ontbindbaarheid van getallen	49
4.3	de Lucas-Lehmer priemtest	54
4.4	Het AKS-algoritme	56
4.5	Opgaven	59
5	Factoriseren	60
5.1	veeltermen over een eindig lichaam	61
5.2	getallen	65
5.3	Opgaven	70
6	Codering	71
6.1	binaire lineaire codes	72
6.2	drie voorbeelden	76
6.3	Opgaven	85
7	Eindopdrachten	87
7.1	de exponent van $(\mathbb{Z}/N\mathbb{Z})^*$	87
7.2	Carmichaelgetallen	87
7.3	AKS-priemtesten	88

7.4	Carmichaelpolynomen	88
7.5	worteltrekken modulo p^n	89
7.6	button madness	89
7.7	factoriseren met $x \bmod (x^2 + ax + 1)$	90
7.8	opnieuw: de kwadratische zeef	91
7.9	de factorisatie van $x^p - 1$	91
7.10	nulpunten vinden in \mathbb{F}_p	91
7.11	repunits	92
7.12	het voetbalpool probleem	92
7.13	de binaire Golay code	92
7.14	Rijndael	93

1 Inleiding

Een pakket voor symbolisch rekenen zoals Magma, Maple of Mathematica kan je op veel middelbare school wiskundeopgaven een antwoord geven:

- hoe ziet de grafiek van $\log(1 + \sin x)/x$ eruit?
- wat is een primitieve van $x \tan(x^2)$?
- waar liggen de nulpunten van $f(x) = \cos(x) - 2x^2$?
- wat is $\lim_{x \rightarrow 0} x^{-4}(2 \cos(x) + e^x + e^{-x})$?

Naast dit soort standaardvragen blijken de pakketten zelfs veel meer te kunnen, zoals allerlei differentiaalvergelijkingen oplossen (welke functies y voldoen aan $y' = \sin(x)y + \sin(2x)$?), differentievergelijkingen (vind een formule voor $a(n)$ als je weet dat $a(n+1) = n^2a(n) - 1$), sommeren (geef een formule voor $\sum_{k=1}^n k^5$) en nog heel veel meer.

In deze cursus willen we aangeven hoe de computer gebruikt wordt bij een aantal onderwerpen uit de algebra. Hoe zou je kunnen nagaan of een getal van wel 60 cijfers een priemgetal is? Hoe schrijf je een veelterm als produkt van veeltermen van een lagere graad? Bovendien gaan we in op een aantal praktische toepassingen van dit soort problemen.

Voor het aanpakken van vragen over bijvoorbeeld priemgetallen of veeltermen is een stuk algebra nodig. Het langste hoofdstuk van deze tekst bevat dan ook een inleiding in de algebra. We leren daarbij welke eigenschappen een concept zoals ‘de gehele getallen’ of ‘de veeltermen $f(X)$ met breuken als coëfficiënten’ gemeenschappelijk hebben, en tot welke structuur zoiets aanleiding geeft. Dit is nogal abstract, maar de toepassingen en de voorbeelden zijn dat niet en bovendien zullen we met deze nieuwe structuren veel oefeningen op de computer doen. Daarbij gebruiken we het pakket Maple.

De hier voor je liggende tekst is ontstaan tijdens de eerste keer dat de cursus gegeven werd: 9 weken lang 2 uur per week tussen december 2000 en februari 2001, waarbij we ongeveer de helft van de tijd gebruikt hebben voor het doen van vraagstukken op de computer. Graag wil ik Robert Carls bedanken voor zijn assistentie. Verder bedank ik alle studenten die dit eerste college volgden en daarbij door hun opmerkingen voor vele verbeteringen in de tekst hebben gezorgd: Ronald van Dijk, Jan Feitsma, Niels van Felijs, Elke van Gerwen, Fieke Geurts, Ronald Hoogma, Gert Jan Kamstra, Nadia Mensink, Georg Muntingh, Roelof Oosterhuis, Sander Platzer, Marcel Pot, Koen de Raedt, Sebastian Rozendal, Jeroen Rijnaard, Anisa Salomons, Marieke Scheijbeler, Jetze Sikkema, Joost Smid, Bert-Jan Steerneman, Jan Tuitman, Jacob Vosmaer, Marieke Weda, Abe Willemsma en Frank de Zeeuw. De in december 2002 toegevoegde veranderingen zijn mede te danken aan Martijn van Noort en Irene Polo.

2 Algebraïsche structuren

De objecten die we in deze cursus gebruiken zijn in vrijwel alle gevallen op een of andere manier opgebouwd uit gehele getallen. We beginnen met een paar voorbeelden, en vervolgens gaan we diverse mogelijkheden een naam geven.

2.1 voorbeelden

Voorbeeld 2.1.1 *Breuken* (rationale getallen), zijn quotiënten a/b waarin a en b gehele getallen zijn, met $b \neq 0$, en we hebben de regel $a/b = c/d$ indien $ad = bc$. Een breuk is dus opgebouwd uit een paar (a, b) van gehele getallen. Preciezer gezegd, het is een equivalentieklasse van paren (a, b) met $b \neq 0$, voor de equivalentierelatie $(a, b) \sim (c, d)$ precies dan, als $ad = bc$.

Breuken kunnen we optellen, aftrekken en vermenigvuldigen en zelfs op elkaar delen (op één bekende uitzondering na: delen door nul gaat niet). Hierbij gaan een aantal bekende regels op, zoals dat $x(y + z) = xy + xz$ voor breuken x, y en z . Zoals we straks zullen zien wordt dit algebraïsch verwoord door te zeggen dat de verzameling \mathbb{Q} van alle breuken een *lichaam* is.

Voorbeeld 2.1.2 De getallen modulo 7 kunnen we ook optellen, aftrekken en vermenigvuldigen. Zo is gewoonlijk $3 \cdot 4 = 12$, en omdat $12 \bmod 7$ gelijk is aan $5 \bmod 7$ schrijven we $(3 \bmod 7) \cdot (4 \bmod 7) = 5 \bmod 7$. Op dezelfde manier is $(6 \bmod 7) \cdot (5 \bmod 7) = 2 \bmod 7$ en $(4 \bmod 7) - (5 \bmod 7) = 6 \bmod 7$.

Iets minder vanzelfsprekend is het feit dat je (alweer met uitzondering van het delen door nul) getallen modulo 7 op elkaar kan delen. Bijvoorbeeld geldt $(3 \bmod 7) : (5 \bmod 7) = 2 \bmod 7$, immers $(2 \bmod 7) \cdot (5 \bmod 7) = 3 \bmod 7$.

Er zijn precies 7 verschillende getallen modulo 7. Samen vormen ze een verzameling met een optelling, een vermenigvuldiging enzovoort. Deze verzameling noteert men als $\mathbb{Z}/7\mathbb{Z}$ of ook wel als \mathbb{F}_7 . Evenals \mathbb{Q} is dit een lichaam.

Voorbeeld 2.1.3 Met de getallen modulo 8 is de situatie iets anders. Ook deze kunnen we optellen, aftrekken en vermenigvuldigen, maar het op elkaar delen van getallen modulo 8 lukt niet altijd, zelfs al zijn beide ongelijk aan nul. Zo kan je bijvoorbeeld geen uitkomst toekennen aan $(3 \bmod 8) : (2 \bmod 8)$. Immers, zou hier $a \bmod 8$ uitkomen, dan zou moeten gelden $(a \bmod 8) \cdot (2 \bmod 8) = 3 \bmod 8$. Welke mogelijkheid we ook voor a proberen, dit blijkt niet het geval te zijn.

Ook al lukt het wel om een uitkomst aan een deling toe te kennen, dan nog kan er iets vreemds gebeuren: beschouw maar eens $(6 \bmod 8) : (2 \bmod 8)$.

Als antwoord zou je $3 \bmod 8$ kunnen geven, immers $(3 \bmod 8) \cdot (2 \bmod 8) = 6 \bmod 8$. Echter ook $7 \bmod 8$ is een correcte uitkomst! Dus het kan zowel voorkomen dat er helemaal geen uitkomst is, als dat er meerdere zijn.

Algebraïci vatten dit samen door te zeggen dat de getallen modulo 8 met de optelling en vermenigvuldiging erop een *ring* vormen, maar geen lichaam. Deze ring wordt genoteerd als $\mathbb{Z}/8\mathbb{Z}$.

2.2 ringen en lichamen: definities

Definitie 2.2.1 Een *ring* bestaat uit een verzameling R waarin twee elementen zijn aangewezen die we $0 \in R$ en $1 \in R$ noemen, en verder zijn op R twee bewerkingen voorgeschreven die we ‘optellen’ en ‘vermenigvuldigen’ noemen. Optellen voegt aan $a, b \in R$ een element uit R toe dat genoteerd wordt als $a + b$, en vermenigvuldigen voegt aan $a, b \in R$ een element uit R genoteerd als $a \cdot b$ of ook wel als ab toe. Hierbij moet aan de volgende regels zijn voldaan:

R1 Voor alle $a, b \in R$ geldt $a + b = b + a$.

R2 Voor alle $a, b, c \in R$ geldt $(a + b) + c = a + (b + c)$.

R3 Voor elke $a \in R$ geldt $a + 0 = a$.

R4 Bij elke $a \in R$ is een $b \in R$ te vinden met $a + b = 0$.

R5 Voor alle $a, b, c \in R$ geldt $(ab)c = a(bc)$.

R6 Voor elke $a \in R$ geldt $1 \cdot a = a$ en $a \cdot 1 = a$.

R7 Voor alle $a, b, c \in R$ geldt $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ en $(a + b) \cdot c = (ac) + (bc)$.

Voorbeeld 2.2.2 De gehele getallen \mathbb{Z} met de gebruikelijke optelling en vermenigvuldiging leveren een voorbeeld van een ring. Evenzo is dat het geval voor \mathbb{Q} , voor de reële getallen \mathbb{R} , voor de complexe getallen \mathbb{C} , en voor de verzameling $\mathbb{Z}/8\mathbb{Z}$ bestaande uit de gehele getallen modulo 8.

Maar bijvoorbeeld de natuurlijke getallen \mathbb{N} vormen met de gewone vermenigvuldiging en optelling geen ring. Immers, voor $2 \in \mathbb{N}$ kan je geen $b \in \mathbb{N}$ vinden zodat $2 + b = 0$. Zo'n getal b bestaat wel, maar niet onder de niet-negatieve gehele getallen.

De deelverzameling $\{-1, 0, 1\}$ van de gehele getallen is, als we er de gewone optelling en vermenigvuldiging op zetten, evenmin een ring. Het lijkt er in dit voorbeeld misschien sterk op dat *wel* aan alle eisen voor een ring

is voldaan. Er is maar één probleem: de optelling is niet goed gedefinieerd. Want als we 1 bij 1 willen optellen krijgen we 2, en dat is geen element van onze verzameling. Hier zie je een beetje waar de naam ring vandaan komt: zoals een gewone ronde ring helemaal gesloten is, er zit nergens een begin of een einde, zo is ook een wiskundige ‘ring’ gesloten onder de bewerkingen optellen en vermenigvuldigen.

2.2.3

De eerste vier eisen in de definitie van een ring gaan over het optellen, de volgende twee over het vermenigvuldigen en alleen de laatste over het combineren van die twee bewerkingen. Hoewel het accent in deze cursus zal liggen op het rekenen in/met ringen en niet zo zeer op de algemene theorie erover, geven we toch een paar eigenschappen van ringen die direct uit de definitie af te leiden zijn.

In een ring R geldt voor elke $a \in R$ dat $a \cdot 0 = 0 = 0 \cdot a$. Om dit in te zien merken we in de eerste plaats op dat vanwege R3, toegepast op $a = 0$, geldt $0 + 0 = 0$. Verder bestaat er, wegens R4 toegepast op het element $a \cdot 0 \in R$, een $b \in R$ met $b + (a \cdot 0) = 0$. En dan is

$$\begin{aligned} 0 &= (a \cdot 0) + b && \text{(zo is } b \text{ gekozen)} \\ &= (a \cdot (0 + 0)) + b && \text{(omdat } 0 + 0 = 0) \\ &= ((a \cdot 0) + (a \cdot 0)) + b && \text{(vanwege R7)} \\ &= (a \cdot 0) + ((a \cdot 0) + b) && \text{(vanwege R2)} \\ &= a \cdot 0 && \text{(vanwege R3 en de keuze van } b). \end{aligned}$$

Op net zo'n manier kan je aantonen dat ook $0 \cdot a = 0$.

In een ring R geldt verder dat er bij elke $a \in R$ precies één $b \in R$ bestaat waarvoor geldt $a + b = 0$. Immers, als $a + b_1 = 0$ maar ook $a + b_2 = 0$, dan volgt

$$\begin{aligned} b_1 &= 0 + b_1 && \text{(vanwege R3 en R1)} \\ &= (a + b_2) + b_1 && \text{(omdat we aannemen } a + b_2 = 0) \\ &= a + (b_1 + b_2) && \text{(vanwege R2 en R1)} \\ &= (a + b_1) + b_2 && \text{(vanwege R2)} \\ &= b_2 && \text{(omdat } a + b_1 = 0 \text{ en vanwege R3)}. \end{aligned}$$

Die ene $b \in R$ die we zo bij een $a \in R$ krijgen, noteren we vanaf nu als $-a$. Dus voor een $a \in R$ is $-a \in R$ het enige element uit R dat voldoet aan $a + (-a) = 0$. Deze $-a$ wordt vaak de *tegengestelde* van $a \in R$ genoemd.

Nu we de tegengestelde van een element uit een ring R hebben gedefinieerd, kunnen we ook zeggen wat *afrekken* in R is: voor $a, b \in R$ schrijven we $a - b := a + (-b)$.

Voor elementen a, b in een ring R geldt $(-a)b = -(ab) = a \cdot (-b)$; anders gezegd, de tegengestelde van ab krijg je door de tegengestelde van a met b te vermenigvuldigen, en ook door a met de tegengestelde van b te vermenigvuldigen. Om te laten zien dat dit inderdaad zo is, moeten we nagaan dat $(-a)b$ (en ook $a \cdot (-b)$) de eigenschap heeft waarmee de tegengestelde van ab wordt vastgelegd. Anders gezegd, we moeten aantonen dat $(ab) + ((-a)b) = 0$. Welnu,

$$\begin{aligned} (ab) + ((-a)b) &= (a + (-a))b && \text{(vanwege R7)} \\ &= 0 \cdot b && \text{(dat is de definitie van } -a) \\ &= 0 && \text{(dat hadden we al afgeleid).} \end{aligned}$$

Definitie 2.2.4 Een *commutatieve ring* is een ring R waarin bovendien nog geldt $ab = ba$ voor alle $a, b \in R$.

Voorbeeld 2.2.5 Alle ringen die we totnutoe in dit hoofdstuk gezien hebben zijn voorbeelden van commutatieve ringen. We zullen ons verderop ook uitsluitend met commutatieve ringen bezighouden, maar hier willen we toch even opmerken dat er ook heel veel ringen zijn waarin elementen a, b voorkomen met $ab \neq ba$. We noemen twee zulke ringen.

De *matrixring* $M_2(\mathbb{Z})$ bestaat uit alle 2×2 matrices met coëfficiënten in \mathbb{Z} . Dat wil zeggen: een element van $M_2(\mathbb{Z})$ heeft de gedaante $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, waarin $a, b, c, d \in \mathbb{Z}$. Het optellen van twee zulke matrices gaat coëfficiëntsgewijs, dus

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}.$$

Vermenigvuldigen doen we met de regel

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}.$$

Met $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ als $0 \in M_2(\mathbb{Z})$ en $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ als $1 \in M_2(\mathbb{Z})$ krijgen we zo inderdaad een ring, zoals met een heleboel rekenwerk zou kunnen worden nagegaan. In deze ring is (bijvoorbeeld)

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

zoals je vrij snel kunt nagaan. Overigens hadden we hier in plaats van \mathbb{Z} net zo goed een willekeurige andere ring R kunnen nemen, en dan ontstaat de matrixring $M_2(R)$.

Een tweede voorbeeld van een niet-commutatieve ring wordt gegeven door de *quaternionen* \mathbb{H} . De manier waarop deze op maandag 16 oktober 1843 voor het eerst door Sir William Rowan Hamilton (lees

www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Hamilton.html

er maar op na) werden gegeven, doet sterk denken aan de manier waarop de complexe getallen \mathbb{C} vanuit de reële getallen \mathbb{R} worden gemaakt. \mathbb{H} bestaat uit alle uitdrukkingen $a + bi + cj + dk$, waarbij $a, b, c, d \in \mathbb{R}$ en i, j, k zijn drie nieuwe symbolen. Optellen gaat volgens

$$(a + bi + cj + dk) + (e + fi + gj + hk) = (a + e) + (b + f)i + (c + g)j + (d + h)k.$$

Het vermenigvuldigen wordt beschreven met de regels $ai = ia$ en $aj = ja$ en $ak = ka$ voor elke $a \in \mathbb{R}$, verder $i^2 = j^2 = k^2 = -1$ en tenslotte $ij = -ji = k$. Uit deze regels volgt dan dat bijvoorbeeld $ki = (-ji)i = (-1)j(-1) = j$, enzovoort.

Het bijzondere van de quaternionen is, dat er bij elke $z \in \mathbb{H}$ ongelijk aan 0 een $w \in \mathbb{H}$ te vinden is waarvoor geldt $zw = wz = 1$. Is namelijk $z = a + bi + cj + dk$, dan voldoet $w = (a^2 + b^2 + c^2 + d^2)^{-1}(a - bi - cj - dk)$ hieraan zoals met enig rekenwerk kan worden nagegaan.

Definitie 2.2.6 Een *lichaam* is een commutatieve ring R met twee extra eigenschappen: er geldt $1 \neq 0$ in R , en verder is er bij iedere $a \neq 0$ in R een $b \in R$ te vinden waarvoor geldt dat $ab = 1$.

Voorbeeld 2.2.7 De ring \mathbb{Z} is geen lichaam, want voor onder andere $2 \in \mathbb{Z}$ bestaat geen $b \in \mathbb{Z}$ zodat $2b = 1$. Zo'n b bestaat wel binnen bijvoorbeeld de reële getallen, namelijk $b = \frac{1}{2}$. Maar dat is nu eenmaal geen geheel getal.

Evenmin vormen de quaternionen \mathbb{H} een lichaam. Weliswaar hebben we daar bij elke $a \neq 0$ een b met $ab = 1$, maar \mathbb{H} is geen *commutatieve* ring.

Wel zijn \mathbb{Q}, \mathbb{R} en \mathbb{C} alle drie lichamen. En evenzo is $\mathbb{Z}/7\mathbb{Z}$ een lichaam. Immers, deze laatste is een commutatieve ring, en aan de regels $(1 \bmod 7) \cdot (1 \bmod 7) = 1 \bmod 7$ en $(2 \bmod 7) \cdot (4 \bmod 7) = 1 \bmod 7$ en $(3 \bmod 7) \cdot (5 \bmod 7) = 1 \bmod 7$ en $(6 \bmod 7) \cdot (6 \bmod 7) = 1 \bmod 7$ zien we dat elk element ongelijk aan nul in $\mathbb{Z}/7\mathbb{Z}$ ergens mee te vermenigvuldigen is zodat het product 1 wordt.

De ring $\mathbb{Z}/8\mathbb{Z}$ is geen lichaam, want er bestaat geen element $b \bmod 8$ zodat $(2 \bmod 8) \cdot (b \bmod 8) = 1 \bmod 8$. Probeer maar alle mogelijke b 's.

Definitie 2.2.8 Een element a in een ring R met de eigenschap dat er een $b \in R$ bestaat zodat zowel ab als ba gelijk zijn aan 1, heet een *eenheid*. De verzameling van alle eenheden in de ring R wordt geschreven als R^* .

2.2.9

Net zoals er bij een willekeurig element in een ring precies één tegengestelde bestaat, bestaat er ook bij elke eenheid a in een ring R precies één $b \in R$ met $ab = ba = 1$. Immers, stel dat zowel $b_1 \in R$ als $b_2 \in R$ voldoen aan $ab_i = b_i a = 1$. Dan volgt dat $b_1 = 1 \cdot b_1 = (b_2 a) b_1 = b_2 (a b_1) = b_2 \cdot 1 = b_2$; ga zelf na welke eigenschappen uit de definitie van een ring hier gebruikt zijn.

De $b \in R$ die op deze manier bij een eenheid $a \in R$ hoort wordt de *inverse* van a genoemd, notatie a^{-1} .

Een lichaam is dus een commutatieve ring waarin ieder element $\neq 0$ een inverse heeft en waarin bovendien $1 \neq 0$.

De eis dat in een lichaam moet gelden $1 \neq 0$ ziet er natuurlijk een beetje gek uit. Er wordt een enkel apart geval mee uitgesloten: stel namelijk dat in een ring R wel geldt dat $1 = 0$. Neem een willekeurig element $a \in R$. Dan is $a = a \cdot 1 = a \cdot 0 = 0$. Kortom, elk element uit deze ring is gelijk aan 0, oftewel, onze ring bestaat uit slechts één element. En dit geval wil men dus in de definitie van een lichaam uitsluiten. In plaats van $1 \neq 0$ hadden we dus net zo goed kunnen zeggen dat de ring minstens twee elementen moet hebben.

Voorbeeld 2.2.10 Voor de ring $\mathbb{Z}/8\mathbb{Z}$ geldt dat

$$(\mathbb{Z}/8\mathbb{Z})^* = \{1 \bmod 8, 3 \bmod 8, 5 \bmod 8, 7 \bmod 8\}.$$

Ieder van deze vier elementen is met zichzelf vermenigvuldigd gelijk aan $1 \bmod 8$, dus er geldt $x^{-1} = x$ voor elke $x \in (\mathbb{Z}/8\mathbb{Z})^*$.

2.3 ringen en lichamen: constructies

We geven hier twee constructies om bij een gegeven ring een andere ring te maken.

2.3.1 Polynoomringen

Gegeven een ring R . De *polynoomring* over R in de variabele X , notatie $R[X]$, bestaat uit alle (formele) uitdrukkingen $r_n X^n + r_{n-1} X^{n-1} + \dots + r_1 X + r_0$, waarin n een willekeurig niet-negatief geheel getal mag zijn en $r_n, \dots, r_1, r_0 \in R$. We spreken daarbij af dat $0X^j = 0$. Elk element r van R is ook op te vatten als een element van $R[X]$ (namelijk met $n = 0$ en $r_0 = r$). In het bijzonder zijn $0 \in R$ en $1 \in R$ dus ook elementen van $R[X]$. Deze nemen we als 0 en als 1 in de definitie van de ringstructuur op $R[X]$. Het optellen in $R[X]$ gaat op een heel gewone manier:

$$(r_n X^n + r_{n-1} X^{n-1} + \dots + r_1 X + r_0) + (s_m X^m + \dots + s_1 X + s_0) = t_k X^k + \dots + t_0,$$

waarin $k = \max(n, m)$ en $t_i = r_i + s_i$ voor $i \leq \min(m, n)$ en $t_i = r_i$ als $n \geq i > m$, en $t_i = s_i$ als $m \geq i > n$. Ook het vermenigvuldigen van polynomen gaat zoals we dat gewend zijn. In een formule ziet dat er uit als

$$(r_n X^n + r_{n-1} X^{n-1} + \dots + r_1 X + r_0) \cdot (s_m X^m + \dots + s_1 X + s_0) = t_k X^k + \dots + t_0,$$

waarbij $k = n + m$, en voor een i met $0 \leq i \leq k$ is $t_i = r_i s_0 + r_{i-1} s_1 + r_{i-2} s_2 + \dots + r_1 s_{i-1} + r_0 s_i$, waarbij we schrijven $r_j = 0$ als $j > n$ en evenzo $s_j = 0$ als $j > m$.

Het is een vervelend maar niet erg moeilijk gebeuren om na te gaan dat op deze manier $R[X]$ inderdaad een ring wordt.

Door deze definitie herhaald toe te passen krijgen we de *polynoomring* in n variabelen $R[X_1, X_2, \dots, X_n]$. Dus bijvoorbeeld $R[X, Y] = (R[X])[Y]$. Deze bestaat uit veeltermen $f_n Y^n + \dots + f_1 Y + f_0$, waarbij elk van de f_i zelf weer een polynoom is in de variabele X met coëfficiënten in R . Schrijven we dat helemaal uit, dan krijgen we uitdrukkingen

$$r_{00} + r_{10}X + r_{01}Y + r_{11}XY + \dots + r_{ij}X^i Y^j + \dots + r_{mn}X^m Y^n.$$

Definitie 2.3.2 De *graad* van een polynoom $f \in R[X]$ met $f \neq 0$ is de hoogste macht van X die in f voorkomt met een coëfficiënt $\neq 0$. Dit noteren we als $\text{graad}(f)$. Voor het geval $f = 0$ bestaat er niet zo'n hoogste macht waarvoor de een coëfficiënt $\neq 0$ is; in dit geval spreken we af $\text{graad}(0) = -\infty$.

Voorbeeld 2.3.3 Stel $f = r_n X^n + \dots + r_1 X + r_0 \in R[X]$ heeft graad n en $g = s_m X^m + \dots + s_1 X + s_0 \in R[X]$ heeft graad m , dus $r_n \neq 0$ en $s_m \neq 0$. Dan geldt $\text{graad}(f + g) = \max(n, m)$, behalve wanneer $n = m \geq 0$ en bovendien $r_n = -s_n$. In dat geval kunnen we alleen maar zeggen dat $\text{graad}(f + g) < \max(n, m)$.

Voor de graad van het product geldt iets soortgelijks. Namelijk, de hoogste macht van X die in fg met een coëfficiënt $\neq 0$ voor zou kunnen komen, is X^{n+m} . Daarvan is de coëfficiënt $r_n s_m$. Nu weten we dat $r_n \neq 0$ en $s_m \neq 0$, maar dat zegt niets over het eventueel nul zijn van het product $r_n s_m$. Bijvoorbeeld in de ring \mathbb{Z} is een product van twee elementen die beide niet nul zijn ook altijd ongelijk aan nul, maar in de ring $\mathbb{Z}/6\mathbb{Z}$ zijn zowel $2 \bmod 6$ als $3 \bmod 6$ niet nul, terwijl hun product *wel* nul is.

We zeggen dat de ring R een ring is zonder nuldelers, wanneer uit een gelijkheid $a \cdot b = 0$ voor $a, b \in R$ volgt, dat $a = 0$ of $b = 0$. Omgekeerd, elementen a, b in een ring R met $a \cdot b = 0$ terwijl toch $a \neq 0$ en $b \neq 0$ heten *nuldelers*.

Is R een ring zonder nuldelers, dan volgt dus dat $\text{graad}(fg) = \text{graad}(f) + \text{graad}(g)$ voor alle $f, g \in R[X]$. In het bijzonder kan je met behulp van deze

regel de eenheden $R[X]^*$ bepalen indien $R \neq \{0\}$ een ring zonder nuldelers is: is namelijk $f \in R[X]^*$, dan bestaat er een $g \in R[X]$ met $fg = 1 = gf$. Voor deze f en g geldt dan $\text{graad}(f) + \text{graad}(g) = \text{graad}(1) = 0$. Hieruit volgt dat f en g beide graad 0 hebben, met andere woorden $f \in R$ en $g \in R$. Maar dit betekent dat f al een eenheid is in R . Conclusie: $R[X]^* = R^*$.

2.3.4 quotiëntringen

We zullen er vanaf nu steeds van uit gaan dat we werken met een commutatieve ring R .

Definitie 2.3.5 Een *ideaal* in een commutatieve ring R is een deelverzameling $I \subset R$ die voldoet aan de volgende drie eigenschappen:

- I1 Het element $0 \in R$ zit in I ;
- I2 Als $a, b \in I$ dan ook hun som $a + b \in I$.
- I3 Als $a \in I$, dan geldt voor *elke* $r \in R$ dat ook $ra \in I$.

Voorbeeld 2.3.6 De gehele getallen die eindigen op een nul vormen een ideaal in \mathbb{Z} . Immers, 0 eindigt op een nul, de som van twee getallen die beide op nul eindigen doet dat ook, en als een getal op nul eindigt en je vermenigvuldigt het met iets geheels, dan eindigt het product opnieuw op nul.

De natuurlijke getallen \mathbb{N} (inclusief 0) vormen *geen* ideaal in \mathbb{Z} . Ze voldoen weliswaar aan I1 en aan I2, maar bij I3 gaat het mis: bijvoorbeeld is $5 \in \mathbb{N}$, en $-1 \in \mathbb{Z}$, maar hun product $-5 \notin \mathbb{N}$.

Als we binnen de ring $\mathbb{Z}[X]$ alle veeltermen nemen die het complexe getal $\sqrt{2} + i$ als nulpunt hebben, dan hebben we een ideaal in $\mathbb{Z}[X]$, zoals redelijk snel is na te gaan. Een voorbeeld van een element $\neq 0$ in dit ideaal is $X^4 - 2X^2 + 9$.

Definitie 2.3.7 Laat R een commutatieve ring zijn en $a_1, \dots, a_n \in R$. Het *ideaal voortgebracht door* a_1, \dots, a_n bestaat uit alle mogelijke uitdrukkingen

$$r_1 a_1 + r_2 a_2 + \dots + r_n a_n,$$

waarbij de r_i tot en met r_n de elementen van R doorlopen.

Deze verzameling wordt genoteerd als $Ra_1 + \dots + Ra_n$ of ook wel als (a_1, \dots, a_n) .

Zo'n verzameling $I = Ra_1 + \dots + Ra_n$ is inderdaad een ideaal: $0 \in I$ omdat $0 = 0 \cdot a_1 + \dots + 0 \cdot a_n$. Verder is voor twee uitdrukkingen van de gevraagde vorm natuurlijk ook hun som weer van die vorm, en evenzo het product met een element uit R .

Voorbeeld 2.3.8 In \mathbb{Z} zijn de idealen (2) en $(4, 6)$ aan elkaar gelijk. Immers, een veelvoud van 4 plus een veelvoud van 6, oftewel een getal $4n + 6m$, is ook te schrijven als $2(2n + 3m)$ oftewel als een veelvoud van 2. Dus er geldt $(4, 6) \subset (2)$. Omgekeerd is een veelvoud van 2, oftewel een getal $2k$ te schrijven als $6k + 4 \cdot (-k)$, dus als een veelvoud van 4 plus een veelvoud van 6. Dus ook $(2) \subset (4, 6)$, en daarmee is bewezen dat de beide verzamelingen hetzelfde zijn.

Bij ieder ideaal I in een commutatieve ring R gaan we nu een nieuwe ring construeren, de *quotiëntring*, ook wel genoemd *factorring* die we als R/I zullen noteren. Een eerste voorbeeld hiervan hebben we al gezien: bij het ideaal $n\mathbb{Z}$ in de ring \mathbb{Z} hebben we de ring bestaande uit 'de gehele getallen modulo n '. Is bijvoorbeeld $n = 8$ dan heeft deze ring als elementen alle $a \bmod 8$. En bijvoorbeeld $13 \bmod 8 = 5 \bmod 8$; algemener: twee elementen $a \bmod 8$ en $b \bmod 8$ zijn hetzelfde precies dan, als we door bij a een veelvoud van 8 op te tellen b kunnen krijgen. Of anders gezegd, als het verschil van a en b een veelvoud van 8 is, oftewel een element is van het ideaal $8\mathbb{Z}$. Op eenzelfde manier gaan we nu meer algemeen de factorring R/I introduceren.

Definitie 2.3.9 Gegeven een ideaal I in een commutatieve ring R . Het quotiënt R modulo I , notatie R/I , bestaat uit alle uitdrukkingen $r \bmod I$ waarbij $r \in R$. Hierbij spreken we af dat $r_1 \bmod I = r_2 \bmod I$ precies dan als $r_1 - r_2 \in I$.

Voorbeeld 2.3.10 Zoals al gezegd, is $\mathbb{Z}/n\mathbb{Z}$ een voorbeeld van zo'n quotiënt. Een wat moeilijker voorbeeld dat we in feite ook al kennen, krijg je door $R = \mathbb{R}[X]$ te nemen en $I = (X^2 + 1)R$. Dan is in R/I bijvoorbeeld $X^2 \bmod I = -1 \bmod I$, want het verschil $X^2 - (-1)$ zit in I . Evenzo is $X^3 \bmod I = -X \bmod I$. En meer algemeen geldt voor elke $n \geq 2$ dat $X^n \bmod I = -X^{n-2} \bmod I$ en ook $-X^n \bmod I = X^{n-2} \bmod I$. Kortom, we kunnen de exponent van X herhaald met 2 verlagen, en elke keer dat we dit doen verandert het teken voor de macht van X . Als we beginnen met een even macht van X , dus met $X^{2m} \bmod I$, dan kunnen we precies m keer de exponent met 2 verlagen, en we komen dan uit op $(-1)^m \bmod I$. Er geldt dus dat $X^{2m} \bmod I = (-1)^m \bmod I$. En waren we met een oneven macht van X begonnen, dus met $X^{2m+1} \bmod I$, dan levert eenzelfde redenering dat

dit gelijk is aan $(-1)^m X \bmod I$. Meer algemeen, als we in $f(X) \bmod I$ alle voorkomende machten van X zo vaak we maar kunnen met 2 verlagen, dan komen we uiteindelijk uit op een veelterm van de vorm $a + bX$ voor zekere $a, b \in \mathbb{R}$, met de eigenschap dat $f(X) \bmod I = a + bX \bmod I$. Kortom, alle elementen van $\mathbb{R}[X]/(X^2+1)$ zijn te schrijven in de vorm $a + bX \bmod (X^2+1)$ waarin $a, b \in \mathbb{R}$. En elementen die zo geschreven zijn, zijn gemakkelijker te onderscheiden: is namelijk $a_1 + b_1X \bmod I = a_2 + b_2X \bmod I$, dan betekent dit dat $a_1 + b_1X - a_2 - b_2X$ een element is van I , oftewel een veelvoud is van $X^2 + 1$. Echter, een veelterm $(X^2 + 1) \cdot g(X)$ met $g(X) \in \mathbb{R}[X]$ heeft graad ≥ 2 , tenzij $g(X) = 0$, en in dat geval is ook het product nul. Dus we concluderen dat $a_1 + b_1X \bmod I$ en $a_2 + b_2X \bmod I$ alleen aan elkaar gelijk zijn wanneer $a_1 = a_2$ en $b_1 = b_2$. Hiermee hebben we een precieze beschrijving van R/I in dit geval.

Definitie 2.3.11 Gegeven een ideaal I in een commutatieve ring R en twee elementen $a, b \in R$.

Het *optellen modulo I* van elementen in R/I gaat met de regel

$$(a \bmod I) + (b \bmod I) = (a + b) \bmod I.$$

Evenzo wordt het *vermenigvuldigen modulo I* van elementen in R/I gegeven door

$$(a \bmod I) \cdot (b \bmod I) = ab \bmod I.$$

Deze definitie levert inderdaad twee bewerkingen op R/I . Om dat in te zien gebruiken we echt dat I een *ideaal* is. In theorie was het namelijk denkbaar dat voor zekere $a_1, a_2, b_1, b_2 \in R$ zou kunnen gelden $a_1 \bmod I = a_2 \bmod I$, evenzo $b_1 \bmod I = b_2 \bmod I$, maar toch $a_1 b_1 \bmod I \neq a_2 b_2 \bmod I$ of iets soortgelijks voor de optelling. Dit is evenwel niet het geval. Wat we daartoe moeten inzien, vanwege de definitie van gelijkheid modulo I , is dat als $a_1 - a_2 \in I$ en ook $b_1 - b_2 \in I$, dan volgt daaruit dat $a_1 + b_1 - (a_2 + b_2) \in I$ en tevens $a_1 b_1 - a_2 b_2 \in I$. Welnu, voor de som volgt dat uit eigenschap I2 van een ideaal; immers $a_1 + b_1 - (a_2 + b_2) = (a_1 - a_2) + (b_1 - b_2)$. Voor het product gebruiken we dat $a_1 b_1 - a_2 b_2 = a_1(b_1 - b_2) + b_2(a_1 - a_2)$. Met behulp van zowel I3 als I2 volgt dat dit inderdaad een element uit I is.

Het is niet erg moeilijk om na te gaan dat met deze bewerkingen de factorring R/I inderdaad een ring is. De nul in deze ring is $0 \bmod I$ en de 1 is $1 \bmod I$.

Voorbeeld 2.3.12 We hebben al gezien dat $\mathbb{R}[X]/(X^2 + 1)$ precies bestaat uit de elementen $a + bX \bmod (X^2 + 1)$. Het optellen van twee zulke elementen

gaat door zowel de a 's als de b 's op te tellen. Voor het vermenigvuldigen hebben we

$$\begin{aligned}(a + bX \bmod I) \cdot (c + dX \bmod I) &= ac + (ad + bc)X + bdX^2 \bmod I \\ &= ac - bd + (ad + bc)X \bmod I,\end{aligned}$$

met $I = (X^2 + 1)$. Dit gaat met precies dezelfde formules als het optellen en het vermenigvuldigen van complexe getallen! De conclusie is dat de factorring $\mathbb{R}[X]/(X^2 + 1)$ in feite dezelfde ring is als \mathbb{C} , wanneer we $a + bX \bmod (X^2 + 1)$ identificeren met $a + bi$.

Voorbeeld 2.3.13 We nemen als ring $R = \mathbb{Z}[X]$ en in R het ideaal $I = (2, X^2 - X - 1)$. Dan bestaat I precies uit de veeltermen van de vorm $2f(X) + (X^2 - X - 1)g(X)$, met $f(X), g(X) \in \mathbb{Z}[X]$. In het bijzonder (kies hier $g(X) = 0$) zitten alle veeltermen met alleen even coëfficiënten in I . Hieruit volgt dat een willekeurige $h(X) \bmod I$ uit R/I ook te schrijven is als $e(X) \bmod I$, waarin $e(X)$ een veelterm is met alleen nullen en enen als coëfficiënten. Verder is natuurlijk $X^2 \bmod I = X + 1 \bmod I$, want het verschil $X^2 - (X + 1)$ zit in I . Nu volgt dat

$$\begin{aligned}X^3 \bmod I &= (X \bmod I) \cdot (X^2 \bmod I) \\ &= (X \bmod I) \cdot (X + 1 \bmod I) \\ &= X^2 + X \bmod I \\ &= (X^2 \bmod I) + (X \bmod I) \\ &= 2X + 1 \bmod I \\ &= 1 \bmod I.\end{aligned}$$

Hiermee kunnen we dan alle machten van X modulo I vereenvoudigen: $X^{3m} \bmod I = (X^3 \bmod I)^m = 1 \bmod I$ en $X^{3m+1} \bmod I = (X \bmod I) \cdot (X^{3m} \bmod I) = X \bmod I$ en $X^{3m+2} \bmod I = (X^2 \bmod I) \cdot (1 \bmod I) = X + 1 \bmod I$.

We zien dus dat voor de veelterm $e(X)$ met alleen nullen en enen als coëfficiënten geldt $e(X) \bmod I = a + bX \bmod I$, waarin we door eventueel nog wat veelvouden van 2 weg te halen mogen aannemen $a, b \in \{0, 1\}$. Kortom, de enige elementen van R/I zijn $0 \bmod I$, $1 \bmod I$, $X \bmod I$ en $1 + X \bmod I$. Deze vier zijn ook echt verschillend in R/I , want een verschil van twee van de gegeven veeltermen zit niet in I .

Het is niet moeilijk om na te gaan dat het optellen en vermenigvuldigen in R/I gaat volgens de volgende tabellen.

+	0 mod I	1 mod I	X mod I	1 + X mod I
0 mod I	0 mod I	1 mod I	X mod I	1 + X mod I
1 mod I	1 mod I	0 mod I	1 + X mod I	X mod I
X mod I	X mod I	1 + X mod I	0 mod I	1 mod I
1 + X mod I	1 + X mod I	X mod I	1 mod I	0 mod I

\cdot	$0 \bmod I$	$1 \bmod I$	$X \bmod I$	$1 + X \bmod I$
$0 \bmod I$	$0 \bmod I$	$0 \bmod I$	$0 \bmod I$	$0 \bmod I$
$1 \bmod I$	$0 \bmod I$	$1 \bmod I$	$X \bmod I$	$1 + X \bmod I$
$X \bmod I$	$0 \bmod I$	$X \bmod I$	$1 + X \bmod I$	$1 \bmod I$
$1 + X \bmod I$	$0 \bmod I$	$1 + X \bmod I$	$1 \bmod I$	$X \bmod I$

Uit de vermenigvuldigingstabel zien we in het bijzonder dat in dit voorbeeld elke $a \neq 0$ in R/I een *inverse* a^{-1} heeft. De ring $\mathbb{Z}[X]/(2, X^2 - X - 1)$ is dus zelfs een lichaam, en wel eentje die uit precies 4 elementen bestaat.

2.4 (eenheden)groepen

We gaan nu wat beter kijken naar de verzameling R^* bestaande uit alle eenheden in een ring R . Natuurlijk kan je eenheden optellen, maar of het resultaat dan weer een eenheid is hangt af van de gekozen eenheden. Zo is altijd $1 \in R$ een eenheid, want $1 \cdot 1 = 1$. Evenzo is de tegengestelde -1 van 1 een eenheid, maar de som van deze twee eenheden is 0 , en dat is alleen in de ring bestaande uit slechts één element een eenheid.

De situatie is veel mooier voor de vermenigvuldiging: het product van twee eenheden is namelijk zelf ook weer een eenheid. Immers, neem $a_1, a_2 \in R^*$. De definitie van ‘eenheid’ zegt dat er $b_1, b_2 \in R$ bestaan zodat $a_1 b_1 = b_1 a_1 = 1$ en $a_2 b_2 = b_2 a_2 = 1$. Voor het product $a_1 a_2$ geldt dan

$$(b_2 b_1)(a_1 a_2) = 1 = (a_1 a_2)(b_2 b_1)$$

zoals eenvoudig uit de eigenschappen van een ring is af te leiden. Hier staat dus dat inderdaad ook $a_1 a_2 \in R^*$.

We hebben nu gezien dat R^* meer is dan alleen maar een verzameling, namelijk het is een verzameling met een vermenigvuldiging erop die aan twee elementen $a, b \in R^*$ een nieuw element $a \cdot b \in R^*$ toevoegt. Verder geldt vanwege de definitie van een eenheid dat als $a \in R^*$, dan is ook $a^{-1} \in R^*$. Gebruik makend van de eigenschappen R5 en R6 van een ring, en van onderstaande definitie, zien we dat R^* een groep is met $1 \in R^*$ als eenheidselement.

Definitie 2.4.1 Een groep bestaat uit een verzameling G waarin een element $e \in G$ is aangewezen dat we het *eenheidselement* van G noemen, en verder is op G een bewerking voorgeschreven die we ‘vermenigvuldigen’ noemen. Vermenigvuldigen voegt aan $a, b \in G$ een element uit G toe dat genoteerd wordt als $a \cdot b$ of ook wel als ab . Hierbij moet aan de volgende regels voldaan zijn:

G1 Voor elk drietal $a, b, c \in G$ geldt $(a \cdot b) \cdot c = a \cdot (b \cdot c)$;

G2 Voor elke $a \in G$ geldt $a \cdot e = e \cdot a = a$;

G3 Bij elke $a \in G$ is een $b \in G$ te vinden met $a \cdot b = b \cdot a = e$.

Voorbeeld 2.4.2 Zoals al opgemerkt vormen de eenheden R^* in een ring R een groep onder vermenigvuldiging; deze groep wordt de *eenhedengroep* van de ring R genoemd.

Ook R zelf is op te vatten als groep, door als bewerking de optelling in R te nemen en $0 \in R$ als eenheidselement.

De deelverzameling $G = \{\pm 1, \pm i, \pm j, \pm k\}$ van de quaternionen \mathbb{H} vormen een groep met als bewerking erop het vermenigvuldigen in \mathbb{H} . Dit is een voorbeeld van een *niet-commutatieve groep*; dat wil zeggen dat er elementen $a, b \in G$ zijn aan te wijzen waarvoor geldt dat $a \cdot b \neq b \cdot a$. Een groep waarin wel voor elk paar elementen a, b geldt $ab = ba$ heet een *commutatieve groep* of ook wel *abelse groep*, naar de Noorse wiskundige Niels Henrik Abel (1802–1829); zie

www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Abel.html

2.4.3

Net als bij een eenheid in een ring, geldt ook voor een willekeurig element a in een groep G dat er slechts één $b \in G$ is waarvoor geldt $ab = ba = e$. Deze b wordt weer de *inverse* van a genoemd, en genoteerd als a^{-1} .

Toepassingen van groepen en ringen zoals we die verderop zullen behandelen hebben vaak te maken met machtsverheffen in een (eenheden)groep. Dat zien we bij het beveiligen van informatie, waar een bericht uit $(\mathbb{Z}/n\mathbb{Z})^*$ wordt versleuteld door het tot een geschikte macht te verheffen; we zien het ook bij primaliteitstests. Hier is een bruikbare methode om na te gaan of een zeker getal p een priemgetal is, dat men met behulp van dit getal een commutatieve ring R_p construeert met de eigenschap dat als R_p een lichaam is, dan is p priem. Nagaan of R_p een lichaam is betekent dat we willen weten of elk element uit R dat $\neq 0$ is, in R_p^* zit. Ruw gezegd komt dit erop neer dat we om aan te tonen dat p geen priemgetal zou zijn, moeten aantonen dat R_p^* niet al te groot is. En omgekeerd, p is *wel* een priemgetal als we weten dat R_p^* heel erg groot is.

Om iets te zeggen over de grootte van een groep G blijkt in veel gevallen het begrip *orde* nuttig.

Definitie 2.4.4 Gegeven een element a in een groep G . De *orde* van a , notatie $\text{ord}(a)$, is de *kleinste* $n \geq 1$ waarvoor geldt $a^n = e$. En als er niet

zo'n n bestaat, met andere woorden als $a^n \neq e$ voor elke $n > 0$, dan schrijven we $\text{ord}(a) = \infty$.

Voorbeeld 2.4.5 In de groep $(\mathbb{Z}/1000\mathbb{Z})^*$ is de orde van 3 mod 1000 gelijk aan 100. Met Maple is dat redelijk snel na te gaan; het commando waarmee een macht $a^n \bmod m$ wordt bepaald ziet er uit als

```
a^n mod m;
```

Dat de honderdste macht van 3 mod 1000 gelijk is aan 1 mod 1000 zie je dan heel snel. Wat dan nog over blijft, is aan te tonen dat je met een kleinere macht van 3 *niet* 1 mod 1000 kan krijgen. De volgende regel in Maple vertelt je dat inderdaad de orde hier 100 is:

```
n:=2: while (3^n mod 1000) <> 1 do n:=n+1 od: n;
```

Als in een groep G twee verschillende machten van een element $a \in G$ gelijk zijn, dan volgt daaruit dat $\text{ord}(a) < \infty$. Immers, stel dat $a^n = a^m$ waarbij $n < m$. Hier staat dat $a^{m-n} \cdot a^n = a^n$, en als we zowel het linker- als het rechterlid hierin met de inverse van a^n vermenigvuldigen dan zien we dat $a^{m-n} = e$. Daarmee hebben we een getal $k > 0$ gevonden met $a^k = e$, dus er bestaat ook een kleinste positieve k met die eigenschap.

Als de groep G uit slechts eindig veel elementen bestaat, dan komen voor elke $a \in G$ in de rij

$$a, a^2, a^3, a^4, a^5, \dots$$

elementen dubbel voor. Er volgt dus dat elk element uit een eindige groep een eindige orde heeft.

Hebben we een element a in een groep G en twee getallen n, m met $1 \leq n < m \leq \text{ord}(a)$, dan is $a^n \neq a^m$. Immers, het bovengegeven argument laat zien dat als $a^n = a^m$, dan was $a^{m-n} = e$ terwijl ook geldt $1 \leq m-n < \text{ord}(a)$. Dit is niet mogelijk vanwege de definitie van 'orde'. We concluderen hieruit, dat als we in een groep G een element van orde k hebben gevonden, dan heeft G minstens k elementen.

Voorbeeld 2.4.6 Neem $I = (7, X^3 + 6X + 2)$ in $\mathbb{Z}[X]$. Het element $X \bmod I$ in de ring $\mathbb{Z}[X]/I$ is een eenheid. Immers, er geldt

$$(X \bmod I) \cdot (3X^2 + 4 \bmod I) = 1 \bmod I$$

(want $3X^3 + 4X - 1 = 3(X^3 + 6X + 2) + (-2X - 1) \cdot 7 \in I$). Met de regel

```
n:=2: while (Powmod(x,n,x^3+6*x+2,x) mod 7) <> 1 do n:=n+1 od: n;
```

kan worden nagegaan dat $\text{ord}(X \bmod I) = 342$ in $(\mathbb{Z}[X]/I)^*$.

Elk element in $\mathbb{Z}[X]/I$ is te schrijven als $aX^2 + bX + c \bmod I$, met $a, b, c \in \{0, 1, 2, 3, 4, 5, 6\}$. Dus de hele ring $\mathbb{Z}[X]/I$ bestaat uit $7^3 = 343$ elementen. Er zijn 342 onderling verschillende machten van $X \bmod I$, en dat zijn allemaal eenheden. Hieruit concluderen we, dat elk element $\neq 0$ in de commutatieve ring $\mathbb{Z}[X]/I$ een eenheid is. Met andere woorden, $\mathbb{Z}[X]/I$ is een *lichaam*, en wel eentje met 343 elementen.

2.4.7

We geven tenslotte nog enkele nuttige eigenschappen van het begrip ‘orde’.

Geldt voor een element a in een groep G dat $a^n = e$ met $n \geq 1$, dan is de orde van a een deler van n .

Schrijf namelijk $k = \text{ord}(a)$. Dan is $k \leq n$. De getallen $k, 2k, 3k, 4k, \dots$ delen de positieve reële getallen op in disjuncte intervallen van lengte k . Als n geen veelvoud van k zou zijn, dan lag n ergens in één van die intervallen. Anders gezegd, er zou een $m > 0$ zijn met de eigenschap $mk < n < (m+1)k$. Hiervoor geldt dan $a^{mk} = (a^k)^m = e^m = e = a^n$. Hieruit volgt $a^{n-mk} = e$, terwijl ook $1 \leq n - mk < (m+1)k - mk = k$. Dit is in tegenspraak met het feit dat k de orde is van a . Conclusie: n is wel een veelvoud van de orde van a , zoals we wilden aantonen.

Bestaat de groep G uit precies n elementen en is $a \in G$, dan is $\text{ord}(a)$ een deler van n .

Schrijf namelijk weer $k = \text{ord}(a)$. De verzameling $H = \{a, a^2, \dots, a^k\}$ is dan, met dezelfde vermenigvuldiging als we op G hebben, ook zelf een groep. Immers $e = a^k \in H$, met a^m zit ook z'n inverse a^{k-m} in H , en de overige groepseigenschappen volgen uit de overeenkomstige voor de groep G . We moeten laten zien dat k een deler is van n , en dat gaan we doen door G op te delen in disjuncte delen die elk uit precies k elementen bestaan. Hebben we dan precies m zulke delen, dan volgt dat $n = mk$ oftewel n is een veelvoud van k .

Welnu, als delen van G nemen we alle deelverzamelingen

$$bH := \{ba, ba^2, \dots, ba^k\}.$$

Elk element van G zit in zo'n deelverzameling, want $b = b \cdot a^k \in bH$. Verder bestaat elke bH inderdaad uit precies k elementen, want als $ba^i = ba^j$ voor zekere i, j met $1 \leq i < j \leq k$, dan zie je door van links met de inverse van b te vermenigvuldigen dat ook zou gelden $a^i = a^j$ en dat is niet het geval omdat de orde van a gelijk is aan k . Tenslotte moeten we inzien dat

we hier een opdeling in *disjuncte* delen hebben. Stel daartoe dat twee zulke verzamelingen verschillend zijn, dus $b_1H \neq b_2H$. De bewering is dat ze dan geen enkel element gemeenschappelijk hebben. Stel namelijk dat ze dat toch hadden. Dan bestaan er i, j met $1 \leq i, j \leq k$ zodat $b_1a^i = b_2a^j$. Door beide leden aan de rechterkant met a^{k-i} te vermenigvuldigen volgt dat $b_1 = b_2a^\ell$ voor zekere ℓ . Maar dan is ook $b_1a = b_2a^{\ell+1}$ en $b_1a^2 = b_2a^{\ell+2}$ enzovoort; kortom, elk element van b_1H zit in b_2H . Omdat beide verzamelingen precies k elementen hebben, zijn ze dus gelijk. Echter onze aanname was dat ze dit niet waren, dus kennelijk is de veronderstelling dat ze dan toch nog iets gemeenschappelijk hadden onjuist. Hiermee is aangetoond dat inderdaad de gegeven opdeling er eentje is zoals gewenst, en dus is k een deler van n .

Als a een element is van een groep bestaande uit precies n elementen, dan is $a^n = e$.

Immers, de orde k van a is een deler van n zoals we hebben gezien. Dus $n = km$ voor zekere m , en dan ook $a^n = a^{km} = (a^k)^m = e^m = e$.

Voorbeeld 2.4.8 Neem $n = 2^{16} + 1 = 65537$. Er geldt $(5 \bmod n)^{2^{15}} = -1 \bmod n$, zoals met Maple heel gemakkelijk is na te rekenen. De 2^{16} de macht van $5 \bmod n$ is het kwadraat hiervan, en dat is $1 \bmod n$. Dus is $5 \bmod n \in (\mathbb{Z}/n\mathbb{Z})^*$, en de orde van $5 \bmod n$ in deze groep is een deler van 2^{16} . Maar die delers kennen we allemaal: het zijn

$$\{1, 2, 4, 8, 16, 2^5, 2^6, \dots, 2^{14}, 2^{15}, 2^{16}\}.$$

Zou de orde van $5 \bmod n$ *niet* 2^{16} zijn, dan was het dus een zekere deler van 2^{15} . En omdat 2^{15} dan een veelvoud was van die orde, zou gelden dat $(5 \bmod n)^{2^{15}} = 1 \bmod n$. We hebben echter al uitgerekend dat daar juist $-1 \bmod n$ uitkwam, en omdat $-1 \bmod n \neq 1 \bmod n$, is dus kennelijk de orde van $5 \bmod n$ geen deler van 2^{15} . Conclusie: die orde is 2^{16} , en dus bestaat $(\mathbb{Z}/n\mathbb{Z})^*$ in dit voorbeeld uit minstens $2^{16} = n - 1$ elementen. Ieder element $\neq 0$ in $\mathbb{Z}/65537\mathbb{Z}$ is dus een eenheid, en dus is $\mathbb{Z}/65537\mathbb{Z}$ een lichaam. We zullen verderop zien dat dit impliceert dat 65537 een priemgetal is.

Nu is het helemaal niet schokkend dat we zo van een getal van 5 cijfers kunnen inzien dat het een priemgetal is. Maar het is wel opmerkelijk hoeveel rekenstappen het kost om tot die conclusie te komen: We moesten er voor bepalen

$$\begin{aligned} b_1 &= (5 \bmod n)^2, \\ b_2 &= b_1^2 \quad (\text{en dat is } (5 \bmod n)^4) \\ &\dots \\ b_{i+1} &= b_i^2 \quad (\text{en dat is } (5 \bmod n)^{2^i}) \\ &\dots \\ b_{15} &= b_{14}^2 \quad (\text{en dat is } (5 \bmod n)^{15}). \end{aligned}$$

Dus al na 15 keer kwadrateren modulo n hebben we kunnen beredeneren dat $n = 2^{16} + 1$ een priemgetal is.

Voorbeeld 2.4.9 Als laatste voorbeeld kijken we naar $F_5 = 2^{32} + 1$. De getallen $F_0 = 2^1 + 1 = 3$, $F_1 = 2^2 + 1 = 5$, $F_2 = 2^4 + 1 = 17$, $F_3 = 2^8 + 1 = 257$ en $F_4 = 2^{16} + 1 = 65537$ zijn alle vijf priemgetallen. Zulke getallen $F_n = 2^{2^n} + 1$ heten *Fermatgetallen* naar de jurist Pierre de Fermat (1601–1665) uit Toulouse. Over Fermat is ontzettend veel op het internet te vinden; met name over wat de *Laatste stelling van Fermat* wordt genoemd. Zie bijvoorbeeld

www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fermat.html

Fermat dacht dat *alle* F_n priemgetallen zouden zijn. De Zwitserse wiskundige Leonard Euler (1707–1783) toonde aan dat dit niet het geval is, want F_5 blijkt deelbaar te zijn door 641. Inmiddels is voor een heleboel grotere F_n ook aangetoond dat het geen priemgetallen zijn. Op

www.prothsearch.net/fermat.html

vind je hierover meer informatie. Het is een open probleem of er na F_4 nog meer priemgetallen F_n bestaan.

Met Maple zien we heel snel dat Euler gelijk had: Het commando

$$(2^{(2^5)+1})/641;$$

laat al zien dat inderdaad 641 een deler is van F_5 . Maar we kunnen ook als volgt redeneren. Veronderstel dat F_5 wel een priemgetal zou zijn. Zoals we verderop zullen zien impliceert dit dat $\mathbb{Z}/F_5\mathbb{Z}$ een lichaam is. Dus elke $a \bmod F_5$ met $1 \leq a \leq F_5 - 1$ is een eenheid, en de groep $(\mathbb{Z}/F_5\mathbb{Z})^*$ bestaat uit precies $F_5 - 1$ elementen. Het antwoord op

$$F_5 := 2^{(2^5)+1} \quad : \quad 5 \&^{(F_5-1)} \bmod F_5 \quad ;$$

laat zien dat de orde van $5 \bmod F_5$ in $(\mathbb{Z}/F_5\mathbb{Z})^*$ niet een deler is van $F_5 - 1$. Maar dat is in tegenspraak met het feit dat in een eindige groep de orde van ieder element een deler is van het aantal elementen van de groep. Kennelijk geldt dus niet dat F_5 een priemgetal is.

Bij het vinden van een priemgetal p dat F_5 deelt is eenzelfde soort redenering heel nuttig. $2^{32} + 1$ is deelbaar door de priem p betekent namelijk precies dat $2^{32} \bmod p$ en $-1 \bmod p$ aan elkaar gelijk zijn. En daaruit volgt dat $(2 \bmod p)^{64} = 1 \bmod p$, dus de orde van $2 \bmod p$ is een deler van 64. Deze orde kan geen deler van 32 zijn, dus is hij precies 64. Dus 64 is een

deler van het aantal elementen in $(\mathbb{Z}/p\mathbb{Z})^*$, en omdat dit aantal $p - 1$ is (we gebruiken opnieuw het nog niet aangetoonde feit dat elke ring \mathbb{Z} modulo een priemgetal een lichaam is), volgt $p - 1 = 64m$ voor zekere m . Conclusie: $p = 64m + 1$. De kleinste priemgetallen van deze vorm zijn

$$p = 193, \quad 257, \quad 449, \quad 577, \quad 641$$

en daar hebben we de door Euler gevonden factor al. Overigens is met nog wat meer theorie zelfs in te zien dat een priemdeeler p van een Fermatgetal F_n de vorm $p = 2^{n+2}m + 1$ heeft.

2.5 machtsverheffen

In meerdere totnutoe gegeven voorbeelden hebben we van een element a uit een groep of een ring een vrij hoge macht a^n bepaald. In Maple wordt dit heel efficiënt gedaan door bijvoorbeeld de commando's `Powmod` en

```
a&^n mod m;
```

Uiteraard kunnen we voor het berekenen van a^n 'gewoon' n keer met a vermenigvuldigen. Een Mapleregel die dit doet ziet er uit als volgt.

```
k := 1 : ans := a : while k < n do ans := ans * a : k := k + 1 od : ans;
```

Een veel snellere manier bestaat eruit dat we zo vaak we maar kunnen een tussenresultaat kwadrateren. Dit leidt tot het volgende algoritme.

```
ans:=1: k:=n: pow:=a:
while k>1
do if (k mod 2)=1 then ans:=ans*pow fi:
   pow:=pow*pow: k:=floor(k/2)
od: ans*pow;
```

Om in te zien dat dit algoritme werkt, beschouwen we eerst een voorbeeld. Stel we willen a^{42} bepalen. Het paar variabelen (ans, pow) heeft dan achtereenvolgens de waarden

$$(1, a), \quad (1, a^2), \quad (a^2, a^4), \quad (a^2, a^8), \quad (a^{10}, a^{16}), \quad (a^{10}, a^{32})$$

en het eindantwoord wordt inderdaad a^{42} . Dit eindantwoord ontstaat als een product $a^{42} = a^2 \cdot a^8 \cdot a^{32}$. Anders gezegd, het algoritme schrijft 42 als een som van machten van 2, rekent ondertussen door herhaald kwadrateren

a^{2^i} uit en vermenigvuldigt de nodige a^{2^i} 's. In formulevorm: de exponent n wordt *binair* geschreven, dat wil zeggen in de vorm

$$n = e_m 2^m + e_{m-1} 2^{m-1} + \dots + e_1 2 + e_0,$$

waarin alle $e_i \in \{0, 1\}$ en $e_m = 1$. En dan is a^n gelijk aan het product van alle a^{2^j} over de waarden van j waarvoor $e_j = 1$. Om te zien of $e_0 = 1$ moet worden nagegaan of n oneven is. En de vraag of $e_1 = 1$ is dezelfde als de vraag of $n \operatorname{div} 2$ (dat is het getal dat binair geschreven eindigt op e_1) oneven is. Zo ook levert $(n \operatorname{div} 2) \operatorname{div} 2$ antwoord op de vraag wat e_2 is, enzovoort.

Omdat voor $n = e_m 2^m + e_{m-1} 2^{m-1} + \dots + e_1 2 + e_0$ met de e_i als boven geldt dat $2^m \leq n < 2^{m+1}$, geldt $m = \lceil \log_2(n) \rceil$ (het grootste gehele getal dat \leq de logaritme van n bij het grondtal 2 is). Het gegeven algoritme moet m keer een kwadraat uitrekenen, namelijk $a^2, a^4 = (a^2)^2$ tot en met $a^{2^m} = (a^{2^{m-1}})^2$. Verder moet het algoritme een aantal keren vermenigvuldigen, en wel precies één keer minder vaak dan er j 's zijn met $e_j = 1$. Het maximale aantal vermenigvuldigingen is dus m . Echter, 'gemiddeld' verwachten we in een binair geschreven getal evenveel enen als nullen, en dus verwachten we 'gemiddeld' ongeveer $m/2$ vermenigvuldigingen in ons algoritme.

In het speciale geval dat we willen machtsverheffen in een *groep* is er een manier waarmee in sommige gevallen het hiervoor gegeven algoritme behoorlijk kan worden versneld.

Bijvoorbeeld, als we van het element a in de groep G de macht a^{127} moeten bepalen, dan gebeurt dit door te schrijven

$$a^{127} = a^{64} \cdot a^{32} \cdot a^{16} \cdot a^8 \cdot a^4 \cdot a^2 \cdot a.$$

Hiervoor moet dan 6 keer gekwadraterd en ook 6 keer vermenigvuldigd worden, in totaal 12 bewerkingen. Echter, er geldt ook

$$a^{127} = a^{128} \cdot a^{-1}.$$

Dit laatste kan worden bepaald door 7 keer te kwadrateren, 1 keer te vermenigvuldigen en 1 keer een inverse uit te rekenen. Het is afhankelijk van de groep G of dit tot een versnelling leidt. In veel groepen is namelijk het bepalen van inverses een veel 'duurdere' operatie dan het vermenigvuldigen. Maar er zijn ook groepen waarin het bepalen van een inverse juist bijzonder eenvoudig is. In dat geval loont het de moeite om een aantal vermenigvuldigingen uit te wisselen tegen een inverse. Een manier om dit te implementeren is gegeven in onderstaand algoritme.

```

ans:=1: k:=n: pow:=a:
while k>1
  do if (k mod 4)=3
    then inv:=inverse(pow): ans:=ans*inv: k:=k+1
    else if (k mod 2)=1 then ans:=ans*pow fi
    fi: pow:=pow*pow: k:=k div 2
  od: ans*pow;

```

Ga zelf na dat dit inderdaad werkt.

2.5.1

Machtsverheffen wordt uiteraard gebruikt bij het vinden van de orde van een element uit een (eindige) groep. Een algoritme dat hier in de praktijk veel voor gebruikt wordt, is de *baby-step giant-step* methode van Dan Shanks (1917–1996), een Amerikaanse getaltheoreticus die onder meer als redacteur van het tijdschrift *Mathematics of Computation* veel aandacht vroeg voor algoritmische aspecten van de getaltheorie.

Stel we willen de orde van het element a in de eindige groep G bepalen. Het algoritme kiest een (niet te groot, niet te klein) positief geheel getal s . Vervolgens worden de ‘baby-steps’ in een tabel opgeslagen:

$$B := \{(e, 0), (a, 1), (a^2, 2), (a^3, 3), \dots, (a^{s-1}, s-1)\}.$$

Uiteraard stoppen we zodra ergens een paar (e, k) met $k > 0$ wordt aangetroffen; immers dan is $a^k = e$, en de eerste keer dat we dit tegenkomen zijn we er bovendien zeker van dat $k > 0$ *minimaal* is met deze eigenschap. Komen we geen paar (e, k) tegen, dan gaan we verder met de ‘giant-steps’: we bepalen achtereenvolgend

$$(a^s, 1), (a^{2s}, 2), \dots, (a^{s^2}, s)$$

en controleren steeds of a^{ks} al ergens als eerste coördinaat van een element van B is voorgekomen. Zodra dit gebeurt hebben we ℓ en k met $a^{ks} = a^\ell$, en dus $a^{ks-\ell} = e$. De echte orde van a is dan een deler van $ks - \ell$. Om die te kunnen vinden, hebben we nog nodig dat de delers van $ks - \ell$ gemakkelijk te berekenen zijn, en dat blijkt in het algemeen beslist niet het geval!

2.6 het euclidische algoritme

De belangrijkste voorbeelden van de in deze paragraaf te behandelen theorie zijn de ringen \mathbb{Z} en $K[X]$, met K een lichaam. Maar om deze niet apart te hoeven behandelen en ook omdat de algemene theorie in wezen niet moeilijker is dan de genoemde voorbeelden, gaan we wat abstracter van start.

Definitie 2.6.1 Een *Euclidische ring* is een commutatieve ring R plus een functie $f : R \rightarrow \mathbb{R}_{\geq 0} \cup \{-\infty\}$, die voldoet aan

E1 $f(0) = -\infty$ en $f(ab) = f(a) + f(b)$ voor alle $a, b \in R$.

E2 Bij elke $a, b \in R$ met $b \neq 0$ bestaan een $q \in R$ en een $r \in R$ zodat $a = qb + r$ en $f(r) < f(b)$.

E3 Bij elke niet lege deelverzameling $S \subset R$ bestaat er een $s \in S$ zodat $f(s)$ minimaal is; dat wil zeggen $f(s') \geq f(s)$ voor elke $s' \in S$.

De naam ‘Euclidisch’ komt af van de Griekse wiskundige en filosoof Euclides (ongeveer 430–360 voor Christus). Het concept ‘ring’ is pas zo’n 2250 jaar na Euclides in de wiskunde geïntroduceerd. Dat toch de beide woorden in deze definitie samenkomen, vindt z’n oorzaak in het feit dat de eigenschappen van gewone gehele getallen die Euclides bestudeerde, meer algemeen worden gevonden in dit type ringen. De eigenschap dat er bij gegeven $a, b \in R$ zekere elementen $q, r \in R$ moeten bestaan zodat geldt $a = qb + r$, wordt wel samengevat door te zeggen dat in R ‘deling met rest’ geldt.

Voorbeeld 2.6.2 De ring \mathbb{Z} is Euclidisch, met als functie f de functie gegeven door $f(n) = \log |n|$. Deling met rest zien we dan als volgt: kies bij gegeven a, b het getal $q \in \mathbb{Z}$ zo, dat $0 \leq \frac{a}{b} - q < 1$. Neem $r := |a - bq|$. Dan is $0 \leq r < |b|$ en dus $f(r) < f(b)$. Bij een deelverzameling $S \subset \mathbb{Z}$ bestaat altijd een $s \in S$ zodat $|s|$ minimaal is; voor deze s is dan ook $f(s)$ minimaal.

Het ‘delen met rest’ kan je ook door Maple laten doen:

```
irem(1012, 343, 'q'); q;
```

geeft je eerst de rest 326 en dan de q zodat geldt $1012 = q \cdot 343 + 326$.

Voorbeeld 2.6.3 De polynoomring $R = K[X]$ waarbij K een lichaam is, is een Euclidische ring. Als functie f nemen we hier de functie die aan een polynoom z’n *graad* toevoegt. Het delen met rest werkt dan volgens de bekende methode van ‘staartdelen’. Voorbeeld: we willen $aX^3 + bX^2 + cX + d$ delen door $uX + v$. Het vergelijken van de hoogste machten van X levert dat de deling $u^{-1}aX^2$ keer opgaat, en we hebben

$$aX^3 + bX^2 + cX + d = u^{-1}aX^2 \cdot (uX + v) + (b - u^{-1}av)X^2 + cX + d.$$

Deze rest heeft graad 2, maar door hem te schrijven als $u^{-1}(b - u^{-1}av)X \cdot (uX + v) + \dots$ is ook daarvan de graad eentje lager te krijgen. Dit is net zo lang te herhalen totdat de rest een graad heeft die kleiner is dan die van $uX + v$.

In het gegeven geval blijkt die rest de constante $-av^3u^{-3} + bv^2u^{-2} - cvu^{-1} + d$ te zijn.

En is er een collectie veeltermen gegeven, dan zit daar altijd een veelterm van minimale graad bij. Dus ook aan E3 is voldaan.

Het ‘delen met rest’ van polynomen is overigens ook in Maple geïmplementeerd. Zo levert de regel

$$\text{rem}(a \cdot X^3 + b \cdot X^2 + c \cdot X + d, u \cdot X + v, X, 'q');$$

achtereenvolgens de rest r en de q uit het hier besproken voorbeeld.

2.6.4

We gaan nu eigenschappen afleiden van elementen en van idealen in een Euclidische ring R met bijbehorende functie f . Om flauwigheden te voorkomen nemen we daarbij aan dat $1 \neq 0$ in R .

Er geldt $f(a) \geq 0$ voor elke $a \in R$ met $a \neq 0$.

Immers, als $f(a) < 0$ dan is $f(a) = -\infty$. Zou dan toch $a \neq 0$, dan bestond vanwege E2 toegepast op $1 = qa + r$ een $r \in R$ met $f(r) < f(a) = -\infty$. Dat kan natuurlijk niet.

Een Euclidische ring is een ring zonder nuldelers.

Zou namelijk gelden $ab = 0$ terwijl $a \neq 0$ en $b \neq 0$, dan was $0 \leq f(a) + f(b) = f(ab) = -\infty$ hetgeen absurd is.

Er geldt $f(1) = 0$.

Immers, $1 = 1 \cdot 1$ en dus $f(1) = f(1) + f(1)$. Omdat $f(1) \geq 0$, is dit alleen mogelijk als $f(1) = 0$.

Er geldt $f(u) = 0$ voor elke $u \in R^$.*

Immers, voor een eenheid u is $1 = u^{-1}u$ en dus $0 = f(1) = f(u) + f(u^{-1})$. Dit kan alleen als zowel $f(u)$ als $f(u^{-1})$ gelijk zijn aan 0.

Is I een ideaal in R dan bestaat er een $a \in R$ zodat $I = R \cdot a$.

Immers, als $I = (0)$ dan is dit duidelijk. En is $I \neq (0)$, pas dan eigenschap E3 toe op $I \setminus \{0\}$. Dat levert een $a \in I$ zodat $a \neq 0$ en $f(a) \leq f(b)$ voor elke $b \in I$ met $b \neq 0$. We beweren dat voor deze a geldt dat $I = Ra$. De inclusie $Ra \subset I$ volgt uit eigenschap I3 van idealen. Omgekeerd, is $c \in I$, dan bestaan er vanwege $a \neq 0$ elementen $q, r \in R$ met $c = qa + r$ en $f(r) < f(a)$. Omdat beide a en c elementen zijn van I , is $r = c - qa$ dat ook. Maar dan

geldt $r = 0$, want $f(a)$ was minimaal gekozen voor de elementen $\neq 0$ in I , en toch $f(r) < f(a)$. Dus volgt $c = qa$, oftewel $I \subset Ra$. Dit bewijst $I = Ra$.

2.6.5

Totnu toe hebben we onder meer gezien dat een Euclidische ring R een commutatieve ring is zonder nuldelers, en dat elk ideaal I er van de vorm Ra is. We gaan nu na in hoeverre hierin het element a is vastgelegd.

Bewering: $Ra = Rb$ geldt precies dan, als $a = ub$ voor een eenheid $u \in R^*$. Is namelijk u zo'n eenheid, dan is elke ra ook te schrijven als $(ru)b$ en omgekeerd elke rb als $ru^{-1}a$. Dus in dat geval is $Ra = Rb$. En is omgekeerd $Ra = Rb$, dan zit $a = 1 \cdot a$ in Rb , dus $a = rb$ voor zekere $r \in R$. Evenzo is $b = 1 \cdot b$ element van Ra , dus $b = sa$ voor zekere $s \in R$. is ofwel a ofwel b gelijk aan 0, dan volgt hieruit dat ook de andere 0 is. En dan is inderdaad $a = ub$, namelijk met $u = 1$. Zijn ze geen van beide 0, dan hebben we $a = rb = (rs)a$, dus $(1 - rs)a = 0$. Omdat R een ring is zonder nuldelers en bovendien $a \neq 0$, concluderen we hieruit dat $1 - rs = 0$. Dus r en s zijn eenheden, zoals we wilden aantonen.

Definitie 2.6.6 Voor elementen a_1, a_2, \dots, a_n in een Euclidische ring R heet elk element $a \in R$ met

$$Ra = (a_1, a_2, \dots, a_n)$$

een *grootste gemene deler* (ggd) van a_1, a_2, \dots, a_n .

Zoals we hebben aangetoond, bestaat deze grootste gemene deler van elk eindig aantal elementen. Bovendien is een ggd op vermenigvuldigen met een eenheid na welbepaald.

We gaan kort in op de naam 'ggd'. Is a de ggd van a_1, \dots, a_n in de ring R , dan volgt daaruit dat elke a_i in het ideaal Ra zit. Dit houdt in dat zo'n a_i te schrijven is als $r_i \cdot a$ voor zekere $r_i \in R$. Inderdaad is a dus een *deler* van elk van de a_i 's, kortom een gemeenschappelijke deler. En a is de grootste in die zin, dat een 'echt' veelvoud van a geen gemeenschappelijke deler van alle a_i 's meer kan zijn. Stel namelijk dat $b = ra$ ook een gemeenschappelijke deler is van alle a_i 's. Dan zou elke a_i te schrijven zijn als $a_i = s_i b$ voor zekere $s_i \in R$. En dus zou elke a_i in het ideaal Rb zitten. Dit levert

$$(a_1, \dots, a_n) \subset Rb \subset Ra = (a_1, \dots, a_n),$$

en dus $Ra = Rb$ oftewel b is a maal een eenheid, en dus is b geen 'echt' veelvoud van a .

Merk op dat wanneer a een ggd is van a_1, \dots, a_n dan betekent dit dat a te schrijven is als

$$a = r_1 a_1 + r_2 a_2 + \dots + r_n a_n$$

voor zekere $r_i \in R$. Immers, per definitie van een ggd is $Ra = Ra_1 + \dots + Ra_n$, en dus is elk veelvoud van a (en in het bijzonder a zelf als combinatie van de a_i 's te schrijven).

Het *Euclidische algoritme* is een algoritme waarmee een ggd a van een aantal elementen a_1, \dots, a_n in een Euclidische ring R wordt bepaald. Bovendien levert het algoritme elementen $r_i \in R$ zodat

$$a = r_1 a_1 + r_2 a_2 + \dots + r_n a_n.$$

Dit gaat als volgt.

- Stap 1. Definiëer voor $1 \leq i, j \leq n$ de elementen $r_{ij} \in R$ door $r_{i,i} = 1$ en $r_{ij} = 0$ als $i \neq j$.
- Stap 2. Als geldt $a_1 = a_2 = \dots = a_n = 0$, dan bestaat het ideaal in kwestie alleen uit 0, en als ggd nemen we dan $a = 0$. Dan is $a = r_{11}a_1 + r_{12}a_2 + \dots + r_{1n}a_n$. We gaan nu uit van het geval dat niet alle a_i gelijk aan 0 zijn. Dan mogen we zelfs aannemen (en dat doen we ook), dat voor elke i geldt $a_i \neq 0$. Immers, de eventuele nullen voegen niks toe aan het door de overige a_i 's voortgebrachte ideaal. Kies $x_i = a_i$ voor elke i .
- Stap 3. Kies een x_i zodat $0 \leq f(x_i) \leq f(x_j)$ voor elke j . Schrijf $x_j = q_j x_i + r_j$ met $q_j, r_j \in R$ en $f(r_j) < f(x_i)$ voor elke j met $j \neq i$ en $x_j \neq 0$.
- Stap 4. Vervang nu voor elke $j \neq i$ het element x_j door r_j . Neem bovendien voor alle $j \neq i$ als nieuwe r_{jk} het element (in de oude variabelen geschreven) $r_{jk} - q_j r_{ik}$ ($k = 1, \dots, n$).
- Stap 5. Als we zo een rijtje krijgen met op de k de plaats een element $x_k \neq 0$ en verder alleen maar nullen, dan nemen we deze x_k als ggd. Ten aanzien van de oorspronkelijke a_i 's geldt dan

$$x_k = r_{k1} a_1 + r_{k2} a_2 + \dots + r_{kn} a_n.$$

En hebben we nog geen rijtje bestaande uit één element $\neq 0$ en verder nullen, dan gaan we terug naar stap 3 van het algoritme.

We leggen nu uit waarom dit algoritme werkt. Aanvankelijk hebben we zekere elementen $x_i \in R$, namelijk $x_i = a_i$. Deze brengen het ideaal $I =$

(a_1, \dots, a_n) voort, en er geldt $x_i = r_{i1}a_1 + \dots + r_{in}a_n$ want zo zijn de r_{ij} 's gekozen.

Gedurende het algoritme proberen we dit zo te houden, terwijl we de x_i 's zoveel mogelijk gaan vervangen door elementen met een kleinere waarde $f(x_i)$. Eigenschap E3 van een Euclidische ring vertelt dat die waarden niet steeds kleiner en toch $\neq -\infty$ kunnen blijven, en dus zal het algoritme na eindig veel stappen klaar zijn.

In een typische stap van het algoritme wordt x_j vervangen door r_j , waarbij $r_j = -q_jx_i + x_j$. Om in te zien dat we hier niet op een ander ideaal overgaan, moeten we aantonen dat

$$Rx_1 + \dots + Rx_j + \dots + Rx_n = Rx_1 + \dots + Rr_j + \dots + Rx_n.$$

Maar dat is het geval omdat we x_j als combinatie van r_j en de overige x_i 's kunnen schrijven, en evenzo r_j als combinatie van alle x_i 's.

Verder vervangen we in zo'n zelfde stap ook alle r_{jk} door $r_{jk} - q_jr_{ik}$. Welnu, voor de stap gold

$$x_j = r_{j1}a_1 + r_{j2}a_2 + \dots + r_{jn}a_n.$$

We moeten inzien dat deze regel behouden blijft, en dat betekent, dat moet gelden

$$r_j = (r_{j1} - q_jr_{i1})a_1 + (r_{j2} - q_jr_{i2})a_2 + \dots + (r_{jn} - q_jr_{in})a_n.$$

Dit is inderdaad het geval: immers, $r_j = -q_jx_i + x_j$, en

$$q_jx_i = q_j(r_{i1}a_1 + r_{i2}a_2 + \dots + r_{in}a_n).$$

Trekken we de gegeven uitdrukkingen voor x_j en voor q_jx_i van elkaar af, dan staat er precies de gezochte formule voor r_j . Hiermee is aangetoond dat het algoritme werkt.

Voorbeeld 2.6.7 We nemen het standaardvoorbeeld van een Euclidische ring, namelijk de ring \mathbb{Z} . De ggd is in het algemeen bepaald op vermenigvuldigen met een eenheid in de ring na. In ons geval zijn ± 1 de enige eenheden, dus een ggd is hier op een teken na bepaald. Om de ggd echt vast te leggen, kunnen we dus bijvoorbeeld eisen dat deze *niet-negatief* moet zijn.

Bijvoorbeeld is $\text{ggd}(336, 126) = 42$. Met het Euclidische algoritme wordt dit als volgt bepaald: $\text{ggd}(336, 126) = \text{ggd}(126, 84) = \text{ggd}(84, 42) = \text{ggd}(42, 0) = 42$.

Hier zien we een opvallende eigenschap van het Euclidische algoritme: we bepalen de grootste van alle gemeenschappelijke delers van twee getallen,

zonder de lijst van alle gemeenschappelijke delers uit te rekenen. Er wordt zelfs helemaal geen poging gedaan om de getallen te ontbinden; en toch levert het algoritme zoals gezegd een (grootste en gemeenschappelijke) deler op.

Om getallen $a, b \in \mathbb{Z}$ te vinden zodat geldt $126a + 336b = 42$, gaat het algoritme als volgt te werk. Er geldt

$$\begin{aligned} 336 &= 1 \cdot 336 + 0 \cdot 126; \\ 126 &= 0 \cdot 336 + 1 \cdot 126. \end{aligned}$$

Het volgende getal dat het algoritme tegenkomt is 84, verkregen als $336 - 2 \cdot 126$, ofwel door in bovenstaande gelijkheden de onderste tweemaal van de bovenste af te trekken. Dat levert

$$84 = 1 \cdot 336 - 2 \cdot 126.$$

Door ook nog te gebruiken dat $42 = 126 - 84$ krijgen we tenslotte (2de min 3de gelijkheid)

$$42 = -1 \cdot 336 + 3 \cdot 126.$$

Dat is de gevraagde schrijfwijze.

Ook Maple kan dit voor je bepalen:

```
igcdex(336, 126, 'a', 'b'); a; b;
```

levert achtereenvolgens de ggd 42 en getallen a, b met $336a + 126b = 42$.

Voorbeeld 2.6.8 We nemen $R = \mathbb{Q}[X]$ en daarin de elementen $g = X^3 - 2X - 4$ en $h = X^3 - 8$. Dan is, als we het Euclidische algoritme volgen, $\text{ggd}(g, h) = \text{ggd}(X^3 - 8, -2X - 4 + 8) = -2X + 4$, want $X^3 - 8 = (-2X + 4)(-\frac{1}{2}X^2 - X - 2)$.

Omdat de ggd op een eenheid na bepaald is, is dan ook $X - 2$ een ggd van g en h . Deze $X - 2$ schrijven als combinatie van f en g gaat op precies dezelfde manier als bij gewone gehele getallen (aan het Euclidische algoritme zie je immers niet met welke Euclidische ring je werkt!):

$$\begin{aligned} X^3 - 2X - 4 &= 1 \cdot g + 0 \cdot h; \\ X^3 - 8 &= 0 \cdot g + 1 \cdot h; \\ -2X + 4 &= g - h. \end{aligned}$$

En dus is $X - 2 = \frac{1}{2}h - \frac{1}{2}g$.

Met Maple kom je als volgt tot dezelfde conclusie:

```
gcdex(X^3-2*X-4, X^3-8, X, 's', 't'); s; t;
```

Dit levert een ggd op plus in de variabelen s en t twee veeltermen zodat $sg + th$ gelijk is aan deze ggd.

Een toepassing van het Euclidische algoritme is het inverteren van elementen in $(R/aR)^*$, als a een element in een Euclidische ring R is. Dat gaat namelijk als volgt.

Stel dat $b \bmod aR$ een eenheid is. Dat betekent dat er een $c \bmod aR$ bestaat zodat $bc \bmod aR = 1 \bmod aR$. Anders gezegd, er is een $c \in R$ zodat $bc - 1 \in aR$, oftewel zodat $bc - 1 = ar$ voor zekere $r \in R$. Conclusie: is $b \bmod aR$ een eenheid, dan zijn er $c, r \in R$ te vinden zodat $cb - ra = 1$. Kortom, dan kunnen we 1 schrijven als combinatie van a en b . Maar dat wil precies zeggen dat $(a, b) = R$, oftewel dat 1 een grootste gemene deler is van a en b . Door het Euclidische algoritme op a en b toe te passen vinden we deze r en c , en dan hebben we ook de inverse $c \bmod aR$ van $b \bmod aR$ te pakken.

En omgekeerd, stel dat 1 een ggd van a en b is. Dan zijn er (met het Euclidische algoritme) $c, r \in R$ te vinden waarvoor geldt $cb - ra = 1$. Dit houdt dan in dat $cb - 1 = ra \in Ra$, en dus is $(c \bmod aR) \cdot (b \bmod aR) = 1 \bmod aR$. Conclusie: $b \bmod aR$ is inverteerbaar.

Zo zien we dat de inverteerbare elementen in R/aR precies alle $b \bmod aR$ zijn waarvoor geldt dat 1 een ggd van b en a is.

Voorbeeld 2.6.9 Neem $p \in \mathbb{Z}$ een priemgetal, dus $p > 1$ en p heeft geen delers behalve $\pm p$ en ± 1 . Als $a \in \mathbb{Z}$ en $1 \leq a < p$, dan hebben a en p geen delers > 1 gemeenschappelijk. Dus is $a \bmod p$ een eenheid in $\mathbb{Z}/p\mathbb{Z}$. Alle $a \bmod p$ met $1 \leq a < p$ zijn verschillend, dus dit levert $p - 1$ verschillende eenheden op in $\mathbb{Z}/p\mathbb{Z}$. We concluderen dat elk element ongelijk aan nul in $\mathbb{Z}/p\mathbb{Z}$ een eenheid is, en dus is $\mathbb{Z}/p\mathbb{Z}$ een *lichaam*.

Om het nog explicieter te maken, neem $p = 43$ en $a = 17$. Dan is $17 \bmod 43$ een eenheid in $\mathbb{Z}/43\mathbb{Z}$. De inverse vinden we met behulp van het Euclidische algoritme:

$$\begin{aligned} 43 &= 1 \cdot p + 0 \cdot a; \\ 17 &= 0 \cdot p + 1 \cdot a; \\ 9 &= 1 \cdot p - 2 \cdot a; \\ -1 &= -2 \cdot p + 5 \cdot a, \end{aligned}$$

en daaruit volgt dat $-5 \bmod 43 = 38 \bmod 43$ de inverse is van $17 \bmod 43$. In Maple is dat allemaal al voorgeprogrammeerd:

```
17&^(-1) mod 43;
```

levert hetzelfde antwoord.

2.6.10

Omdat voor p een priemgetal $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ een lichaam is, werkt het Euclidische algoritme ook in de ring $\mathbb{F}_p[X]$ bestaande uit de ‘polynomen modulo p ’. Dat kan tot een paar vergissingen leiden:

In $\mathbb{Q}[X]$ geldt $\text{ggd}(X^2, X + 3) = 1$, zoals ofwel direct ofwel met Maple eenvoudig is na te gaan. Als we dit antwoord modulo 3 willen zien, kan dat met het Maplecommando

$$\text{gcd}(X^2, X+3) \text{ mod } 3;$$

wat keurig het antwoord 1 oplevert. Echter, met $a \text{ mod } 3$ geschreven als \bar{a} ,

$$\textit{dit is niet de ggd van } X^2 \textit{ en } X + \bar{3} \textit{ in } \mathbb{F}_3[X]!!!$$

Immers, $X + \bar{3} = X$, dus die ggd is X .

Maple kan dit ook wel voor je uitrekenen. Daartoe moet het commando `Gcd` in plaats van `gcd` worden gebruikt:

$$\text{Gcd}(X^2, X+3) \text{ mod } 3;$$

levert *wel* het gezochte antwoord X .

2.7 de chinese reststelling

De chinese reststelling is een uitspraak over quotiëntringen die veel algemener geldt dan alleen maar voor quotiënten van Euclidische ringen. Toch zullen we ons hier tot die Euclidische ringen beperken, omdat de voor ons belangrijkste toepassingen van dit type zijn en ook omdat de uitspraak er in dit geval ietsje eenvoudiger uitziet. We beginnen met een beschrijving van de factorring R/Ra voor een Euclidische ring R en een $a \neq 0$ in R .

2.7.1

Stel dat $a \neq 0$ een element is in een Euclidische ring R . Elke $b \in R$ is dan te schrijven als $b = qa + r$, met $f(r) < f(a)$. Dit houdt in het bijzonder in, dat $b \text{ mod } Ra = r \text{ mod } Ra$. Conclusie: elk element van R/aR is te schrijven in de gedaante $r \text{ mod } aR$, waarin $f(r) < f(a)$.

Voorbeeld 2.7.2 In het geval van de ring \mathbb{Z} kunnen we bovendien nog afspreken dat we een rest r altijd ≥ 0 willen hebben. In dat geval betekent $r_1 \text{ mod } a\mathbb{Z} = r_2 \text{ mod } a\mathbb{Z}$ en bovendien $0 \leq r_1, r_2 < |a|$, dat $r_1 = r_2$. Kortom, zoals we al wisten, $\mathbb{Z}/a\mathbb{Z}$ bestaat voor $a \neq 0$ precies uit alle elementen $r \text{ mod } |a|$, met $0 \leq r < |a|$.

Voorbeeld 2.7.3 In een ring $K[X]$ waarbij K een lichaam is, voldoet de functie f in de definitie van ‘Euclidisch’ bovendien aan de eigenschap $f(g+h) \leq \max(f(g), f(h))$. Hieruit volgt dat ook in dit geval de elementen $r \bmod aK[X]$ met $f(r) < f(a)$ onderling verschillend zijn: is namelijk $r_1 - r_2$ een veelvoud van a , dan is ofwel $r_1 = r_2$, ofwel $\text{graad}(r_1 - r_2) \geq \text{graad}(a)$. Dit laatste is niet mogelijk, dus $r_1 = r_2$.

Met andere woorden: als $a \in K[X]$ een veelterm is van graad $n \geq 1$, dan bestaat $K[X]/aK[X]$ precies uit alle elementen $a_{n-1}X^{n-1} + \dots + a_1X + a_0 \bmod aK[X]$. Al deze elementen zijn onderling verschillend.

We hebben nog een begrip nodig alvorens we de chinese reststelling kunnen formuleren.

Definitie 2.7.4 Als R_1 en R_2 twee ringen zijn, dan wordt de *productring* $R_1 \times R_2$ als volgt gegeven.

De elementen van $R_1 \times R_2$ zijn alle paren (r_1, r_2) , waarin $r_1 \in R_1$ en $r_2 \in R_2$.

De nul van $R_1 \times R_2$ is het paar $(0, 0)$, waarin de eerste ‘0’ het nulelement van R_1 is en de tweede ‘0’ het nulelement van R_2 .

De ‘1’ in $R_1 \times R_2$ is op eenzelfde manier gegeven als het paar $(1, 1)$.

Optellen en vermenigvuldigen in $R_1 \times R_2$ gaat coördinaatsgewijs. Dus

$$(r_1, r_2) + (s_1, s_2) = (r_1 + s_1, r_2 + s_2)$$

en evenzo

$$(r_1, r_2) \cdot (s_1, s_2) = (r_1 \cdot s_1, r_2 \cdot s_2).$$

Je kan zelf nagaan dat de productring van twee ringen inderdaad weer een ring is.

Voorbeeld 2.7.5 Het product $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ bestaat uit vier elementen, namelijk de vier mogelijke paren $(a \bmod 2, b \bmod 2)$. Met zichzelf vermenigvuldigd levert elk van deze paren gewoon weer hetzelfde paar op.

Dit maakt deze productring tot een wezenlijk andere ring dan de ring $\mathbb{Z}/4\mathbb{Z}$. Weliswaar heeft ook die precies vier elementen, maar kwadrateren is er niet gelijk aan niksdoen.

2.7.6

Het is niet zo moeilijk om uit te vinden wat de eenheden in een productring $R_1 \times R_2$ zijn: het eenheidselement is er $(1, 1)$, dus een element (a, b) is een eenheid precies dan, als er een (c, d) is met $(ac, bd) = (1, 1)$. Dit houdt in dat $ac = 1$ in R_1 , dus a is een eenheid in R_1 , en evenzo $bd = 1$ in R_2 , dus ook is b een eenheid in R_2 . Kortom, de eenhedengroep $(R_1 \times R_2)^*$ bestaat precies uit alle paren (a, b) met $a \in R_1^*$ en $b \in R_2^*$.

2.7.7

Als R een Euclidische ring is en $a, b \in R$ elementen met $\text{ggd } 1$, dan geldt dat de ringen R/abR en $R/aR \times R/bR$ in wezen dezelfde ringen zijn. Dat houdt in dat elk element in de ene ring precies overeenkomt met een element in de andere (en omgekeerd), en dit overeenkomen voert sommen in sommen en producten in producten over. Dit heet de *chinese reststelling*.

Om te laten zien dat dit inderdaad zo is, moeten we uitleggen volgens welk voorschrift een element $r \bmod abR$ overeenkomt met een paar uit $R/aR \times R/bR$. Welnu, dat paar is $(r \bmod aR, r \bmod bR)$.

Dit kunnen we inderdaad zo voorschrijven: als namelijk zou gelden $r_1 \bmod abR = r_2 \bmod abR$, dan is $r_1 - r_2$ een veelvoud van ab , dus zeker een element van aR en ook van bR . Dus het paar $(r \bmod aR, r \bmod bR)$ hangt niet af van de keuze van een $s \in R$ met $s \bmod abR = r \bmod abR$.

Het is duidelijk dat dit voorschrift sommen in sommen, producten in producten, nulelement in nulelement en eenheidselement in eenheidselement overvoert.

We willen vervolgens zien, dat het niet voorkomt dat twee *verschillende* elementen uit R/abR kunnen overeenkomen met *hetzelfde* element uit $R/aR \times R/bR$. Stel daarvoor, dat zou gelden

$$(r \bmod aR, r \bmod bR) = (s \bmod aR, s \bmod bR).$$

Dit houdt in dat $r - s$ een veelvoud is van a en tevens een veelvoud is van b . Dus $r - s = r_1a$ en $r - s = r_2b$ voor zekere $r_1, r_2 \in R$. We gaan nu gebruiken dat 1 een ggd is van a en b . Dit betekent, dat er $x, y \in R$ zijn waarvoor geldt dat $xa + yb = 1$. Er volgt, dat

$$(r - s) = x(r - s)a + y(r - s)b = xr_2ab + yr_1ab = (xr_2 + yr_1)ab,$$

dus $r - s \in abR$. Anders gezegd, $r \bmod abR = s \bmod abR$.

Tenslotte willen we inzien dat *ieder* element van $R/aR \times R/bR$ overeenkomt met een element uit R/abR . Anders gezegd, gegeven een paar $(r_1 \bmod aR, r_2 \bmod bR)$, dan zoeken we één $r \in R$ waarvoor geldt dat zowel $r \bmod aR = r_1 \bmod aR$ als ook $r \bmod bR = r_2 \bmod bR$. Ook daarvoor gebruiken we dat a en b $\text{ggd } 1$ hebben. Kies dus weer $x, y \in R$ met $xa + yb = 1$. Dan blijkt $r = xar_2 + ybr_1$ te werken. Immers, $r \bmod aR = ybr_1 \bmod aR = r_1 \bmod aR$, omdat $ybr_1 - r_1 = -xar_1 \in aR$. En evenzo is $r \bmod bR = xar_2 \bmod bR = r_2 \bmod bR$.

Hiermee is de chinese reststelling bewezen.

Voorbeeld 2.7.8 Een typisch getallenvoorbeeld: omdat $\text{ggd}(9, 11) = 1$, moet er een getal n tussen 0 en 100 bestaan zodat $n \bmod 9 = 7 \bmod 9$ en

$n \bmod 11 = 6 \bmod 11$. Dat is namelijk het getal zodat $n \bmod 99$ in de chinese reststelling overeenkomt met $(7 \bmod 9, 6 \bmod 11)$.

Het bewijs van de chinese reststelling laat zien hoe je n vindt: schrijf eerst

$$1 = 5 \cdot 9 - 4 \cdot 11$$

en vervolgens

$$n' = 5 \cdot 9 \cdot 6 - 4 \cdot 11 \cdot 7 = -38.$$

Neem dan $n = 99 - 38 = 61$; ga zelf na dat deze n inderdaad aan de gestelde eisen voldoet.

Voorbeeld 2.7.9 Vanwege de chinese reststelling is de ring $\mathbb{Z}/91\mathbb{Z}$ in wezen dezelfde ring als $\mathbb{Z}/7\mathbb{Z} \times \mathbb{Z}/13\mathbb{Z}$. Een eenheid in die laatste ring is een paar (a, b) waarbij $a \in (\mathbb{Z}/7\mathbb{Z})^*$ en $b \in (\mathbb{Z}/13\mathbb{Z})^*$. Nu heeft $(\mathbb{Z}/13\mathbb{Z})^*$ precies 12 elementen, want 13 is een priemgetal en we weten dat dan elk element $\neq 0$ in $\mathbb{Z}/13\mathbb{Z}$ een eenheid is. In het bijzonder volgt dan dat $b^{12} = 1$ in $(\mathbb{Z}/13\mathbb{Z})^*$. Op dezelfde manier zien we in dat ook geldt $a^6 = 1$ in $(\mathbb{Z}/7\mathbb{Z})^*$. En dus $(a, b)^{12} = (1, 1)$.

Hieruit volgt dat in de overeenkomstige ring $\mathbb{Z}/91\mathbb{Z}$ ook zal gelden dat $x^{12} = 1$ voor elke $x \in (\mathbb{Z}/91\mathbb{Z})^*$. Dit is dus een groep bestaande uit $6 \cdot 12 = 72$ elementen, en elk element in die groep heeft een orde die een deler is van 12.

2.8 Opgaven

1. Probeer de in de Inleiding genoemde problemen met behulp van Maple op te lossen.
2. Bepaal alle eenheden in $\mathbb{Z}/n\mathbb{Z}$ voor $n = 9$, $n = 10$ en $n = 11$. Is een van deze ringen een lichaam?
3. Bepaal, voorzover ze bestaan, $(n \bmod 21)^{-1}$ in $\mathbb{Z}/21\mathbb{Z}$, voor $n = 10$, $n = 11$, $n = 12$, $n = 13$.
4. Bepaal $(3 \bmod 1000)^{100}$. Wat kan je op grond hiervan zeggen over de laatste drie cijfers van $3^{10000003}$?
5. In het proefschrift van Hans de Jong, die op 17 december 1999 aan de RuG bij professor Henk Broer promoveerde, staat de stelling dat $\begin{pmatrix} 1 \bmod 256 & 1 \bmod 256 \\ 1 \bmod 256 & 2 \bmod 256 \end{pmatrix}^{192} = 1$ in de ring $M_2(\mathbb{Z}/256\mathbb{Z})$. Controleer deze stelling. Kan je een dergelijke stelling bedenken door overal waar hier 256 staat 128 te schrijven? En andere machten van 2?
6. Neem $R = \mathbb{Z}/12\mathbb{Z}$. Probeer in $R[X]$ twee veeltermen van elk graad 3 te vinden waarvoor geldt dat hun product graad n heeft, voor elke $n \in \{-\infty, 0, 1, 2, 3, 4, 5, 6\}$.
7. We nemen $R = \mathbb{Z}/4\mathbb{Z}$, en we schrijven \bar{a} voor het element $a \bmod 4$ van R . Wat is R^* ? We gaan nu $R[X]^*$ bepalen en we zien dan in het bijzonder dat $R[X]^*$ niet gelijk is aan R^* . Neem (als voorbeeld) een $f \in R[X]$ van graad 3, met $\bar{1}$ als coëfficiënt van X^3 . Bereken verschillende producten fg in $R[X]$. Wat kan je zeggen over de graad van fg ?
Beantwoord dezelfde vraag als de coëfficiënt van X^3 in f gelijk is aan $\bar{2}$, en ook als deze $\bar{3}$ is.
Probeer te beredeneren dat $R[X]^*$ precies uit alle veeltermen van de gedaante $\bar{1} + \bar{2} \cdot h(X)$ bestaat, waarbij $h(X) \in R[X]$ willekeurig gekozen kan worden. (Wat is de inverse van zo'n veelterm?)
8. Ga na dat de factorring $\mathbb{Q}[X]/(X^2 - 5)$ in feite dezelfde ring is als de ring bestaande uit alle getallen van de vorm $a + b\sqrt{5}$ met $a, b \in \mathbb{Q}$. Is $\mathbb{Q}[X]/(X^2 - 5)$ een lichaam? Is $\mathbb{R}[X]/(X^2 - 5)$ een lichaam?
9. Neem $R = \mathbb{Z}[X]$ en $I = (4, 2X)$. Probeer de elementen van R/I zo eenvoudig mogelijk te schrijven. Laat zien dat R/I oneindig veel

onderling verschillende elementen heeft. Probeer te beredeneren dat $(R/I)^*$ uit slechts twee elementen bestaat.

10. Toon aan dat de ring $\mathbb{Z}[X]/(13, X^3 - 2)$ een lichaam is. Bepaal daartoe de orde in de eenhedengroep van een aantal elementen.
11. Een voorbeeld van een groep waarin ‘inverse nemen’ heel veel goedkoper is dan vermenigvuldigen, is de groep

$$G := \{a + b\sqrt{5} \mid a, b \in \mathbb{Z} \text{ en } a^2 - 5b^2 = 1\}.$$

Dit is een groep met als operatie erop de gewone vermenigvuldiging van reële getallen, en met $1 \in \mathbb{R}$ als eenheidselement.

Ga na dat de inverse van $a + b\sqrt{5} \in G$ gegeven wordt door $a - b\sqrt{5}$. Schrijf en vergelijk verschillende algoritmen voor machtsverheffen in G . Bepaal ook $(9 + 4\sqrt{5})^{60}$.

12. Bepaal een grootste gemene deler van $X^{17} - X$ en $X^7 + X^3$ in de ring $(\mathbb{Z}/17\mathbb{Z})[X]$.
13. Vind getallen $a, b, c \in \mathbb{Z}$ waarvoor geldt dat

$$450a + 216b + 729c = 9.$$

14. Wat is de inverse van $X^2 \bmod (X^7 - 2X + 1)$ in $\mathbb{Q}[X]/(X^7 - 2X + 1)$? Kan je in diezelfde ring ook $X^2 - 1 \bmod (X^7 - 2X + 1)$ inverteren?
15. Hoeveel elementen heeft de ring $(\mathbb{Z}/5\mathbb{Z})[X]/(X^3 + X + 1)$? Ga na dat deze ring een lichaam is.
16. Vind n met $0 < n < 1000$ zodat n rest 3 heeft bij deling door 7, tegelijkertijd rest 5 bij deling door 11 en rest 7 bij deling door 13.

2.9 een diagnostische toets

De volgende opgaven zijn bedoeld om je een indruk te geven in hoeverre je de basisstof uit het achter ons liggende hoofdstuk beheerst. Mochten deze opgaven nog veel problemen opleveren, dan is het dringend aan te raden nog wat extra tijd aan Hoofdstuk 2 te besteden alvorens verder te gaan met de in de komende hoofdstukken te behandelen toepassingen.

1. Vind in $(\mathbb{Z}/240\mathbb{Z})^*$ een element met orde 4.
2. Leg uit waarom er in $(\mathbb{Z}/240\mathbb{Z})^*$ geen element met orde 3 bestaat.
3. Beredeneer waarom in de matrixring $M_2(\mathbb{Z}/4\mathbb{Z})$ de matrix $\begin{pmatrix} \bar{1} & \bar{1} \\ \bar{1} & \bar{3} \end{pmatrix}$ geen eenheid is.
4. Hoeveel vermenigvuldigingen zijn nodig om bij een willekeurig element a in een ring de macht a^{100} uit te rekenen?
5. Zijn in \mathbb{Z} de idealen $3\mathbb{Z}$ en $24\mathbb{Z} + 15\mathbb{Z}$ gelijk? Waarom wel/niet?
6. Neem $I = (X^3, 8) \subset \mathbb{Z}[X]$. Is $X + 3 \bmod I$ in $\mathbb{Z}[X]/I$ inverteerbaar?
7. Uit hoeveel elementen bestaat R^* , als $R = (\mathbb{Z}/2\mathbb{Z})[X]/(X^2)$?
8. Bepaal $\text{ggd}(X^2 + X + 1, X^4 - 1)$ in $\mathbb{Q}[X]$.
9. Bepaal $\text{ggd}(X^2 + X + 1, X^4 - 1)$ in $(\mathbb{Z}/3\mathbb{Z})[X]$.
10. Welke veelterm in $\mathbb{Q}[X]$ van graad ≤ 7 heeft rest $X + 1$ bij deling door $X^2 + 1$ en rest 1 bij deling door $X^3 + X + 1$ en rest X^2 bij deling door $X^3 + 2$?

3 Cryptografie

In dit hoofdstuk behandelen we een toepassing van het rekenen modulo N , en dan met name van het machtsverheffen modulo N . Dit wordt onder meer gebruikt bij het beveiligen van gegevens op bankpassen, mobiele telefoons en vele soorten netwerken.

Het RSA-systeem waartoe we ons hoofdzakelijk beperken, is een beveiligingstechniek die bijvoorbeeld wordt toegepast in sommige WAP-telefoons (Nokia, ...). Ook gebruiken oudere versies van PGP ('pretty good privacy'; een voor het versleutelen van e-mail gebruikte techniek) het RSA-systeem.

Cryptografie houdt zich bezig met het beveiligen van informatie. Dat kan door de informatie om te zetten in een geheimschrift (encryptie), zodat als een ongewenst persoon de informatie in handen krijgt deze eerst moet worden ontcijferd (decryptie) om er iets begrijpelijks van te maken. En daarnaast is beveiliging mogelijk door alleen maar toegang te geven tot de informatie als je een of andere sleutel kent. Denk daarbij aan pincodes en aan de tokens/calculators die gebruikt worden om toegang tot allerlei netwerken te krijgen.

Een recent en erg leuk populair-wetenschappelijk boek over het onderwerp verscheen in augustus 1999: Simon Singh, *The code book. The science of secrecy from ancient Egypt to quantum cryptography* uitgegeven door Fourth Estate Ltd. Een Nederlandse vertaling met als titel *Code, de wedloop tussen makers en brekers van geheime codes en cijferschrift* verscheen bij De Arbeiderspers, eveneens in 1999. Het boek bevat onder meer een lijstje van 10 versleutelde berichten die aan de eerste ontcijferaar ervan een flink geldbedrag zouden opleveren. Ruim dertien maanden na het verschijnen van het lijstje, in oktober 2000, was deze hele 'Cipher Challenge' opgelost, zoals op

www.simonsingh.com/cipher.htm

valt na te lezen. Dat cryptografie nog steeds bijzonder actueel is, zie je bijvoorbeeld aan het werk van de Chinese wiskundige Xiaoyun Wang: zie

en.wikipedia.org/wiki/Xiaoyun_Wang

en ook het NRC-Handelsblad artikel dat begin 2006 over haar werk is geschreven: www.nrc.nl/scholieren/wiskunde/article224970.ece

3.1 eenheden modulo N

Het RSA-systeem is genoemd naar de drie bedenkers ervan: Ronald L. Rivest, Adi Shamir en Leonard M. Adleman introduceerden het in 1977, toen ze

alledrie werkzaam waren bij MIT. Een belangrijke eigenschap van RSA is, dat iedereen mag weten met welke ‘sleutel’ een boodschap gecodeerd is. Zelfs met deze informatie is het ontcijferen van een bericht nog steeds heel moeilijk, althans voorzover men weet. Het succes van het systeem is gebaseerd op het feit dat we voor een groot getal N geen efficiënte methode kennen om het aantal eenheden modulo N te bepalen.

Definitie 3.1.1 Voor een geheel getal $N \neq 0$ noteren we met $\phi(N)$ het aantal elementen van $(\mathbb{Z}/N\mathbb{Z})^*$.

Anders gezegd, $\phi(N)$ is het aantal eenheden modulo N . Zoals we bij de theorie over het Euclidische algoritme hebben gezien, is elk element van $\mathbb{Z}/N\mathbb{Z}$ te schrijven als $a \bmod N$, met $0 \leq a < |N|$. En zo’n $a \bmod N$ is een eenheid precies dan, als $\text{ggd}(a, N) = 1$. Dus $\phi(N)$ is ook gelijk aan het aantal getallen a met $0 \leq a < |N|$ waarvoor geldt $\text{ggd}(a, N) = 1$. De functie ϕ wordt de *Euler phi functie* genoemd, naar de in Zwitserland geboren wiskundige Leonhard Euler (1707-1783);

www-groups.dct.st-and.ac.uk/~history/Mathematicians/Euler.html

Voorbeeld 3.1.2 Er geldt $\phi(12) = 4$, want 1, 5, 7 en 11 zijn alle getallen tussen 0 en 11 die ggd 1 hebben met 12; dit zijn er precies 4. De groep $(\mathbb{Z}/12\mathbb{Z})^*$ bestaat dus kennelijk uit de elementen 1 mod 12 en 5 mod 12 en 7 mod 12 en 11 mod 12. Elk van deze elementen blijkt overigens z’n eigen inverse te zijn, anders gezegd

$$(a \bmod 12) \cdot (a \bmod 12) = 1 \bmod 12$$

voor elke a met $\text{ggd}(a, 12) = 1$.

Om bijvoorbeeld $\phi(10000)$ uit te rekenen is het voldoende om na te gaan, hoeveel getallen ≥ 0 en < 10000 er zijn die *geen* ggd 1 hebben met 10000. Dat is het geval voor alle even getallen (dat zijn er 5000) en voor alle getallen die eindigen op een 5 (daarvan zijn er 1000). Conclusie: $\phi(10000) = 10000 - (5000 + 1000) = 4000$.

Is $N = p^n$ een macht (met $n \geq 1$) van een priemgetal p , dan zijn de getallen onder N die *niet* ggd 1 hebben met N natuurlijk precies de veelvouden van p , dus

$$0, p, 2p, 3p, \dots, p^2, (p+1)p, \dots, (p^{n-1} - 1)p.$$

Dat zijn er in totaal p^{n-1} . We concluderen

$$\phi(p^n) = p^n - p^{n-1} = (p-1)p^{n-1}.$$

Met behulp van de Chinese Reststelling is er nog meer te zeggen over de Euler phi functie. Stel namelijk dat N en M grootste gemene deler 1 hebben. We weten, dat dan $\mathbb{Z}/NM\mathbb{Z}$ en $(\mathbb{Z}/N\mathbb{Z}) \times (\mathbb{Z}/M\mathbb{Z})$ in feite dezelfde ring is. In het bijzonder hebben de twee genoemde ringen dan evenveel eenheden, en dus geldt

$$\phi(NM) = \phi(N) \cdot \phi(M) \quad \text{als} \quad \text{ggd}(N, M) = 1.$$

De twee genoemde eigenschappen stellen ons in staat $\phi(N)$ te bepalen voor elke N die we kunnen schrijven als een product van priemgetallen. Zo is bijvoorbeeld $10000 = 2^4 \cdot 5^4$ en dus $\phi(10000) = \phi(2^4)\phi(5^4) = 2^3 \cdot 4 \cdot 5^3 = 4000$ zoals we al eerder zagen.

In Maple maakt de Euler phi functie deel uit van een pakket van functies die wat met getaltheorie te maken hebben, vandaar dat daar eerst dit pakket moet worden aangeroepen:

```
with(numtheory):
phi(10000);
```

levert het getal $\phi(10000)$.

3.2 worteltrekken modulo N .

Voor een element $a \bmod N$ in $\mathbb{Z}/N\mathbb{Z}$ schrijven we nu kortweg \bar{a} . Stel, dat $e > 0$ voldoet aan de eigenschap $\text{ggd}(e, \phi(N)) = 1$. Als \bar{a} een eenheid modulo N is, en iemand geeft ons $\bar{b} := \bar{a}^e$, dan is het mogelijk om daaruit \bar{a} terug te vinden. Dat klinkt misschien vanzelfsprekend, maar het is toch iets minder eenvoudig dan het wellicht op het eerste gezicht lijkt: je zou natuurlijk voor elke c met $1 \leq c < N$ kunnen proberen of $(c \bmod N)^e$ gelijk is aan \bar{b} . Behalve dat dit nogal tijdrovend zou kunnen zijn, is er ook nog het probleem dat *als* je een c te pakken hebt met de eigenschap $\bar{c}^e = \bar{b}$, hoe weet je dan dat $\bar{c} = \bar{a}$? Anders gezegd, we moeten nagaan dat het niet mogelijk is dat $\bar{c} \neq \bar{a}$ terwijl toch $\bar{c}^e = \bar{a}^e$. Welnu, omdat \bar{a} een eenheid is, volgt uit $\bar{c}^e = \bar{a}^e$ dat $(\bar{c} \cdot \bar{a}^{-1})^e = \bar{1}$. Hieruit volgt dat $\bar{c}\bar{a}^{-1}$ in de groep $(\mathbb{Z}/N\mathbb{Z})^*$ zit, en daarin een orde $d \geq 1$ heeft die een deler is van e . Maar deze orde d is ook, zoals in elke eindige groep geldt, een deler van het aantal elementen van de groep, dus hier van $\phi(N)$. Hieruit volgt dat d elke combinatie van de vorm $ne + m\phi(N)$ deelt. We weten dat ook $1 = \text{ggd}(e, \phi(N))$ als zo'n combinatie te schrijven is, en dus is $d \geq 1$ een deler van 1. Conclusie: de orde van $\bar{c}\bar{a}^{-1}$ is gelijk aan 1. Omdat het eenheidselement in een groep het enige element is met orde 1, volgt dan dat $\bar{c}\bar{a}^{-1} = \bar{1}$, en dus door links en rechts met \bar{a} te vermenigvuldigen, dat $\bar{c} = \bar{a}$.

Wat we hier hebben aangetoond, is dat we uit de e -de macht van een eenheid modulo N die eenheid kunnen terugvinden, mits $\text{ggd}(e, \phi(N)) = 1$. De methode van gewoon stuk voor stuk alle elementen modulo N proberen werkt, maar is bepaald niet efficiënt. We geven hier een veel betere methode, die gebruik maakt van het Euclidische algoritme. Gegeven is dus $\bar{b} = \bar{a}^e$. Schrijf $f := \phi(N)$. Weliswaar kennen we \bar{a} nog niet, maar wat we wel weten is dat $\bar{a}^f = \bar{1}$ omdat de orde van \bar{a} een deler moet zijn van het aantal elementen van de groep $(\mathbb{Z}/N\mathbb{Z})^*$. Voor elk geheel getal k is dan natuurlijk ook $\bar{a}^{kf} = (\bar{a}^f)^k = \bar{1}$, en dus (vermenigvuldig links en rechts met \bar{a}) volgt, dat $\bar{a}^{kf+1} = \bar{a}$. Dit gaan we voor een handig gekozen k gebruiken om \bar{a} te vinden. Er geldt namelijk $\text{ggd}(e, f) = 1$, en dus zijn met het Euclidische algoritme getallen k, d te vinden zodat $de - kf = 1$. Hebben we die, dan blijkt $\bar{a} = \bar{b}^d$. Immers,

$$\bar{b}^d = (\bar{a}^e)^d = \bar{a}^{kf+1} = \bar{a}.$$

Voorbeeld 3.2.1 Neem $N = 128$. Het element $3 \bmod 128$ is de 17de macht ergens van. In dit geval is $\phi(N) = 64$ en we vinden met het Euclidisch algoritme

$$4 \cdot 64 - 15 \cdot 17 = 1,$$

dus $(3 \bmod 128)^{-15} = 67 \bmod 128$ is het element van $\mathbb{Z}/128\mathbb{Z}$ waarvan de 17de macht gelijk is aan $3 \bmod 128$. Met Maple kunnen we dat ook vinden: ga na dat

```
with(numtheory):
igcdex(17, phi(128), 'd', 'm'): 3&^d mod 128;
```

hetzelfde (correcte) antwoord oplevert.

3.3 RSA

Het RSA-systeem werkt als volgt. Begin met een heel groot getal N en een e met $1 < e < \phi(N)$. Stel we hebben een bericht, dat we opgeslagen hebben als een groot getal $b < N$. Versleutel nu b als $(b \bmod N)^e$. Zoals we hebben gezien, kan hieruit b worden teruggevonden, tenminste als aan de volgende twee voorwaarden is voldaan:

$$\text{ggd}(b, N) = 1 \quad \text{en tevens} \quad \text{ggd}(e, \phi(N)) = 1.$$

Een eenvoudige manier om dit te gebruiken is, om als provider of superuser aan je users de combinatie (N, e) te verschaffen. Het getal $\phi(N)$ hou je echter geheim, evenals het getal d met de eigenschap $de + k\phi(N) = 1$. Ontvangt de provider dan een door een user versleuteld bericht, dan kan hij

het ontcijferen door het (modulo N) tot de macht d te verheffen. Omgekeerd kan hijzelf een bericht versleutelen door het tot de macht d te verheffen. De users ontcijferen dat dan weer door er vervolgens ook nog eens de e -de macht van te nemen.

Uiteraard kan je aan *verschillende* users ook onderling verschillende getallen e geven; op die manier verschaft je aan elke user een soort ‘digitale handtekening’: alleen hij of zij kent e , en de superuser kan dat verifiëren door na te gaan of de user ‘aan de andere kant van de lijn’ uit een aantal random gestuurde $b^d \bmod N$ de bijbehorende $b \bmod N$ kan terugvinden (door ze met behulp van z’n chip of computer modulo N weer tot de e -de macht te laten verheffen).

We gaan nu puntsgewijs in op de vraag, of een dergelijk systeem een goede beveiliging kan bieden.

- Zoals we al hebben gezien, werkt het ontcijferen door eerst $\phi(N)$ te kennen en daarmee vervolgens bij de gegeven e een bijbehorende d te vinden. De superuser moet er dus voor zorgen dat hij $\phi(N)$ kent. Dat lukt hem zeker als hij de ontbinding van N als product van priemgetallen kent, want dan hebben we een formule voor $\phi(N)$. Omgekeerd wil hij uiteraard niet, dat ook anderen $\phi(N)$ kunnen bepalen. Dus er moet zeker voor gezorgd worden dat het niet eenvoudig is om N in priemfactoren te ontbinden. Bijvoorbeeld voor N een product van een paar hoge machten van kleine priemgetallen nemen ($N = 2^{512}$ of $N = 10^{150}$) is erg onverstandig. Veel beter is het, om voor N een product van twee ongeveer even grote priemgetallen te nemen, dus $N = pq$ met zowel p als q priemgetallen van zo’n 80 cijfers. Dan is $\phi(N) = \phi(p)\phi(q) = (p-1)(q-1)$. Het uitrekenen van deze $\phi(N)$ is precies even moeilijk als het factoriseren van N . Immers, we hebben al gezien dat als we N gefactoriseerd hebben dan hebben we een eenvoudige formule voor $\phi(N)$. Omgekeerd, kennen we $\phi(N)$, dan blijken p en q de oplossingen te zijn van de tweedegraads vergelijking

$$X^2 - (N + 1 - \phi(N))X + N = 0,$$

zoals je door in te vullen gemakkelijk nagaat. En de oplossingen van zo’n vergelijking zijn snel te vinden.

- Op een naïeve manier zal een ongewenst persoon niet snel uit onze N van zo’n 150 cijfers het getal $\phi(N)$ kunnen bepalen: van een getal kleiner dan N nagaan of de ggd met N gelijk is aan 1 gaat weliswaar heel snel, maar zoiets 10^{150} maal te moeten uitrekenen is zelfs voor de huidige supercomputers een ondoenlijke klus.

Wel kunnen we hier opmerken, dat 150 cijfers op dit moment juist *binnen* het bereik ligt van wat anno 2000 met de beste factorisatiemethoden in priemfactoren kan worden ontbonden. Maar door onze cijfers bijvoorbeeld een factor 10^{20} groter te maken liggen we weer ver(?) buiten dat bereik...

- Een voorwaarde in het systeem is, dat alle berichten/getallen b die worden verstuurd de eigenschap $\text{ggd}(b, N) = 1$ hebben. Een gebruiker zou anders door bij elk bericht deze ggd te berekenen, tegen een factor van N kunnen aanlopen. Die kans is evenwel heel klein. Er zijn namelijk N mogelijke boodschappen, namelijk $0 \bmod N$ tot en met $N - 1 \bmod N$. Daarvan zijn er precies $\phi(N)$ eenheden. De kans dat een ‘zomaar’ gekozen boodschap dus geen eenheid is, is gelijk aan $(N - \phi(N))/N$. In ons geval met $N = pq$ is dit gelijk aan $(p + q - 1)/pq = 1/p + 1/q - 1/N$. Met als p en q priemgetallen groter dan 10^{74} is die kans nog minder dan $2 \cdot 10^{-74}$, en dat is volstrekt verwaarloosbaar.
- Je zou misschien verwachten dat als $\text{ggd}(b, N) \neq 1$, dan kan je uit $b^e \bmod N$ niet de boodschap b reconstrueren. In het algemeen kan dat inderdaad niet. Neem bijvoorbeeld $N = 4$ en $b = 2$ en $e = 3$. Dan is voldaan aan $\text{ggd}(e, \phi(N)) = 1$. Echter $b^e \bmod N = 0 \bmod N = 0^e \bmod N$ in dit geval, terwijl toch $b \neq 0$.

Echter, in het geval dat N een product is van twee *verschillende* priemgetallen p en q , dan doet dit probleem zich niet voor. Anders gezegd, dan geldt voor *iedere* b dat $b^{ed} \bmod N$ gelijk is aan $b \bmod N$, met e en d als gebruikelijk. Dit kunnen we begrijpen met behulp van de Chinese reststelling. Deze zegt in ons geval dat de ring $\mathbb{Z}/N\mathbb{Z}$ in feite gelijk is aan de productring $(\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$. Elke $b \bmod N$ correspondeert dan met een paar $(a \bmod p, c \bmod q)$. En de de -de macht daarvan is $(a^{de} \bmod p, c^{de} \bmod q)$. We moeten aantonen dat dit ook weer met b correspondeert, met andere woorden, dat

$$a^{de} \bmod p = a \bmod p \quad \text{en} \quad c^{de} \bmod q = c \bmod q.$$

Is $a \bmod p$ een eenheid in $\mathbb{Z}/p\mathbb{Z}$, dan is de bewering voor a juist, want $(\mathbb{Z}/p\mathbb{Z})^*$ heeft $p - 1$ elementen, dus $a^{p-1} \bmod p = 1 \bmod p$, en omdat $de - 1$ een veelvoud is van $p - 1$ volgt dan ook $a^{de-1} \bmod p = 1 \bmod p$ waaruit het gevraagde volgt. En is omgekeerd $a \bmod p$ geen eenheid, dan is $a \bmod p = 0 \bmod p$ omdat p een priemgetal is. En iedere macht van $a \bmod p$, dus ook de de -de, is dan opnieuw gelijk aan $a \bmod p$.

Precies hetzelfde argument werkt voor $c \bmod q$, dus daarmee is het bewijs voltooid.

- Uiteraard zorgt de superuser ervoor dat de getallen e die hij aan de gebruikers verstrekt de eigenschap $\text{ggd}(e, \phi(N)) = 1$ hebben.
- Je zou je heel misschien kunnen voorstellen, dat er mogelijk een snel algoritme is om uit $b^e \bmod N$ weer $b \bmod N$ te reconstrueren, zonder gebruik te maken van $\phi(N)$ en zo'n d . Tot dusver heeft niemand zekerheid of zoiets kan, maar de verwachting dat het onmogelijk is leidt tot een nog groter vertrouwen in de veiligheid van een systeem als RSA.
- Om een bruikbare implementatie van het hier beschreven RSA-systeem met N een product van twee priemgetallen te krijgen, is het wel nodig dat we aan priemgetallen van de gezochte grootte kunnen komen. Hiervoor zijn uitstekende algoritmen bekend. Met bijvoorbeeld het Maple commando

```
nextprime(1074);
```

krijg je het kleinste priemgetal van 75 cijfers. (In feite is het razendsnelle algoritme dat daarvoor wordt gebruikt ‘probabilistisch’, dat wil zeggen dat er een (heel kleine) kans bestaat dat de output toch niet priem is. Maar voor praktische doeleinden wordt deze kans verwaarloosd.)

3.4 Opgaven

1. Bereken $\phi(10^n)$, voor $n = 1, 2, \dots, 6, \dots$
2. Probeer voor elke n met $1 \leq n \leq 20$ te bepalen of de vergelijking $\phi(N) = n$ een oplossing $N \in \mathbb{Z}$ heeft.
3. Gebruik Maple om oplossingen van de vergelijking $\phi(2^n + 1) = 2^n$ te vinden. Wat valt je op?
4. Schrijf een Maple programmaatje dat het aantal positieve gehele getallen a onder een grens N telt met de eigenschap $\text{ggd}(a, N) = 1$. Dit programma berekent dus $\phi(N)$.
5. Van welk element uit $(\mathbb{Z}/10000\mathbb{Z})^*$ is 3 mod 10000 de 17de macht? Kan je dezelfde vraag beantwoorden wanneer we 10000 vervangen door 10^{10} ?
6. Ga na voor welke grootte van N Maple nog redelijk snel $\phi(N)$ lijkt te kunnen bepalen.
7. Wat is (in RSA) de d die hoort bij het decoderen van de sleutel (N, e) , waarin je N vindt met behulp van het Maple commando

```
N:=nextprime(10^12)*nextprime(10^12-10^10);
```

en $e = 1007$? Vind ook nog andere sleutels e bij deze N .

Probeer deze opgave ook met 10^{12} overal vervangen door een hogere macht van 10.

4 Irreducibiliteit

Een priemgetal is een geheel getal $\neq \pm 1$ dat niet te schrijven is als een product ab van gehele getallen die ook beide $\neq \pm 1$ zijn. Het is dus niet te *ontbinden*, oftewel het is irreducibel. Evenzo is de veelterm $X^3 + X + 1 \in \mathbb{F}_2[X]$ niet te schrijven als een product van twee veeltermen die beide niet constant zijn. We geven in dit hoofdstuk methoden waarmee kan worden nagegaan of zo'n polynoom danwel zo'n getal irreducibel is. Zo is bekend dat $2^{30402457} - 1$ (een getal bestaande uit 9152052 cijfers) een priemgetal is. In februari 2006 was dit het grootst bekende priemgetal, zie

<http://www.utm.edu/research/primes/largest.html>

voor veel informatie van dit soort.

4.1 polynomen over een eindig lichaam

Anders dan in het geval van gewone gehele getallen, kennen we voor polynomen over een eindig lichaam heel efficiënte algoritmen zowel om te testen of ze irreducibel zijn, als om ze te ontbinden. Een electro-technicus en wiskundige die hier belangrijke bijdragen heeft geleverd is de Amerikaan Elwyn Berlekamp. Hij legt zelf op zijn webpagina

www.math.berkeley.edu/~berlek/poly.html

uit hoe dit gedaan kan worden. We behandelen nu een paar details die met irreducibiliteit te maken hebben; in het volgende hoofdstuk komen we dan terug op het factoriseren.

4.1.1 monische polynomen

Een polynoom van de vorm $X^d + a_{d-1}X^{d-1} + \dots + a_0$ noemen we *monisch*. Een monisch polynoom is er dus eentje waarbij de coëfficiënt van de hoogste macht van X die voorkomt, 1 is. We zeggen ook wel 'een polynoom met *kopcoëfficiënt* 1'. Als we een monisch polynoom van graad d_1 vermenigvuldigen met een willekeurig polynoom van graad d_2 , dan heeft het product graad $d_1 + d_2$. In het bijzonder kan een monisch polynoom dus niet een nuldeeler zijn. Het product van twee monische polynomen is zelf ook weer monisch. Verder werkt het 'delen met rest' over een willekeurige ring, als we maar delen door een monisch polynoom. Met andere woorden: is f monisch, dan is elk polynoom g te schrijven als $g = qf + r$ waarbij $\text{graad}(r) < \text{graad}(f)$.

Uit dit 'delen met rest' volgt, net zoals we dat bij Euclidische ringen hebben gezien, dat als R een commutatieve ring is en $f \in R[X]$ een monisch

polynoom, dan is elk element van de factorring $R[X]/(f)$ te schrijven als $r \bmod f$ met $\text{graad}(r) < \text{graad}(f)$. Bovendien geldt voor zulke r, r' dat $r \bmod f \neq r' \bmod f$ zodra $r \neq r'$. In het geval dat R eindig is, dus uit slechts een eindig aantal elementen bestaat, kan je uit de hier gegeven beschrijving van $R[X]/(f)$ zien, dat ook dit weer een eindige ring is. Heeft R precies n elementen en is $d = \text{graad}(f)$, dan zijn er in zo'n polynoom $r \in R[X]$ van graad $< d$ precies d coëfficiënten uit R te kiezen, dus er zijn n^d zulke polynomen r . De ring $R[X]/(f)$ bestaat dan uit precies n^d elementen.

4.1.2 eindige lichamen

Een *eindig lichaam* is een lichaam met maar eindig veel elementen. In andere algebracursussen leer je, dat er alleen voor elke macht $q = p^n$ van een priemgetal p een eindig lichaam bestaat dat q elementen heeft. En voor zo'n q bestaat er dan precies één zo'n lichaam. We schrijven \mathbb{F}_q voor dit lichaam met q elementen.

Voorbeeld 4.1.3 Is p een priemgetal, dan is $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$.

We laten nu zien dat er ook een lichaam met p^2 elementen bestaat. Daarvoor beschouwen we de verzameling van alle monische polynomen $X^2 + aX + b$ van graad 2 in $\mathbb{F}_p[X]$. Daarvan zijn er p^2 . Een aantal hiervan is irreducibel. Immers, een niet irreducibele is te schrijven als $(X + \alpha)(X + \beta)$ voor zekere $\alpha, \beta \in \mathbb{F}_p$. Van polynomen van dit laatste soort zijn er p (namelijk de gevallen waarbij $\alpha = \beta$) plus $(p^2 - p)/2$ (namelijk eentje bij ieder ongeordend paar $\{\alpha, \beta\}$ uit \mathbb{F}_p). Er blijven dus nog $p^2 - p - (p^2 - p)/2 = (p^2 - p)/2$ irreducibele polynomen over, en voor elke p is dit een positief aantal. Gegeven zo'n monisch irreducibel polynoom f , is $\mathbb{F}_p[X]/(f)$ een lichaam. Immers, $\mathbb{F}_p[X]$ is een Euclidische ring, en omdat f irreducibel is, heeft elke $r \in \mathbb{F}_p[X]$ met $r \neq 0$ en $\text{graad}(r) < 2$ grootste gemene deler 1 met f , dus $r \bmod f$ is een eenheid. Zo krijgen we dus een lichaam \mathbb{F}_{p^2} . Als je goed naar deze constructie kijkt zie je, dat we op dezelfde manier uitgaande van een lichaam \mathbb{F}_q zo een lichaam \mathbb{F}_{q^2} kunnen maken.

Het rekenen in $\mathbb{F}_p[X]/(f)$ met behulp van Maple gaat zoals je dat in onderstaand voorbeeld ziet.

```
p:=5; f:=X^2+X+1;
Powmod(X-2,-1,f,X) mod p;
rem((X-2)*%,f,X) mod p;
```

4.1.4 irreducibele polynomen over eindige lichamen

Als f een monisch en irreducibel polynoom in $\mathbb{F}_q[X]$ is van graad d , dan is f een deler van $X^{q^d} - X$. Dit zien we als volgt. In het geval $f = X$ is

het duidelijk. In het geval $f \neq X$ geldt dat $\text{ggd}(X, f) = 1$, dus $X \bmod f$ is een eenheid in het lichaam $\mathbb{F}_q[X]/(f)$. Dit lichaam heeft q^d elementen, en de bijbehorende eenhedengroep heeft dan $q^d - 1$ elementen. Er volgt dat $(X \bmod f)^{q^d-1} = 1 \bmod f$. Dit betekent precies, dat $X^{q^d-1} \bmod f$ en $1 \bmod f$ hetzelfde element van $\mathbb{F}_q[X]/(f)$ zijn, oftewel dat f een deler is van $X^{q^d-1} - 1$. Er volgt dat f ook een deler is van $X \cdot (X^{q^d-1} - 1) = X^{q^d} - X$.

4.1.5 irreducibiliteit testen over een eindig lichaam

Stel nu dat f een polynoom over \mathbb{F}_q is van graad $e \geq 1$. Als f *niet* irreducibel is, dan heeft f een irreducibele factor g die we zo kiezen, dat z'n graad d zo klein mogelijk is. Dan is $1 \leq d \leq e/2$, want f kan niet het product zijn van minstens twee polynomen van graad $> e/2$. Er geldt dan dat g zowel f als $X^{q^d} - X$ deelt. In het bijzonder is dus $\text{ggd}(f, X^{q^d} - X) \neq 1$.

Hiermee hebben we een manier gevonden om de irreducibiliteit van f te testen. We geven die hier in Maple voor een $f \in \mathbb{F}_p[X]$:

```
readlib(powmod);
irred:=proc(f,p)
local e2, d, Xpd;
    e2:=iquo(degree(f,X),2); d:=1;
    Xpd:=powmod(X,p,f,X) mod p;
    while ((d<=e2)and((Gcd(f,Xpd-X)mod p)=1))
        do d:=d+1: Xpd:=powmod(Xpd,p,f,X) mod p od;
    if d>e2 then print('true') else print('false') fi
end;
irred(X^17-X^3+2,7);
irred(X^17-X^15+2,7);
```

Let erop, dat hier niet rechtstreeks $\text{ggd}(f, X^{q^d} - X)$ wordt berekend, maar eerst met behulp van snel machtsverheffen $X^{p^d} \bmod f$. Het algoritme levert bijvoorbeeld, dat $X^{17} - X^3 + 2$ reducibel, en $X^{17} - X^{15} + 2$ irreducibel is in $\mathbb{F}_7[X]$. Dit algoritme moet op z'n meest $\text{graad}(f)/2$ maal een p -de macht modulo f uitrekenen, en even zo vaak een ggd van twee polynomen van graad $\leq \text{graad}(f)$ bepalen. De hiervoor benodigde rekentijd is aanzienlijk kleiner dan wat nodig is om alle mogelijke (monische) veeltermen van graad $\leq \text{graad}(f)/2$ te beschouwen en stuk voor stuk er voor te beslissen of het een deler van f is.

Het hier gegeven algoritme bepaalt overigens tevens de kleinste graad d waarvoor f een irreducibele factor van graad d bevat.

4.2 ontbindbaarheid van getallen

Om na te gaan of een geheel getal $N > 1$ een priemgetal is, gaan we nu een paar methoden bespreken die op eenzelfde soort ideeën gebaseerd zijn als die we hiervoor bij polynomen al zagen.

We beginnen met een observatie die we ook bij veeltermen al hadden kunnen maken. Namelijk: de uitspraak ‘ N is een priemgetal’ is precies gelijkwaardig met de uitspraak ‘ $\mathbb{Z}/N\mathbb{Z}$ is een lichaam’. Immers, we hebben al meerdere keren opgemerkt (en gebruikt), dat als N een priemgetal is dan volgt daaruit dat $\mathbb{Z}/N\mathbb{Z}$ een lichaam is. Omgekeerd, als we zouden weten dat $\mathbb{Z}/N\mathbb{Z}$ een lichaam is, dan is het niet mogelijk dat N toch nog een echte ontbinding toelaat. Want zou $N = n \cdot m$ voor zekere n, m met $1 < n, m < N$, dan waren enerzijds $n \bmod N$ en $m \bmod N$ niet nul in $\mathbb{Z}/N\mathbb{Z}$, en dus waren het eenheden (want we veronderstellen dat $\mathbb{Z}/N\mathbb{Z}$ een lichaam is). En anderzijds geldt $(n \bmod N) \cdot (m \bmod N) = 0 \bmod N$. Dit is onmogelijk zoals je kan zien door beide kanten van deze gelijkheid met de inverse van $n \bmod N$ te vermenigvuldigen: je vindt dan dat $m \bmod N = 0 \bmod N$ en dat is niet waar. Hiermee is aangetoond dat inderdaad beide uitspraken gelijkwaardig zijn.

4.2.1 De Fermat-test

Is p een priemgetal en $a \in \mathbb{Z}$, dan geldt $a^p \bmod p = a \bmod p$. Anders geformuleerd, dan is $a^p - a$ deelbaar door p . Deze uitspraak zijn we al eerder bij de bespreking van het RSA-systeem tegengekomen. Het heet *de kleine stelling van Fermat* naar de ontdekker ervan. Nogmaals kort waarom de uitspraak geldt: als $a \bmod p$ een eenheid in \mathbb{F}_p is, dan is $(a \bmod p)^{p-1} = 1 \bmod p$ omdat $p - 1$ het aantal elementen is van de groep \mathbb{F}_p^* . Hieruit volgt de bewering voor dat geval, en als $a \bmod p$ geen eenheid is dat geldt $a \bmod p = 0 \bmod p$ waaruit het beweerde direct volgt.

Je kan deze kleine stelling van Fermat ook wel zonder groepen en zonder rekenen modulo p bewijzen, bijvoorbeeld met volledige inductie naar het getal a .

Uit de stelling volgt direct een algoritme waarmee je in veel gevallen kan zien dat een gegeven getal *niet* priem is: gegeven N , bepaal voor een aantal getallen a of $a^N \bmod N$ gelijk is aan $a \bmod N$. Is dat voor zekere a niet het geval dan weet je dat N geen priemgetal kan zijn. Maar als voor elke a die je probeert wel geldt dat $a^N \bmod N$ gelijk is aan $a \bmod N$, dan kan je daar geen conclusie over het al of niet priem zijn van N aan verbinden!

Voorbeeld 4.2.2 Neem $N = 91$ en $a = 3$, dan kunnen we met

$$3^{91} \bmod 91;$$

nog geen uitsluitstel geven op de vraag of 91 priem is. Echter, $2^{91} \bmod 91$ blijkt $37 \bmod 91$ te zijn, en daaruit concluderen we dat 91 te ontbinden moet zijn. Merk op dat deze test ons niet vertelt hoe zo'n ontbinding er dan uitziet!

Een getal N noemen we een a -pseudopriem als N niet priem is terwijl toch geldt dat $a^N \bmod N = a \bmod N$. Het zojuist gegeven voorbeeld laat dus ondermeer zien dat 91 een 3-pseudopriem is. Uit

$$\begin{aligned} 2^{341} &\bmod 341; \\ 3^{341} &\bmod 341; \end{aligned}$$

zie je dat 341 een 2-pseudopriem is.

Onze 'kleine Fermat-test' is geen echte priemtest vanwege het bestaan van de pseudopriemen. Nu zijn bijvoorbeeld sommige 2-pseudopriemen niet ook nog 3-pseudopriem, zoals we bij $N = 341$ zagen. Maar er zijn ook getallen die zowel 2-pseudopriem als 3-pseudopriem zijn; de kleinste $N > 1$ met die eigenschap is $N = 561$. Zo zou je verder kunnen gaan door vervolgens na te gaan of sommige van de pseudopriemen die je nog over hebt wellicht geen 4-pseudopriem zijn. Maar dat heeft geen zin: algemeen geldt, dat als N zowel a -pseudopriem als b -pseudopriem is, dan is N ook ab -pseudopriem. Immers, het gegeven zegt dat N niet priem is, en dat $a^N \bmod N = a \bmod N$ en $b^N \bmod N = b \bmod N$. Daaruit volgt dat

$$(ab)^N \bmod N = (a^N \bmod N)(b^N \bmod N) = (a \bmod N)(b \bmod N) = ab \bmod N,$$

dus inderdaad is N ook een ab -pseudopriem. Dit betekent dat we de kleine Fermat-test alleen maar hoeven uit te voeren voor getallen a die priem zijn; bovendien testen we alleen een eigenschap van $a \bmod N$ dus we mogen ons beperken tot $a < N$, en tenslotte geldt dat als we modulo N een priem a kunnen schrijven als product van kleinere priemen die al behandeld zijn, dan is de test voor a ook overbodig.

4.2.3 Carmichaelgetallen

Een getal $N > 2$ dat geen priemgetal is maar wel een a -pseudopriem voor elke a , heet een Carmichaelgetal. Deze getallen zijn genoemd naar de Amerikaanse wiskundige Robert D. Carmichael (1879–1967).

Er bestaan oneindig veel zulke Carmichaelgetallen, zoals in 1994 door de Amerikaanse wiskundigen William (Red) Alford, Andrew Granville en Carl Pomerance voor het eerst is aangetoond. De 'kleine Carmichaelgetallen' zijn onder andere door Richard Pinch uit Cambridge allemaal bepaald; vanaf zijn webpagina

kan je bijvoorbeeld alle 246683 Carmichaelgetallen kleiner dan 10^{16} downloaden.

Zoals we hebben gezien, kan je met de ‘kleine Fermat-test’ de Carmichaelgetallen niet van de priemgetallen onderscheiden. We geven nu een alternatieve beschrijving van Carmichaelgetallen. Hiermee is, als tenminste de factorisatie in priemfactoren van een gegeven getal bekend is, heel snel na te gaan of dat getal Carmichael is.

- Een Carmichaelgetal N is *kwadraatvrij*, dat wil zeggen dat voor geen enkel priemgetal p dat N deelt ook p^2 een deler is van N .

Zou namelijk gelden $N = p^2m$, dan geldt voor $a = p$ dat $a^N - a$ geen veelvoud is van N . Immers, a^{N-1} is natuurlijk door p deelbaar en dus is $a^{N-1} - 1$ dat niet. In $a(a^{N-1} - 1) = a^N - a$ zitten dus precies evenveel factoren p als in a , en dat is er maar 1. En dus kan $a^N - a$ niet N maal iets anders zijn, want in N komen minstens twee factoren p voor.

- Een Carmichaelgetal kan dus geschreven worden als $N = p_1 \cdot p_2 \cdot \dots \cdot p_t$ met $t > 1$, waarin de p_i onderling verschillende priemgetallen zijn. Omgekeerd, als we zo’n product hebben dan is dat een Carmichaelgetal indien geldt dat voor elke i het getal $p_i - 1$ een deler is van $N - 1$.

Dat dit inderdaad zo is hebben we eigenlijk al bij het bespreken van het RSA-systeem gezien: vanwege de Chinese reststelling is de ring $\mathbb{Z}/N\mathbb{Z}$ in feite dezelfde ring als het product

$$\mathbb{Z}/p_1\mathbb{Z} \times \mathbb{Z}/p_2\mathbb{Z} \times \dots \times \mathbb{Z}/p_t\mathbb{Z}.$$

En hebben we in laatstgenoemde ring een element $a = (\bar{a}_1, \bar{a}_2, \dots, \bar{a}_t)$, dan geldt dat a^N als coördinaten \bar{a}_i^N heeft. Omdat N te schrijven is als 1 plus een veelvoud van $p_i - 1$, is $\bar{a}_i^N = \bar{a}_i$ en dus $a^N = a$.

Voorbeeld 4.2.4 Met het genoemde criterium is bijvoorbeeld heel snel te zien dat 561 een Carmichaelgetal is: $561 = 3 \cdot 11 \cdot 17$ en 2, 10 en 16 zijn alle drie delers van 560.

- Zonder bewijs vermelden we, dat omgekeerd ook voor ieder Carmichaelgetal N geldt dat $p - 1$ een deler is van $N - 1$ voor elke priemdelers p van N . Om na te gaan dat dit zo is, wordt gebruikt dat er in $(\mathbb{Z}/p\mathbb{Z})^*$ een element bestaat waarvan de orde precies gelijk is aan $p - 1$. We zullen dit hier niet bewijzen.

4.2.5 Sterke pseudo-priemen

Het volgende argument blijkt te leiden tot een sterke verbetering van de kleine Fermat-test. Stel we hebben een priemgetal $p > 2$. Dat is natuurlijk een oneven getal, en dus is $p - 1$ een even getal. Door $p - 1$ zo vaak het maar mogelijk is te delen door 2, krijgen we zonder dat dit significant veel rekentijd kost een schrijfwijze

$$p - 1 = 2^t \cdot m,$$

waarin $t \geq 1$ en waarin m oneven is. Neem nu een klein (priem)getal $a < p$, en beschouw $\bar{a} := a \bmod p \in \mathbb{F}_p^*$ en $\bar{b} := \bar{a}^m$. Er geldt $\bar{b}^{2^t} = \bar{a}^{m \cdot 2^t} = \bar{1}$, dus de orde van \bar{b} in de groep \mathbb{F}_p^* is een (positieve) deler van 2^t . Deze delers kennen we: het zijn

$$1, 2, \dots, 2^{t-1}, 2^t.$$

Nu zijn er twee mogelijkheden.

- Ten eerste zou de orde van \bar{b} gelijk aan 1 kunnen zijn. In dit geval is $\bar{a}^m = \bar{b} = \bar{1}$.
- De overblijvende mogelijkheid is dat die orde gelijk is aan 2^s waarbij $1 \leq s \leq t$. We kijken nu naar $\bar{c} := \bar{b}^{2^{s-1}}$. Er geldt $\bar{c} \neq \bar{1}$, want anders was de orde van \bar{b} kleiner dan 2^s geweest. Verder

$$\bar{c}^2 = (\bar{b}^{2^{s-1}})^2 = \bar{b}^{2^s} = \bar{1}.$$

Hieruit volgt dat $(\bar{c} - \bar{1})(\bar{c} + \bar{1}) = \bar{0}$, en omdat \mathbb{F}_p een lichaam is en bovendien $\bar{c} - \bar{1} \neq \bar{0}$, impliceert dit $\bar{c} = \bar{-1}$.

Samengevat, zien we het volgende. Uitgaande van een klein (priem)getal $a < p$, bepalen we eerst $\bar{b} := \bar{a}^m$. Die kan gelijk zijn aan $\bar{1}$ of aan $\bar{-1}$. Is dit geen van beide het geval, dan komen we in de rij

$$\bar{b}^2, (\bar{b}^2)^2 = \bar{b}^4, (\bar{b}^4)^2 = \bar{b}^8, \dots, \bar{b}^{2^{t-1}}$$

een keer $-1 \bmod p$ tegen.

Om nu na te gaan of een oneven getal $N > 2$ een priemgetal is, kijken we of N voor een aantal kleine priemmen $a < N$ deze zelfde eigenschap heeft. Dit leidt tot het volgende algoritme.

```
misschienpriem:=proc(N,a)
  local m,t,min1,b,s;
  min1:=N-1;
  m:=min1;
```

```

s:=0;
t:=0;
while m mod 2 <>0 do
  m:=m/2;
  t:=t+1
od;
b:=a&^m mod N;
if b=1 or b=min1 then
  return(true)
else
  while (s<t-1) and (b<>min1) do
    b:=b&^2 mod N;
    s:=s+1
  od;
  if b=min1 then
    return(true)
  else
    return(false);
  fi;
fi;
end:

```

Deze test wordt ook wel de Miller-Rabin test genoemd, naar de Amerikaanse informatici Gary L. Miller en Michael O. Rabin die in 1976 en in 1980 dit algoritme beschreven. Als voor zekere a een getal N *niet* priem is, maar toch passeert N de hier beschreven test, dan wordt N een *sterke a -pseudopriem* genoemd.

Voorbeeld 4.2.6 Het getal $2047 = 89 \cdot 23$ is een sterke 2-pseudopriem. Hetzelfde geldt voor $3277 = 29 \cdot 113$. Maar door de test behalve voor $a = 2$ ook voor $a = 3$ te doen is zonder een factorisatie te kennen al in te zien dat geen van beide een priemgetal is.

Het getal 1373653 blijkt het kleinste getal te zijn dat zowel een sterke 2-pseudopriem als een sterke 3-pseudopriem is. Het is niet priem want 829 is een deler ervan.

Het getal 25326001 is al iets lastiger: dit blijkt het kleinste getal te zijn dat zowel voor $a = 2$ als voor $a = 3$ als voor $a = 5$ een sterke a -pseudopriem is. Door de test ook nog voor $a = 7$ uit te voeren is echter in te zien dat het hier toch niet om een priemgetal gaat.

Als een getal voor verschillende kleine waarden a de hier gegeven test doorstaat, ‘dan is de kans groot dat het een priemgetal is’. Deze wat vage uitspraak is veel preciezer te maken, maar we gaan er hier niet verder op in.

4.3 de Lucas-Lehmer priemtest

Er bestaan bijzonder efficiënte algoritmen waarmee voor heel grote getallen nog vrij snel kan worden nagegaan of het priemgetallen zijn. Dit onderwerp valt buiten het bestek van ons college. Hier beperken we ons tot getallen van een heel bijzondere soort, en daarvoor beschrijven we een heel snelle priemtest.

Schrijf $M_n = 2^n - 1$; dit heet het n -de Mersennegetal naar de Franse geestelijke Marin Mersenne (1588–1648), waarover je meer op

www.scruz.net/~luke/mersenne.htm

kan lezen. Een Mersennegetal dat een priemgetal is, heet een *Mersennepriem*.

Als n even is en $n > 2$, dan is M_n geen priemgetal. Immers, schrijf $n = 2n'$, dan

$$M_n = 2^{2n'} - 1 = (2^{n'} + 1)(2^{n'} - 1),$$

en omdat $n > 2$ volgt dat beide hier gegeven factoren > 1 zijn. Meer algemeen kan je inzien dat n zelfs een priemgetal moet zijn, wil M_n priem zijn. Dit blijkt uit de factorisatie

$$X^k - 1 = (X - 1)(X^{k-1} + \dots + X + 1)$$

voor elke $k \geq 2$: is $n = mk$, dan volgt door in deze veelterm $X = 2^m$ in te vullen dat

$$M_n = (2^m)^k - 1 = (2^m - 1)(2^{m(k-1)} + \dots + 2^m + 1),$$

en hierin zijn beide factoren > 1 als zowel m als k groter dan 1 zijn. Dus Mersennepriemen hebben de vorm $2^p - 1$ waarin p een priemgetal is.

Voorbeeld 4.3.1 Voor $p = 2, 3, 5$ en 7 krijgen we $M_2 = 3$ en $M_3 = 7$ en $M_5 = 31$ en $M_7 = 127$. Dit zijn inderdaad priemgetallen. Echter, $M_{11} = 2047 = 23 \cdot 89$ is geen priemgetal.

Er bestaat een heel snelle methode waarmee voor een gegeven Mersennegetal M_n met n oneven kan worden nagegaan of M_n priem is. Deze methode werd voor het eerst beschreven door de Franse wiskundige Edouard Lucas (1842–1891) en zo'n vijftig jaar later werd de methode verfijnd door de Amerikaanse getaltheoreticus Derrick H. Lehmer (1905–1991). De test heet dan ook *de Lucas-Lehmer test*, en werkt als volgt.

```
M:=2^n-1; a:=4;
for m from 1 to n-2 do a:=(a^2-2) mod M od;
if a=0 then print('priem') else print('niet priem') fi;
```

Het opmerkelijke hier is dat we van een getal M dat met n bits beschreven kan worden, primaliteit kunnen vaststellen door $n - 2$ vermenigvuldigingen (en eenvoudig aftrekken) van getallen modulo M . Merk verder op, dat het algoritme niet eerst test of n wel priem is.

We zullen nu bewijzen dat als het algoritme beweert dat M_n priem is, dan is dat ook inderdaad zo. De omkering hiervan, dus dat M_n niet priem is als het algoritme een $a \neq 0$ oplevert, is ook waar maar daarvoor geven we geen bewijs. We gebruiken in ons bewijs dat $n > 1$ oneven is.

Het algoritme begint met een getal $a_1 = 4$ en vervolgens berekent het $a_2 = a_1^2 - 2$, $a_3 = a_2^2 - 2$ enzovoort tot en met $a_{n-1} = a_{n-2}^2 - 2$. Al die getallen worden voortdurend modulo M_n genomen, en kennelijk moeten we laten zien dat als a_{n-1} nul is modulo M_n , oftewel als a_{n-1} een veelvoud is van M_n , dan volgt daaruit dat M_n priem is.

Om dit te bewijzen nemen we een willekeurig priemgetal p dat een deler is van M_n . Schrijf $\bar{a}_i := a_i \bmod p$. Onze aanname impliceert, dat $\bar{a}_{n-1} = \bar{0}$. Dit gaan we vertalen in een uitspraak over de eenhedengroep in een zekere ring.

Schrijf $\gamma := \bar{2}^{(n+1)/2} \in \mathbb{F}_p$. Dan is $\gamma^2 = \bar{2}$, want p deelt $M_n = 2^n - 1$, dus p deelt ook $2^{n+1} - 2$ en dat wil precies zeggen dat $\bar{2}^{n+1} = \bar{2}$. Omdat $\gamma^2 = \bar{2}^{n+1}$ volgt dus $\gamma^2 = \bar{2}$, zoals beweerd. De ring die we nu gaan gebruiken, is

$$R := \mathbb{F}_p[X]/(X^2 + \gamma X - \bar{1}).$$

Een element $f(X) \bmod (X^2 + \gamma X - \bar{1})$ in R zullen we kortweg schrijven als $\overline{f(X)}$. In R is het element \overline{X} een eenheid, want $\overline{X}(\overline{X} + \gamma) = \bar{1}$ omdat $X^2 + \gamma X - \bar{1}$ in het ideaal $(X^2 + \gamma X - \bar{1})$ zit. Als we de hier gebruikte gelijkheid $\overline{X}^2 + \gamma \overline{X} = \bar{1}$ links en rechts vermenigvuldigen met \overline{X}^{-1} , dan volgt

$$\overline{X} + \gamma = \overline{X}^{-1},$$

en dus $\overline{X} - \overline{X}^{-1} = -\gamma$. Door links en rechts het kwadraat te nemen en te gebruiken dat $\gamma^2 = \bar{2}$, volgt

$$\overline{X}^2 + \overline{X}^{-2} = \gamma^2 + \bar{2} = \bar{4} = \bar{a}_1,$$

kwadrateren we dit nogmaals waarbij we het ‘dubbele product’ weer naar de rechterkant van het gelijkteken brengen, dan staat er vervolgens

$$\overline{X}^{2^2} + \overline{X}^{-2^2} = \bar{a}_1^2 - \bar{2} = \bar{a}_2.$$

Zo doorgaande volgt dat eveneens

$$\overline{X}^{2^i} + \overline{X}^{-2^i} = \bar{a}_i$$

voor elke $i \geq 1$. Door dit voor $i = n - 1$ te gebruiken, zien we dat onze aanname dat a_{n-1} een veelvoud was van M_n nu oplevert, dat

$$\overline{X}^{2^{n-1}} + \overline{X}^{-2^{n-1}} = \overline{a_{n-1}} = \overline{0}.$$

Conclusie: $\overline{X}^{2^{n-1}} = -\overline{X}^{-2^{n-1}}$, en dus, door links en rechts met $\overline{X}^{2^{n-1}}$ te vermenigvuldigen,

$$\overline{X}^{2^n} = \overline{-1}.$$

Het kwadraat hiervan is $\overline{1}$, en dus zien we dat de orde van \overline{X} in de groep R^* een deler is van 2^{n+1} . Die orde kan geen deler zijn van 2^n , want $\overline{X}^{2^n} \neq \overline{1}$. En zo hebben we uiteindelijk de oorspronkelijke aanname ‘vertaald’ in de uitspraak, dat de orde van \overline{X} in R^* gelijk is aan 2^{n+1} .

We weten hieruit, dat R^* minstens 2^{n+1} elementen heeft, en dus heeft de hele ring R zelfs minstens $2^{n+1} + 1$ elementen. Anderzijds kunnen we vanwege het delen met rest in $\mathbb{F}_p[X]$ elk element uit $\mathbb{F}_p[X]/(X^2 + \gamma X - \overline{1})$ schrijven als $\overline{\alpha X + \beta}$ met $\alpha, \beta \in \mathbb{F}_p$. En daaraan zien we dat de ring R precies p^2 elementen heeft. Kennelijk geldt dus

$$p^2 \geq 2^{n+1} + 1 > 2^n - 1 = M_n.$$

De priemdelers p van M_n die we hier beschouwen, was echter willekeurig. We zien dus, dat *iedere* priemdelers van M_n groter is dan de wortel uit M_n . Was M_n niet priem, dan zou het product van de minstens twee priemfactoren uit M_n dus groter zijn dan M_n , en dat is natuurlijk onmogelijk. De conclusie hieruit is, dat M_n *wel* een priemgetal is, zoals we wilden bewijzen.

De Lucas-Lehmer test behoort overigens tot de standaardcommando's in Maple. Met

```
with(numtheory): mersenne(n);
```

krijg je de mededeling *false* als M_n niet priem is, en je krijgt M_n als deze *wel* een priemgetal is. En met

```
with(numtheory): mersenne([n]);
```

krijg je zelfs de n -de Mersennepriem (tenminste, voor kleine getallen $n \dots$).

4.4 Het AKS-algoritme

Op 6 augustus 2002 publiceerden de informatici Manindra Agarwal, Neeraj Kayal en Nitin Saxena van het Indian Institute of Technology in Kampur een preprint op hun website met als titel *PRIMES is in P*. Al snel verbreidde zich

het nieuws dat nu eindelijk bewezen was dat priemgetallen en samengestelde getallen in polynomiale tijd te onderscheiden zijn. Dezelfde maand berichtten diverse kranten en tijdschriften hierover. Op

<http://www.cse.iitk.ac.in/news/primality.html>

lees je hierover meer.

Het AKS-algoritme dat zij bedachten kreeg vooral zoveel aandacht, omdat met wat analytische getaltheorie te *bewijzen* was dat het primaliteit van elke gegeven input N kan nagaan in een tijd begrensd door een polynoom in $\log(N)$.

Het eerste en meest elementaire deel van het AKS-algoritme gaat na, of de input N te schrijven is als m^k voor gehele getallen $m, k \geq 2$. Maple zou dat bijvoorbeeld als volgt kunnen.

```
ma:=proc(N) local m,k:
  m:=N: for k from 2 to N while m>1 do
    m:=floor(evalf(N^(1/k))):
    if m^k=N then print(m,k); f:=1: fi;
  od; end proc:
```

Zodra $N^{1/k} < 2$, oftewel $N < 2^k$, dus $k > \log(N)/\log(2)$, zal dit algoritme stoppen. Dus het aantal stappen hier is zeker begrensd in termen van $\log(N)$.

Het minder elementaire deel moet vervolgens bepalen, of een getal dat *geen* echte macht is, een priemgetal is of niet. Daarvoor wordt een eigenschap van priemgetallen gebruikt:

Is p priem en $a \in \mathbb{F}_p$, dan geldt

$$(x + a)^p - x^p - a = 0 \quad \text{in } \mathbb{F}_p[x].$$

En omgekeerd, als deze gelijkheid geldt voor elke $a \bmod p$, dan is p priem.

Immers, is p priem, dan weten we in $\mathbb{F}_p[x]$ dat $(x + a)^p = x^p + a^p = x^p + a$. En omgekeerd, als zo'n gelijkheid geldt voor elke a , dan ook voor $a = 1$, en dus is volgens het binomium van Newton

$$\binom{p}{k} \equiv 0 \pmod{p}$$

voor $k = 1, \dots, p - 1$. Zou p geen priemgetal zijn, schrijf dan $p = q^e r$ met q priem en r geen veelvoud van q . Voor $k = q^e$ volgt dan

$$\binom{p}{q^e} = p(p-1) \cdots (p-q+1)/(q!)$$

en daarvan is het gemakkelijk na te gaan dat het deelbaar is door q^{e-1} maar niet door q^e , dus ook niet door p .

In het AKS-algoritme wordt niet nagegaan of bovenstaand polynoom 0 modulo N is, maar slechts of het modulo N een veelvoud is van een geschikt gekozen $x^r - 1$. Hoe je precies aan zo'n r komt, behandelen we hier niet; de genoemde website geeft er alle informatie over. Een kort Maple-programma dat zo'n test uitvoert (met een vastgekozen $r = 97$) is:

```
simplaks:=proc(N) local f,a:
  f:=0: for a from 1 to 50 while f=0 do
    f:=Rem(((Powmod(x+a,N,x^97-1,x)mod N-a-
      Powmod(x,N,x^97-1,x)mod N)mod N,x^97-1,x)mod N);
    a:=a+1 od: if a<51 then print("niet priem"); fi
end proc:
```

Om de theorie achter AKS goed te laten werken, volstaat het zeker niet om $r = 97$ te nemen. Echter, de polynomen waarmee hier gerekend wordt bestaan in het algemeen uit ongeveer r termen, en in elk daarvan komen getallen modulo N voor. Maar wie van getallen bestaande uit bijvoorbeeld miljoen cijfers wil nagaan of ze priem zijn, die mag natuurlijk ervan uitgaan dat voor het benodigde rekenwerk heel wat geheugenruimte wordt gebruikt.

4.5 Opgaven

1. De veelterm X^2+1 is over sommige lichamen \mathbb{F}_p irreducibel, over andere \mathbb{F}_p niet. Test dit met behulp van Maple voor een aantal kleine priemmen p , en probeer hierin een regelmaat te vinden. Je kan dit doen door het in dit hoofdstuk beschreven algoritme te implementeren; je kan ook het Maplecommando `Irreduc(X^2+1) mod p`; gebruiken.
Probeer dezelfde vraag te beantwoorden voor $X^2 + X + 1$.
2. Tel de reducibele en irreducibele polynomen van de vorm $X^2 + d$ in $\mathbb{F}_p[X]$. Kijk voor kleine p of je met Maple dezelfde antwoorden vindt.
3. Probeer (met hulp van Maple) alle monische irreducibele veeltermen van graad 3 over \mathbb{F}_2 en ook over \mathbb{F}_3 te vinden. Kan je op grond van de theorie beredeneren dat het gevonden aantal juist is?
4. Gebruik het in dit hoofdstuk beschreven algoritme om irreducibiliteit van veeltermen over \mathbb{F}_p te testen in een aantal concrete gevallen.
Vergelijk de snelheid met de Mapleregel `Berlekamp(f,X) mod p` voor bijvoorbeeld $f = X^2 + 1$ en een aantal vrij grote priemgetallen p .
5. Bepaal alle 13-pseudopriemen tussen 2 en 100.
6. Ga na dat elk van de volgende getallen een Carmichaelgetal is:
$$1105, 1729, 2465, 2821, 6601, 8911.$$
7. Probeer verschillende getallen a te vinden waarvoor geldt dat 703 een sterke a -pseudopriem is.
8. Controleer dat 3215031751 een sterke a -pseudopriem is voor elke $a \in \{2, 3, 5, 7\}$. In 1980 hebben de Amerikaanse wiskundigen John Selfridge en Sam Wagstaff (uiteeraard met behulp van een computer) aangetoond dat dit het enige getal < 25000000000 is met die eigenschap. Met andere woorden, voor elk ander getal onder die grens is door alleen maar de Miller-Rabin test voor 2, 3, 5 en 7 op het getal uit te voeren na te gaan, of het een priemgetal is.
9. Probeer met de Lucas-Lehmer test en Maple een aantal priemgetallen $p > 1000$ te vinden waarvoor M_p priem is.
10. Implementeer een naïeve versie van het AKS-algoritme, zoals hier beschreven. Ga na hoe vaak de ‘loop’ in het algoritme in de regel doorlopen wordt voor een samengesteld getal.

5 Factoriseren

In 1991 werd in Californië een lijst gegenereerd van grote getallen: de kleinste heeft 100 decimalen, de grootste 500. Het zijn geen van alle priemgetallen, en de uitdaging aan iedereen is, een van deze getallen te factoriseren. Deze getallen hebben de naam RSA- n meegekregen, waarbij n het aantal decimalen van getal RSA- n is. Via

www.rsasecurity.com/rsalabs/challenges/factoring/lists.html

vind je meer informatie over deze lijst plus de geldbedragen die met het factoriseren te verdienen zijn.

Getallen van meer dan 100 cijfers zijn lastig weer te geven op de pagina's van een dictaat. We breken ze daarom af en we gebruiken een klein lettertype. Wat voorbeelden:

RSA-100 = 15226050279225333605356183781326374297180681149613806886579084945801229 \\
63258952897654000350692006139
RSA-110 = 35794234179725868774991807832568455403003778024228226193532908190484670 \\
252364677411513516111204504060317568667
RSA-120 = 22701048129543736333425996094749366889587533646608478003817325824700916 \\
2675779735389791151574049166747880487470296548479
RSA-130 = 18070820886874048059516561644059055662781025167694013491701270214500566 \\
62540244048387341127590812303371781887966563182013214880557
RSA-140 = 21290246318258757547497882016271517497806703963277216278233383215381949 \\
984056495911366573853021918316783107387995317230889569230873441936 471
RSA-150 = 15508981247834844050960675437001186177065454583099543065546694577431263 \\
270346346595436333502757772902539145399678741402700350163177218684 0890795964683
RSA-155 = 10941738641570527421809707322040357612003732945449205990913842131476349 \\
984288934784717997257891267332497625752899781833797076537244027146 743531593354333897.

Zie

<http://www.npac.syr.edu/factoring/overview/RSAFCAList.txt>

voor een langere lijst. De bovenstaande getallen zijn inmiddels (eind 2000) gefactoriseerd op RSA-150 na. Voor RSA-140 en RSA-155 gebeurde dat in februari resp. augustus 1999. De uiteindelijke factorisatie in deze (en in meer!) gevallen is uitgevoerd op de supercomputer van SARA in Amsterdam door onder andere de wiskundige Herman te Riele. Via diens webpagina

www.cwi.nl/~herman

vind je hier meer over. Hoewel RSA-150 zelfs eind februari 2006 nog steeds niet gefactoriseerd is, lukte het een team van Duitse wiskundigen uit Bonn (Friedrich Bahr, Jens Franke en Thorsten Kleinjung) al twee keer, het record van Te Riele te verbreken: op 1 april 2003 gaven ze de factorisatie van RSA-160, en op 6 mei 2005 die van RSA-200. Op

www.rsasecurity.com/rsalabs/node.asp?id=2097

houdt men de verdere ontwikkelingen omtrent het factoriseren van de RSA-lijst bij.

Ter controle: de factorisatie van RSA-155 is

$$\begin{aligned} &102639592829741105772054196573991675900716567808038066803341933521790711307779 \\ &\quad \times \\ &106603488380168454820927220360012878679207958575989291522270608237193062808643. \end{aligned}$$

We gaan in dit hoofdstuk in op een paar ideeën op het gebied van factorisatie van getallen en van veeltermen.

5.1 veeltermen over een eindig lichaam

Het factoriseren van een (monische) veelterm over een lichaam bestaat uit twee gedeeltes. Aan elk hiervan besteden we een paragraaf; het eerste deel blijkt over een willekeurig lichaam te werken en in het tweede deel gaan we echt gebruiken dat we een polynoom over een *eindig* lichaam beschouwen.

5.1.1 meervoudige factoren

Gegeven $f = a_n X^n + \dots + a_1 X + a_0$, definiëren we met de gebruikelijke formule de *afgeleide* f' van f als

$$f' = na_n X^{n-1} + \dots + 2a_2 X + a_1.$$

Merk daarbij op, dat f hier een veelterm met coëfficiënten in een *willekeurige* (commutatieve) ring R mag zijn. Een coëfficiënt ma_m zoals in de afgeleide voorkomt, met $m \in \mathbb{Z}$, $m > 0$ en $a_m \in R$, interpreteren we daarbij als

$$\underbrace{a_m + a_m + \dots + a_m}_{m \times}.$$

Evenals in het bekende klassieke geval, voldoet de zo gegeven afgeleide aan de productregel $(fg)' = f'g + fg'$ zoals bijvoorbeeld kan worden nagegaan door in deze uitdrukking links en rechts de coëfficiënten van elke X^m te vergelijken.

Uit deze productregel volgt dat de afgeleide van een macht f^k van f gelijk is aan $kf^{k-1} \cdot f'$. Voor $k = 2$ is dat precies de productregel toegepast op $f \cdot f$, en het algemene geval volgt met volledige inductie naar k .

Stel nu dat K een lichaam is, en dat $f \in K[X]$ een *meervoudige factor* bevat. Dit wil zeggen dat f te schrijven is als $f = g^k \cdot h$, waarbij $k \geq 2$. De afgeleide van f is dan

$$f' = kg^{k-1}g'h + g^kh' = g^{k-1} \cdot (kg'h + gh').$$

In het bijzonder zien we dat dit een veelvoud is van g^{k-1} . De conclusie is, dat f en f' een factor g^{k-1} gemeenschappelijk hebben. In het bijzonder is dus $e := \text{ggd}(f, f')$ een factor van f met graad ≥ 1 (want g^{k-1} is een deler van e) en ook $\text{graad}(e) \leq \text{graad}(f) - 1$ (want e deelt de afgeleide van f en die heeft een graad die $\leq \text{graad}(f) - 1$ is). Er is één uitzondering waarin dit argument niet klopt: Het blijkt namelijk mogelijk te zijn, dat $f' = 0$, het nulpolynoom. Op dit speciale geval komen we nog terug.

Samenvattend zien we, dat als f een meervoudige factor heeft, en $f' \neq 0$, dan krijgen we een ontbinding van f waarin $\text{ggd}(f, f')$ een van de factoren is.

Omgekeerd, als $\text{ggd}(f, f') = 1$ dan kan vanwege bovenstaand argument f geen meervoudige factor hebben. In het bijzonder geldt dus dat als we schrijven

$$f = \text{ggd}(f, f') \cdot h,$$

dan heeft h geen meervoudige factoren.

Voorbeeld 5.1.2 In de volgende Maple-commando's zie je hoe de veelterm $x^5 + x^4 + x^3 \in \mathbb{F}_3[x]$ wordt gefactoriseerd.

```
f:=x^5+x^4+x^3; df:=diff(f,x); g:=Gcd(f,df) mod 3;
h:=Quo(f,g,x) mod 3;
dg:=diff(g,x); g2:=Gcd(g,dg) mod 3; h2:=Quo(g,g2,x) mod 3;
g2;
```

Dit levert de factorisatie $f = (x + \bar{2})^2 x^3$ op.

Merk op dat we hierboven een voorbeeld zien van iets merkwaardigs dat over bijvoorbeeld \mathbb{Q} of \mathbb{R} niet voorkomt: het polynoom $g2 = x^3$ dat we hier tegenkomen heeft in $\mathbb{F}_3[x]$ als afgeleide het nulpolynoom. Kennelijk is het dus over \mathbb{F}_3 mogelijk dat een “niet-constante functie” toch nul als afgeleide heeft! Dit gebeurt meer algemeen in $\mathbb{F}_p[x]$ precies voor alle polynomen waarin uitsluitend termen ax^n voorkomen met n een veelvoud van p . In zo'n geval blijkt het polynoom overigens heel eenvoudig te factoriseren:

- In $\mathbb{F}_p[x]$ is $(f + g)^p = f^p + g^p$.
- In $\mathbb{F}_p[x]$ geldt

$$a_n x^{pn} + a_{n-1} x^{(n-1)p} + \dots + a_1 x^p + a_0 = (a_n x^n + a_{n-1} x^{(n-1)} + \dots + a_1 x + a_0)^p.$$

De tweede formule volgt uit de eerste door op te merken dat voor $a \in \mathbb{F}_p$ geldt $a^p = a$, en dus $(ax)^p = ax^p$. De eerste formule kennen we eigenlijk allang voor heel kleine waarden van p : bijvoorbeeld geldt $(f + g)^2 = f^2 + 2fg + g^2$, en in \mathbb{F}_2 geldt $\bar{2} = \bar{0}$, dus ook $2fg = \bar{0}$ en daarom $(f + g)^2 = f^2 + g^2$. Evenzo hebben we voor $p = 3$ dat $(f + g)^3 = f^3 + 3f^2g + 3fg^2 + g^3$, en in \mathbb{F}_3 is dat gelijk aan $f^3 + g^3$. Meer algemeen is er voor elke p een formule voor $(f + g)^p$ in termen van de producten $f^n g^{p-n}$, en het blijkt dat daarin als coëfficiënten alleen getallen voorkomen die deelbaar zijn door het priemgetal p , behalve voor $n = 0$ en voor $n = p$. Hieruit volgt de formule.

5.1.3 Berlekamp's algoritme

In de vorige paragraaf hebben we gezien, dat het relatief eenvoudig is om meervoudige factoren uit een polynoom f over een lichaam te halen. We houden ons nu bezig met een monische $f \in \mathbb{F}_p[X]$ waarvan we veronderstellen dat f niet constant is, niet irreducibel is, en geen meervoudige factoren heeft. Deze f is dan te schrijven als

$$f = f_1 \cdot \dots \cdot f_t,$$

waarin de f_i monisch en irreducibel en onderling verschillend zijn.

Deze voorwaarde op de f_i impliceert, dat de factorring $\mathbb{F}_p[X]/(f)$ in feite dezelfde ring is als het product

$$\mathbb{F}_p[X]/(f_1) \times \dots \times \mathbb{F}_p[X]/(f_t).$$

Neem elementen $a_1, \dots, a_t \in \mathbb{F}_p$ die niet allemaal gelijk zijn, en bekijk in de gegeven productring het element

$$a := (\bar{a}_1, \dots, \bar{a}_t)$$

met $\bar{a}_i = a_i \bmod f_i$. Dit element correspondeert met een zekere $g \bmod f$ uit $\mathbb{F}_p[X]/(f)$ waarbij $g \in \mathbb{F}_p[X]$ een veelterm van graad kleiner dan $\text{graad}(f)$ is. We geven een paar eigenschappen van deze $\bar{g} = g \bmod f$.

Ten eerste geldt $\bar{g}^p = \bar{g}$, want voor de corresponderende $a = (\bar{a}_1, \dots, \bar{a}_t)$ geldt eveneens $a^p = a$.

Verder is g geen constant polynoom. Immers, in de Chinese reststelling die we hier toepassen correspondeert $h \bmod f$ met het rijtje $(h \bmod f_1, \dots, h \bmod f_t)$. Is h hier een constante, dan zijn niet alle $a_i - h$ gelijk aan nul want niet alle a_i zijn gelijk. En zo'n $a_i - h \neq 0$ is geen veelvoud van f_i , dus $a_i \bmod f_i \neq h \bmod f_i$ en dus $a \neq (h \bmod f_1, \dots, h \bmod f_t)$.

Het polynoom $g - a_1$ is een veelvoud van f_1 . Immers, $g - a_1$ correspondeert in $\mathbb{F}_p[X]/(f_1) \times \dots \times \mathbb{F}_p[X]/(f_t)$ met het rijtje

$$(g - a_1) \bmod f_1, \dots, (g - a_1) \bmod f_t,$$

wat gelijk moet zijn aan $(a_1 - a_1, a_2 - a_1, \dots, a_t - a_1)$. Dus $(g - a_1) \bmod f_1 = 0 \bmod f_1$, hetgeen precies wil zeggen dat $g - a_1$ een veelvoud is van f_1 .

De conclusie is dat $\text{ggd}(g - a_1, f)$ een echte deler is van f . Immers, deze ggd is $\neq 0$ en het is een veelvoud van f_1 . Verder heeft de ggd een graad die kleiner is dan die van f , want $g - a_1$ heeft een positieve graad die kleiner is dan de graad van f .

Omgekeerd correspondeert elke $\bar{g} = g \bmod f$ waarbij g een positieve graad heeft die kleiner is dan de graad van f , en waarbij bovendien geldt $\bar{g}^p = \bar{g}$, met zo'n rijtje $= (\bar{a}_1, \dots, \bar{a}_t)$ voor zekere $a_i \in \mathbb{F}_p$ die niet allemaal gelijk zijn. Immers, \bar{g} levert zeker zo'n rijtje met alle $\bar{a}_i \in \mathbb{F}_p[X]/(f_i)$. Omdat f_i irreducibel is, is deze ring een lichaam. In dit lichaam voldoet \bar{a}_i aan de vergelijking $X^p - X = 0$, en alle $a \bmod f_i$ met $a \in \mathbb{F}_p$ voldoen hier ook aan. Echter, zo'n vergelijking van graad p over een lichaam kan slechts p oplossingen hebben, en dus moet \bar{a}_i wel gelijk zijn aan zo'n $a \bmod f_i$ met $a \in \mathbb{F}_p$. Dat het rijtje a_i 's niet constant is volgt (zoals we al eerder gezien hebben) uit de voorwaarde op de graad van g .

Bovenstaande redenering leidt tot het volgende algoritme om f te factoriseren. Deze techniek werd voor het eerst beschreven door Elwyn Berlekamp in 1967 en wordt daarom het *Berlekamp algoritme* genoemd.

- Vind een 'niet-constante' $g \bmod f$ die voldoet aan $(g \bmod f)^p = (g \bmod f)$.
- Probeer voor de elementen $\bar{m} = \bar{0}, \bar{1}, \dots, \overline{p-1}$ of $\text{ggd}(g - \bar{m}, f) \neq 1$ is. Is dit het geval, dan heb je een echte deler van f gevonden.

De reden dat dit ook echt *snel* werkt, is dat het vinden van zo'n g heel effectief gedaan kan worden. Immers, schrijf

$$g = b_0 + b_1X + \dots + b_dX^d$$

voor de gevraagde g , waarin $d = \text{graad}(f) - 1$. Zoals we al eerder noemden, geldt dan

$$g^p = b_0 + b_1X^p + \dots b_dX^{dp}.$$

Om dus $g^p \bmod f$ uit te rekenen, hebben we alleen maar $X^p \bmod f$ en daarvan het kwadraat tot en met de d -de macht nodig. De zo verkregen g^p moet gelijk zijn aan g , en deze eis levert een stelsel lineaire vergelijkingen in de variabelen b_0 tot en met b_d . Er bestaan heel goede algoritmen waarmee zo'n stelsel kan worden opgelost.

Tenslotte dient voor de gevonden g een aantal grootste gemene delers te worden berekend. Als p vrij klein is, dan is dit geen probleem. Er bestaan verfijningen van en variaties op de ideeën uit deze paragraaf voor het geval dat p wel heel groot is, en ook wanneer de graad van f vrij groot is.

De Maplecommando's

```
Berlekamp(X^23+X^2+X+1,X) mod 1009;
Factor(X^17+2) mod 101;
```

zijn gebaseerd op de hier beschreven methode. Het tweede commando gebruikt daarbij een variant (voor het eerst beschreven in 1981) die afkomstig is van de Amerikaanse informaticus David G. Cantor en de Duits-Amerikaanse wiskundige Hans Zassenhaus (1912–1991). Zie

www.math.ohio-state.edu/Research/Hans.Zassenhaus/

5.2 getallen

De twee methoden om getallen te factoriseren die hier behandeld gaan worden, zijn beide al vervangen door meer succesvolle algoritmen. De *Pollard $p - 1$ methode* lijkt een beetje op de moderne *elliptische krommen methode* oftewel *ECM*, een voor het eerst in 1987 door de toen in Amsterdam werkende wiskundige Hendrik W. Lenstra jr. beschreven algoritme. De andere methode die we behandelen heet de *kwadratische zeef methode*. Een verbetering hiervan, de *multiple polynomial quadratic sieve* (MPQS) werd in 1986 bedacht door de Amerikaanse wiskundige Peter Montgomery. Een nog recentere generalisatie van het idee van de kwadratische zeef is de *number field sieve* (NFS). deze is in 1988 geïntroduceerd door John Pollard. Anno 2001 zijn MPQS en NFS de snelste factorisatietechnieken die we kennen. Met de laatstgenoemde methode hebben Herman te Riele en anderen in 1999 de getallen RSA-140 en RSA-155 gefactoriseerd.

5.2.1 Pollard $p - 1$

In 1974 introduceerde de Engelse wiskundige John M. Pollard (dezelfde man die later ook de hiervoor al genoemde NFS bedacht) een factorisatiemethode die daarna de naam ‘Pollard $p - 1$ ’ kreeg. Dit werkt als volgt, wanneer we N willen factoriseren.

- Bedenk een vrij groot getal k dat een veelvoud is van alle kleine getallen onder een zekere grens B .
- Neem een $a > 1$ en bereken $a_k := a^k \bmod N$.
- Kijk of $d := \text{ggd}(a_k - 1, N)$ een echte deler is van N . Zoja, dan ben je er in geslaagd om N te factoriseren. Zoniet, dan kan je een andere a en eventueel ook andere B en k proberen.

We proberen uit te leggen waarom dit zou kunnen werken. Stel dat p een (vooralsnog onbekende...) priemdeler is van N . Als voor een zekere ℓ geldt dat $(a \bmod N)^\ell = 1 \bmod N$, dan wil dit precies zeggen dat $a^\ell - 1$ een veelvoud is van N . Omdat p een deler van N is, geldt dan ook dat $a^\ell - 1$ een veelvoud is van p , oftewel $(a \bmod p)^\ell = 1 \bmod p$. Uit deze redenering volgt, dat de orde van $a \bmod p$ in \mathbb{F}_p^* een *deler* is van de orde van $a \bmod N$ in $(\mathbb{Z}/N\mathbb{Z})^*$. Die orde van $a \bmod p$ zou daarbij best een stuk kleiner kunnen zijn dan de orde van $a \bmod N$. Dit is precies wat het algoritme hoopt te kunnen gebruiken: de exponent k die wordt gebruikt heeft heel wat delers, dus er is een redelijke kans dat de gekozen a de eigenschap heeft dat de orde van $a \bmod p$ een van die delers van k is. En dan is p een deler van zowel N als van $a^k - 1$, dus van $\text{ggd}(a^k - 1, N)$. Als N dan maar niet een deler is van $a^k - 1$, of anders gezegd, als de orde van $a \bmod N$ geen deler van k is, dan is die ggd echt kleiner dan N zelf en dus levert deze een factor van N .

De priemdeler p van N kennen we uiteraard van te voren nog niet, en dus kunnen we ook niet zomaar voorspellen welke orde een zekere $a \bmod p$ in \mathbb{F}_p^* zal hebben. Wel zegt de theorie ons, dat die orde een deler zal zijn van $p - 1$. Wij hopen dat de orde een deler is van onze gekozen k , en die is een product van machten van kleine priemgetallen. Onze kans van slagen lijkt dus het grootst indien ook $p - 1$ deelbaar is door veel machten van kleine priemgetallen. Anders gezegd: het Pollard $p - 1$ algoritme is redelijk geschikt voor het vinden van priemfactoren p die de eigenschap hebben dat $p - 1$ deelbaar is door veel machten van kleine priemen. De naam van het algoritme hangt samen met deze eigenschap. Niet elke N heeft echter zulke priemdelers, dus niet voor elke N zal het algoritme goed werken.

Voorbeeld 5.2.2 We gebruiken hier een exponent die deelbaar is door alle positieve getallen tot en met 40, om een getal van 14 cijfers te factoriseren:

```
N:=10004804253107;
k:=2^5*3^3*5^2*7*11*13*17*19*23*29*31*37;
for a from 2 to 30 do igcd(N,-1+a^k mod N) od;
```

Voor $a = 29$ blijken we dan een factor van N te vinden: kennelijk is de grootste gemene deler van N en $29^k - 1$ gelijk aan 100049.

5.2.3 De kwadratische zeef

De Amerikaanse wiskundige Carl Pomerance, die we ook al bij het behandelen van Carmichaelgetallen tegenkwamen, introduceerde in 1982 de kwadratische zeef methode. Aan de basis hiervan staat een heel eenvoudig idee: stel we hebben getallen x en y met de eigenschap dat $(x \bmod N)^2 = (y \bmod N)^2$. Dit betekent dat $x^2 - y^2$ een veelvoud is van N , dus N deelt het product $(x + y)(x - y)$. Een priemdelers p van N moet dan een factor zijn van $x + y$ of van $x - y$. Met een beetje geluk zitten niet al die priemdelers p in dezelfde van deze twee factoren, en als dat zo is dan zijn $\text{ggd}(x + y, N)$ en $\text{ggd}(x - y, N)$ echte factoren van N .

De vraag is natuurlijk hoe je dan wel aan zulke x en y komt, en daarvoor levert de ‘kwadratische zeef’ een oplossing. Neem een verzameling kleine priemgetallen

$$\{p_1, p_2, \dots, p_t\}.$$

Deze verzameling wordt de *factorbasis* genoemd. Laat vervolgens w het grootste gehele getal $\leq \sqrt{N}$ zijn, en beschouw voor zekere $n > 0$ de rij getallen

$$s(-n), s(-n + 1), \dots, s(-1), s(0), s(1), \dots, s(n)$$

waarin $s(i) = (i + w)^2 - N$. Merk op dat $s(i) = i^2 + 2iw + w^2 - N$ en $0 < N - w^2 < 1 + 2\sqrt{N}$ (tenzij N een kwadraat is; in dat geval hoeven we alleen nog maar het veel kleinere getal w te factoriseren). Hieruit volgt dat als we n maar heel veel kleiner dan de wortel uit N kiezen dan zijn alle $s(i)$ in absolute waarde niet erg veel groter dan de wortel uit N . Merk verder op dat $s(i) \bmod N = (i + w)^2 \bmod N$, dus modulo N is elke $s(i)$ een kwadraat.

Nu volgt de ‘zeefstap’ van het algoritme: we gaan zoveel mogelijk getallen $s(i)$ proberen te schrijven als producten van machten van de priemgetallen p_1 tot en met p_t . Dit kan vrij efficiënt: begin met p_1 en kijk of een zekere $s(i)$ deelbaar is door p_1^d . Is dit het geval, dan is eveneens elk van $s(i + p_1^d)$, $s(i + 2p_1^d)$, $s(i + 3p_1^d)$ deelbaar door p_1^d . Immers, zo’n verschil $s(i + kp_1^d) - s(i)$

is een veelvoud van p_1^d zoals je bijvoorbeeld door het uit te schrijven ziet. Dus als we ergens een factor p_1^d vinden, dan mogen we als een soort zeef ook meteen bij een heleboel andere getallen $s(j)$ de aanwezigheid van een factor p_1^d concluderen.

Na dit zeven hebben we ongetwijfeld een aantal $s(i)$'s nog niet als product van machten van p_j 's geschreven, maar de overige zien er uit als

$$s(i) = (-1)^{e_{0,i}} p_1^{e_{1,i}} p_2^{e_{2,i}} \cdot \dots \cdot p_t^{e_{t,i}}.$$

Voor elk rijtje $(x_{-n}, \dots, x_0, x_1, \dots, x_n)$ bestaande uit alleen maar nullen en enen, is $\prod_i s(i)^{x_i}$ natuurlijk precies het product van de $s(i)$ waarbij i de eigenschap $x_i = 1$ heeft. Nemen we $x_i = 0$ voor elke i waarvoor we $s(i)$ niet in termen van de p_j 's gefactoriseerd hebben, dan is het product $\prod_i s(i)^{x_i}$ eveneens volledig gefactoriseerd in termen van de factorbasis. Hierbij heeft het priemgetal p_j als exponent $f_j := e_{j,-n}x_{-n} + \dots + e_{j,1}x_1 + \dots + e_{j,n}x_n$. Zijn al deze f_j even, en is bovendien $e_{0,-n}x_{-n} + \dots + e_{0,0}x_0 + \dots + e_{0,n}x_n$ even, dan hebben we een oplossing van $(x \bmod N)^2 = (y \bmod N)^2$ gevonden: namelijk, met $x = \prod (i+w)^{x_i}$ en $y = \prod p_j^{f_j/2}$.

De x_{-n}, \dots, x_n die hiervoor nodig zijn, zijn op te vatten als oplossingen van het stelsel lineaire vergelijkingen over \mathbb{F}_2 gegeven als

$$e_{j,-n}x_{-n} + \dots + e_{j,0}x_0 + \dots + e_{j,n}x_n = \bar{0},$$

voor $0 \leq j \leq t$. Als het aantal x_j dat hierin niet van te voren is bepaald *groter* is dan het aantal vergelijkingen, dan bestaat voor zo'n stelsel een oplossing waarbij niet alle x_j gelijk aan nul zijn. Met andere woorden, we moeten minstens $t+2$ van de $s(i)$'s volledig tijdens de zeefstap gefactoriseerd hebben, dan werkt de methode.

Het succes van deze techniek en de verfijningen en generalisaties ervan berust enerzijds op het feit dat met name de 'zeefstap' heel geschikt is om door een heleboel computers parallel te worden uitgevoerd. Een belangrijk kenmerk van de recente successen op factorisatiegebied is dan ook, dat een groot aantal mensen erbij betrokken is die allemaal via e-mail of internet programma's toegewezen krijgen en zo een deel van het rekenwerk uitvoeren.

Daarnaast blijkt ook het oplossen van het lineaire stelsel over \mathbb{F}_2 met behulp van een supercomputer vrij goed te doen, ondanks de enorme aantallen variabelen en het aantal vergelijkingen in praktische gevallen. De reden hiervoor is dat sommige in de numerieke wiskunde gebruikte oplosmethoden ook erg goed toepasbaar blijken te zijn voor dit soort stelsels over een eindig lichaam. Dit is van groot belang omdat in de recente factorisaties het aantal benodigde priemen in de factorbasis rond 10^6 ligt. Dergelijk grote lineaire stelsels zijn zonder heel goede methodes niet op te lossen.

Voorbeeld 5.2.4 We gaan $N = 90751$ factoriseren. Hiertoe bepalen we eerst de grootste gehele w met $w^2 \leq N$:

```
N:=90751; w:=isqrt(N);
```

Vervolgens factoriseren we voor een paar kleine getallen i de uitkomst van $s(i) := (w + i)^2 - N$:

```
for i from -2 to 2 do ifactor((w+i)^2-N) od;
```

Uit de uitkomsten concluderen we dat $s(0) \cdot s(-2)$ gelijk is aan $(2 \cdot 3^2 \cdot 5^2)^2$. Modulo N is $s(0)s(-2)$ ook het kwadraat van $w(w - 2)$, en dus proberen we als volgt een factor van N te vinden:

```
igcd(N, 2*3^2*5^2+w*(w-2));
```

Dit levert inderdaad een factor van N op.

5.3 Opgaven

1. Gebruik voor verschillende grote priemgetallen p het polynoom

```
f := expand((X-(1/3 mod p))*(X-(1/5 mod p)) mod p);
```

om te onderzoeken of je een verschil opmerkt in de snelheid van de algoritmen `Berlekamp() mod p` en `Factor() mod p`.

2. Een andere manier om sommige polynomen over \mathbb{F}_p te factoriseren zagen we al in het hoofdstuk over irreducibiliteit: een irreducibele factor van f die graad d heeft, is een deler van $X^{p^d} - X$, en dus van $\text{ggd}(f, X^{p^d} - X)$. Gebruik dit om een aantal polynomen te ontbinden. Kan je uitleggen waarom dit niet werkt voor de f uit de vorige opgave?
3. Gebruik Pollard $p-1$ om elk van de getallen $10^7 + 2n + 1$ met $0 \leq n \leq 8$ te ontbinden.
4. In hoofdstuk 2 is onder meer gezegd dat voor een priemdelers p van een Fermatgetal $F_m = 2^{2^m} + 1$ geldt, dat $p - 1$ een veelvoud is van 2^{m+2} . Wie dus F_m met de Pollard $p - 1$ methode wil factoriseren, doet er verstandig aan een exponent k te kiezen die een veelvoud is van 2^{m+2} . Probeer dit te gebruiken om met de Pollard $p - 1$ methode F_5 en F_6 te factoriseren.
5. Zoek met behulp van `?ifactor` uit welke factorisatietechnieken zoal in Maple zijn voorgeprogrammeerd. Tot welke grootte kan Maple op ons systeem nog getallen binnen enkele seconden factoriseren?
6. Probeer met behulp van de kwadratische zeef 10001 en 100001 te factoriseren.
7. Probeer zoveel mogelijk eigenschappen van n te vinden waaronder het getal $10^n + 1$ *niet* priem is.

6 Codering

Bij cryptografie ging het vooral om het beveiligen van gegevens. Een persoon of computer die beveiligde data weet te onderscheppen, dient het zo lastig mogelijk te hebben om de gegevens ook daadwerkelijk te ontcijferen. Bij coderen ligt het probleem heel anders. Hier versleutelen we onze data ook, maar dat wordt alleen maar gedaan omdat we zo de kans op fouten tijdens het verzenden willen tegengaan. De bedoeling is dat zelfs wanneer door bijvoorbeeld storingen een deel van de gegevens is veranderd of verloren gegaan, we toch uit de rest nog de originele boodschap kunnen reconstrueren. Bekende toepassingen worden gevonden in communicatie met bijvoorbeeld satellieten, en vooral in CD-spelers. Maar zelfs in streepjescodes, ISBN-nummers en de nummers op bankbiljetten zit een code.

De *Universal Product Code* (UPC) wordt voor streepjescodes op vrijwel ieder product gebruikt. Meestal zie je dan de zogeheten ‘UPC versie A’. Deze beschrijft een product als een rijtje van $6 + 6 = 12$ cijfers, eerst 6 links en daarnaast 6 rechts van een scheiding in het midden. Zo’n cijfer wordt weergegeven als een rij van 7 verticale witte of zwarte strepen. Twee witte strepen naast elkaar ziet er dan uit als een iets bredere witte streep, drie witte naast elkaar als een nog bredere, en evenzo zien we een aantal zwarte strepen naast elkaar als een bredere zwarte streep. De 6 ‘linkercijfers’ beginnen allemaal met een witte streep geheel links, ze hebben een oneven aantal zwarte strepen, en ze eindigen met een zwarte streep geheel rechts. Bovendien, als we niet op de breedte letten, dan ziet elk linkercijfer eruit als wit-zwart-wit-zwart. En precies omgekeerd beginnen de ‘rechtercijfers’ met een zwarte streep links, ze hebben een even aantal zwarte strepen, en ze hebben een witte streep geheel rechts. Letten we niet op de breedte van de strepen, dan ziet elk rechtercijfer er hetzelfde uit: zwart-wit-zwart-wit. Bijvoorbeeld een 0 ziet er, als ‘linkercijfer’ uit als een witte streep die drie eenheden breed is, gevolgd door een zwarte streep die twee eenheden breed is, gevolgd door een witte streep van één eenheid breed en dan tenslotte een zwarte streep met breedte één. We noteren dit als 0001101. Voor het coderen

wordt verder de volgende tabel gebruikt.

linkercijfer-code	cijfer	rechtercijfer-code
0001101	0	1110010
0011001	1	1100110
0010011	2	1101100
0111101	3	1000010
0100011	4	1011100
0110001	5	1001110
0101111	6	1010000
0111011	7	1000100
0110111	8	1001000
0001011	9	1110100

Om het automatisch scannen van zo'n streepjescode te vereenvoudigen, zijn verder geheel links en geheel rechts van het rijtje van 12 cijfers drie strepen zwart-wit-zwart (oftewel 101) toegevoegd die naar beneden toe iets langer doorlopen dan de strepen bij gewone gecodeerde cijfers. En als scheiding tussen de 6 linkercijfers en de 6 rechtercijfers staan 5 strepen 01010 die eveneens iets verder naar beneden doorlopen. Hiermee kan voorkomen worden dat een product 'ondersteboven' gelezen wordt. Uiteraard zorgt ook het verschil tussen even en oneven aantallen zwarte strepen links en rechts daar al voor.

6.1 binaire lineaire codes

De symbolen op een toetsenbord zijn op allerlei manieren om te zetten in rijtjes nullen en enen. Bijvoorbeeld ASCII oftewel de "*American Standard Code for Information Interchange*" is zo'n manier waarbij 128 karakters worden omgezet in de getallen 0 tot en met 127. Binair geven we dit weer als 0000000 tot en met 1111111. De getallen kleiner dan 32 (dat zijn degenen die binair beginnen met twee nullen) horen daarbij bij speciale karakters, en zo ook 127 (daarbij hoort het 'delete'-karakter). De overigen staan in de volgende tabel.

0100000	[spatie]	0110011	3	1000110	F	1011001	Y	1101100	l
0100001	!	0110100	4	1000111	G	1011010	Z	1101101	m
0100010	"	0110101	5	1001000	H	1011011	[1101110	n
0100011	#	0110110	6	1001001	I	1011100	\	1101111	o
0100100	\$	0110111	7	1001010	J	1011101]	1110000	p
0100101	%	0111000	8	1001011	K	1011110	^	1110001	q
0100110	&	0111001	9	1001100	L	1011111	_	1110010	r
0100111	'	0111010	:	1001101	M	1100000	'	1110011	s
0101000	(0111011	;	1001110	N	1100001	a	1110100	t
0101001)	0111100	<	1001111	O	1100010	b	1110101	u
0101010	*	0111101	=	1010000	P	1100011	c	1110110	v
0101011	+	0111110	>	1010001	Q	1100100	d	1110111	w
0101100	,	0111111	?	1010010	R	1100101	e	1111000	x
0101101	-	1000000	@	1010011	S	1100110	f	1111001	y
0101110	.	1000001	A	1010100	T	1100111	g	1111010	z
0101111	/	1000010	B	1010101	U	1101000	h	1111011	{
0110000	0	1000011	C	1010110	V	1101001	i	1111100	
0110001	1	1000100	D	1010111	W	1101010	j	1111101	}
0110010	2	1000101	E	1011000	X	1101011	k	1111110	~

Bijvoorbeeld het woordje ‘ja’ wordt in ASCII weergegeven als een rij 11010101100001. Als hierin een enkele 0 of 1 verkeerd zou staan, of verkeerd gelezen wordt, dan ontstaat een ander woord. Zo zou 10010101100001 (alleen de tweede 1 is veranderd) gelezen worden als ‘Ja’; geen ernstig verschil, maar wel iets anders dan we hadden geschreven.

De kunst bij coderingstheorie is nu om aan zo’n rijtje nullen en enen op zo’n manier nog extra informatie toe te voegen, dat we daardoor kunnen zien of er in het gelezen rijtje een fout zit, en zoja dat we die fout zelfs kunnen corrigeren. Zo zouden we aan de 7 bits van een ascii-symbool een achtste kunnen toevoegen die ervoor zorgt, dat elk ascii-symbool in z’n 8 bits een even aantal 1-en heeft. Ontvangen we dan een letter waarin we een oneven aantal 1-en zien, dan weten we dat er iets fout is. Alleen weten we natuurlijk niet welke bit(s) verkeerd staan.

Een *binaire code* van lengte n is een manier om de symbolen uit een bepaald ‘alfabet’ weer te geven door rijtjes $001 \cdots 1$ enzovoorts, bestaande uit precies n nullen en enen. Meestal zeggen we simpelweg dat de binaire code de verzameling is van precies die rijtjes die met een symbool uit het gegeven alfabet corresponderen, zonder dat we precies zeggen hoe die correspondentie dan wel werkt. Het getal n hier wordt dus de *lengte* van de code genoemd.

We schrijven \mathbb{F}_2^n voor de verzameling van *alle* rijtjes bestaande uit n nullen en enen. Een binaire code is dus niets anders als een deelverzameling van \mathbb{F}_2^n .

Elk element $a = a_1a_2 \cdots a_n$ (alle a_i uit \mathbb{F}_2) heeft een zogeheten *Hammingnorm* $|a|$, gedefinieerd als het aantal coördinaten van a dat $\neq 0$ is. Dus bijvoorbeeld $|0010001| = 2$ en $|1111110| = 6$. Dit begrip is genoemd naar de Amerikaanse wiskundige Richard W. Hamming (1915–1998). Zie ook

www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Hamming.html

Twee elementen a en b uit \mathbb{F}_2^n kunnen we coördinaatsgewijs (in $\mathbb{F}_2!$) optellen. Het resultaat noteren we als $a + b$. Zo is bijvoorbeeld

$$0010001 + 1111110 = 1101111.$$

Met deze bewerking wordt \mathbb{F}_2^n een (commutatieve) groep, met $00 \cdots 0$ als eenheidselement.

Een binaire code $C \subset \mathbb{F}_2^n$ heet een *lineaire binaire code* als C , met de op \mathbb{F}_2^n gegeven optelling, zelf ook een groep is. Dit is precies dan het geval, indien $00 \cdots 0 \in C$, en bovendien bij elke $a, b \in C$ is ook $a + b \in C$.

Als we op \mathbb{F}_2^n ook nog een *scalair vermenigvuldiging* met elementen uit \mathbb{F}_2 definiëren, namelijk $1 \cdot a = a$ en $0 \cdot a = 00 \cdots 0$, is daarmee \mathbb{F}_2^n een *lineaire ruimte over het lichaam \mathbb{F}_2* . Een lineaire binaire code verdient zijn naam aan het feit, dat het niets anders is dan een lineaire deelruimte van \mathbb{F}_2^n . Zo'n lineaire deelruimte C heeft een *basis*, dat wil zeggen een aantal elementen $a_1, \dots, a_k \in C$ met de eigenschap dat geen van de a_j een som is van andere a_i 's, en elk element uit C is een som van a_j 's. Het (voor iedere keuze van een basis gelijke) aantal elementen in zo'n basis heet de *dimensie* van C . De dimensie van een lineaire binaire code wordt standaard met de variabele k aangeduid.

Een $C \subset \mathbb{F}_2^n$ heeft tenslotte een *minimale (Hamming)afstand* die als d wordt genoteerd. Dit is per definitie de *minimale* waarde die de Hamming-norm van $a - b$ aanneemt, als a en b onderling verschillende elementen uit C zijn. Per definitie is $|a - b|$ precies het aantal coördinaten waarin a en b verschillen. Als een code C dus minimale afstand d heeft, dan wil dit zeggen dat er elementen in C zijn die op precies d plaatsen van elkaar verschillen, en er zijn geen elementen in C die op *minder* dan d plaatsen verschillen. Merk op dat als C een *lineaire* code is, dan is met $a, b \in C$ ook hun verschil $a - b \in C$, dus de minimale afstand is dan gelijk aan de minimale waarde $|c|$ waarbij c de elementen $\neq 00 \cdots 0$ van C doorloopt.

Een binaire $[n, k, d]$ -code (of ook wel, *binaire code van type $[n, k, d]$*), is per definitie een lineaire binaire code $C \subset \mathbb{F}_2^n$ met dimensie k en minimale afstand d . We geven een aantal eigenschappen van zulke binaire $[n, k, d]$ -codes. Stel dat $C \subset \mathbb{F}_2^n$ zo'n code is.

- Neem aan dat voor $v \in \mathbb{F}_2^n$ en $a \in C$ geldt, dat $|v - a| \leq e$ voor een e met $2e \leq d - 1$. Dan bestaat er geen ander element $b \neq a$ in C waarvoor ook geldt $|v - b| \leq e$.

Immers, als v en b in hoogstens e coördinaten van elkaar verschillen, en v en a ook, dan kunnen a en b hooguit in $2e$ coördinaten verschillend zijn. Maar zowel a als b zitten in C , en de minimale afstand in C is meer dan $2e$, dus dit is alleen maar mogelijk als $a = b$.

Uit de hier bewezen uitspraak volgt, dat de code C ‘fouten kan verbeteren’: als een element c uit C wordt gelezen als \tilde{c} , met $|\tilde{c} - c| < d/2$, dan kan je uit \tilde{c} het element c terugvinden. Dus als er minder dan $d/2$ fouten zijn gemaakt, dan zijn die alle te verbeteren.

- De ruimte \mathbb{F}_2^n bestaat uit 2^n elementen, en evenzo heeft C precies 2^k elementen.

Immers, in \mathbb{F}_2^n heb je n coördinaten, en op elk van die plaatsen kan je een 0 of een 1 zetten. Dit levert in totaal 2^n mogelijkheden. Hetzelfde argument werkt voor C : daarvoor bestaat een basis c_1, \dots, c_k . Een willekeurig element van C is dan te schrijven als

$$\lambda_1 c_1 + \lambda_2 c_2 + \dots + \lambda_k c_k,$$

waarin je de $\lambda_i \in \mathbb{F}_2$ vrij kan kiezen. Dat levert 2^k mogelijkheden.

- Er geldt $d + k \leq n + 1$.

Immers, C bestaat uit 2^k elementen. En als je van elk element uit C de laatste $d - 1$ coördinaten weglaat, dan hou je op die manier 2^k onderling verschillende elementen uit \mathbb{F}_2^{n-d+1} over. Want als twee elementen $c_1, c_2 \in C$ precies dezelfde $n - d + 1$ eerste coördinaten hebben, dan verschillen ze hooguit nog in de laatste $d - 1$ coördinaten terwijl $d - 1$ kleiner is dan de minimale afstand in C . Dat kan dus niet voor $c_1 \neq c_2$. We concluderen dat $2^k \leq 2^{n-d+1}$ en dus $d + k \leq n + 1$ zoals we wilden aantonen.

Een binaire code $C \subset \mathbb{F}_2^n$ heet *cyclisch* wanneer met elke $(a_0, a_1, \dots, a_{n-1}) \in C$ ook het element $(a_{n-1}, a_0, a_1, \dots)$ in C zit.

De theorie van cyclische lineaire binaire codes is heel mooi in algebra te vertalen. Daartoe identificeren we de ruimte \mathbb{F}_2^n met de ring $R_n := \mathbb{F}_2[X]/(X^n - 1)$. Het element $(a_0, \dots, a_{n-1}) \in \mathbb{F}_2^n$ correspondeert daarbij met $a_0 + a_1 X + \dots + a_{n-1} X^{n-1} \pmod{(X^n - 1)}$ uit R_n . Hieronder gaat optellen in \mathbb{F}_2^n over in het gewone optellen van veeltermen. En het cyclisch doorschuiven zoals dat in de definitie van een cyclische code gebeurt, wordt precies

het vermenigvuldigen met $X \bmod (X^n - 1)$ in R_n . Immers,

$$\begin{aligned} & X \bmod (X^n - 1) \\ & \quad \times \\ & a_0 + a_1X + \dots + a_{n-1}X^{n-1} \bmod (X^n - 1) \\ & \quad = \\ & a_{n-1} + a_0X + \dots + a_{n-2}X^{n-1} \bmod (X^n - 1), \end{aligned}$$

want $a_{n-1}X^n \bmod (X^n - 1) = a_{n-1} \bmod (X^n - 1)$. Dus een cyclische lineaire binaire code is op te vatten als een deelverzameling $C \subset R_n$ die 0 bevat en de eigenschap heeft dat met c en c' ook $c + c'$ en xc in C zitten, waarbij $x = X \bmod (X^n - 1)$. Maar als dan $c \in C$, dan is ook x^2c en verder elke $x^j c$ een element van C , en dus ook elke som $a_0c + a_1xc + \dots + a_{n-1}x^{n-1}c = (a_0 + \dots + a_{n-1}x^{n-1})c \in C$. De conclusie is, dat zo'n cyclische lineaire binaire code precies een *ideaal* is in de ring $\mathbb{F}_2[X]/(X^n - 1)$.

Omgekeerd is elk ideaal in deze ring zo'n cyclische lineaire binaire code, want met twee elementen zit ook hun som erin, en het product met ieder element, dus in het bijzonder met $X \bmod (X^n - 1)$ is weer een element van het ideaal.

6.2 drie voorbeelden

We gaan drie soorten binaire codes behandelen, namelijk de Hamming code, bepaalde 'Reed-Solomon-codes' en tenslotte 'Rijndael'.

6.2.1 De Hamming code

Schrijf $R := \mathbb{F}_2[X]/(X^7 - 1)$ en $x = X \bmod (X^7 - 1) \in R$. Elk element in R is dan op unieke wijze te schrijven als $a_0 + a_1x + \dots + a_6x^6$, met $a_0, \dots, a_6 \in \mathbb{F}_2$. Omdat

$$X^7 - 1 = (X - 1)(X^3 + X + 1)(X^3 + X^2 + 1)$$

in $\mathbb{F}_2[X]$, is $x^3 + x + 1$ geen eenheid in R . De (standaard) *Hamming code* $H \subset R$ bestaat uit alle elementen uit het ideaal $R \cdot (x^3 + x + 1)$.

Dit is een cyclische lineaire binaire code met lengte 7, geïntroduceerd door de al eerder genoemde Richard Hamming in 1950. Ter illustratie van het rekenen met cyclische lineaire codes, zullen we nu de dimensie en de minimale afstand van H bepalen: H blijkt een $[7, 4, 3]$ -code te zijn.

We gebruiken daartoe, dat H precies het beeld is van de afbeelding

$$\varphi : R \rightarrow H$$

gegeven door $\varphi(f(x)) = f(x) \cdot (x^3 + x + 1)$. Deze φ is een lineaire afbeelding, en we zullen gebruik maken van een resultaat uit de lineaire algebra dat zegt dat dan (in onze situatie, met een R die dimensie 7 heeft)

$$\dim \text{Ker}(\varphi) + \dim H = 7.$$

Een aantal elementen van H kennen we al, namelijk $1101000 = 1 + x + x^3$ en $0110100 = x + x^2 + x^4$ en $0011010 = x^2 + x^3 + x^5$ en $0001101 = x^3 + x^4 + x^6$. Dit viertal is onafhankelijk (geen enkele is een som van anderen), dus $\dim H \geq 4$. Verder kennen we ook wel elementen uit $\text{Ker}(\varphi)$, oftewel elementen $f(x) \in R$ met de eigenschap $\varphi(f(x)) = 0$. Immers,

$$(x-1)(x^3+x^2+1)(x^3+x+1) = x^7 - 1 = 0,$$

dus zitten $(x-1)(x^3+x^2+1)$ en $x(x-1)(x^3+x^2+1)$ en $x^2(x-1)(x^3+x^2+1)$ alledrie in $\text{Ker}(\varphi)$. Deze zijn onafhankelijk, dus $\dim \text{Ker}(\varphi) \geq 3$.

We hebben nu gezien dat

$$4 \leq \dim H = 7 - \dim \text{Ker}(\varphi) \leq 7 - 3 = 4,$$

en dus is $\dim H = 4$.

Eenzelfde argument laat algemener zien dat als $X^n - 1 = f(X) \cdot g(X)$ in $\mathbb{F}_2[X]$, waarin $\text{graad}(f) = \ell$, dan heeft de code gegeven door het ideaal $(f(X) \bmod (X^n - 1))$ in $\mathbb{F}_2[X]/(X^n - 1)$ dimensie $n - \ell$.

Rest ons nog, de minimale afstand d van H te bepalen. De regel $d + k \leq n + 1$ levert ons dat $d \leq 7 + 1 - 4 = 4$. Maar dat wisten we ook wel zonder deze regel, want bijvoorbeeld $x^3 + x + 1 \in H$ heeft Hammingnorm 3. De vraag is dus, of er in H ook nog elementen $\neq 0$ zitten waarin hoogstens twee verschillende machten x^i voorkomen. Dat blijkt niet het geval te zijn: ten eerste, als een element x^i in H zou zitten, dan zou vanwege het cyclisch zijn ook $x^7 = 1$ een element van H zijn. Echter, H is een ideaal, en als 1 daar in zit, dan ook elke $f(x) \cdot 1$, en dus zou gelden $H = R$. Dat is niet het geval zoals we bijvoorbeeld al aan de dimensies gezien hebben.

Dan nog de mogelijkheid dat $x^a + x^b \in H$, met $0 \leq a < b \leq 6$. Opnieuw vanwege het cyclisch zijn, zou hieruit volgen dat H een element van de vorm $1 + x^c$ bevat met $1 \leq c \leq 6$. Dus in R zou gelden $x^c + 1 = f(x) \cdot (x^3 + x + 1)$ voor zekere $f(x) \in R$. Dit impliceert in $\mathbb{F}_2[X]$ een gelijkheid

$$X^c + 1 = f(X) \cdot (X^3 + X + 1) + g(X) \cdot (X^7 - 1)$$

voor zekere $f(X), g(X) \in \mathbb{F}_2[X]$. Alle termen in het rechterlid hiervan zijn veelvouden van $X^3 + X + 1$, dus zou $X^c + 1$ dat ook moeten zijn. Anders gezegd,

$$X^c \bmod (X^3 + X + 1) = 1 \bmod (X^3 + X + 1).$$

Hiermee zijn we snel klaar: $X^3 + X + 1$ is irreducibel in $\mathbb{F}_2[X]$ en dus is $\mathbb{F}_2[X]/(X^3 + X + 1)$ een lichaam, en wel eentje met 8 elementen. In dit lichaam is $X \bmod (X^3 + X + 1) \neq 0$, dus het is een eenheid, en die heeft een orde die een deler is van $8 - 1 = 7$. Omdat die orde niet 1 is, komt er dan 7 uit. De c die we hadden zou dan een veelvoud van die orde moeten zijn en dat is onmogelijk omdat $c \leq 6$. Conclusie: zo'n som van twee monomen in H bestaat niet.

Uiteraard hadden we dat ook wel zonder zoveel theorie kunnen zien, want H bestaat slechts uit $2^4 = 16$ elementen en het is niet veel werk om die alle 16 op te schrijven.

Omdat $d/2 > 1$ voor de Hammingcode, kan deze code één fout verbeteren. We geven in een voorbeeld aan, hoe dit in z'n werk gaat.

Stel dat we een bericht $1001001 = 1 + x^3 + x^6$ ontvangen. Ten eerste gaan we na of dit een element van H is:

$$\text{Rem}(1+X^3+X^6, X^3+X+1, X) \bmod 2;$$

levert als antwoord $X^2 + X + 1$. Dit betekent dat we een rest overhouden wanneer we $1 + x^3 + x^6$ door $x^3 + x + 1$ proberen te delen, dus dit bericht zit niet in de code H . Er kan dan hooguit 1 element van H zijn dat op slechts een enkele plek van ons bericht verschilt. Neem aan dat dit het element $f(x)(x^3 + x + 1)$ is, en dat het verschil in x^i zit. Dan zou gelden

$$f(x)(x^3 + x + 1) + x^i = 1 + x^3 + x^6,$$

en in het bijzonder zou dan ook X^i rest $X^2 + X + 1$ moeten hebben bij deling door $X^3 + X + 1$. Met het programmaatje

```
i:=0;
while (Rem(X^i,X^3+X+1,X)mod 2)<>X^2+X+1
  do i:=i+1 od;
print('i');
```

zie je dat dit het geval is voor $i = 5$. Kortom, het gecorrigeerde bericht is kennelijk 1001011.

6.2.2 Reed-Solomon codes

In 1960 introduceerden de Amerikaanse electrotechnici Irving S. Reed en Gustave Solomon (toentertijd beide werkzaam bij het Massachusetts Institute of Technology) een nieuwe manier om codes te maken. We geven hier een paar ideeën uit hun constructie. Het zijn dit soort codes die bijvoorbeeld in CD's

worden gebruikt. Eigenlijk zullen we het onszelf hier iets moeilijker maken dan strikt noodzakelijk. We gaan namelijk zoals we dat steeds deden *binair* codes maken. De gebruikelijke Reed-Solomon codes doen dat niet: daar wordt een element uit de code beschouwd als een rijtje elementen uit \mathbb{F}_{2^m} voor zekere m . Wij gaan alles uitdrukken in rijtjes nullen en enen.

Daartoe beginnen we met het lichaam \mathbb{F}_{2^m} . Een in de praktijk veel gebruikt voorbeeld is \mathbb{F}_{128} . Wij kiezen hier voor het gemak $m = 4$, dus \mathbb{F}_{16} . Daarvoor kiezen we het irreducibele polynoom $X^4 + X + 1 \in \mathbb{F}_2[X]$. Dan is $\mathbb{F}_{16} = \mathbb{F}_2[X]/(X^4 + X + 1)$. Met $x := X \bmod (X^4 + X + 1)$ kunnen we elk element uit \mathbb{F}_{16} geven als een rijtje $a_0a_1a_2a_3 = a_0 + a_1x + a_2x^2 + a_3x^3$, met alle $a_i \in \mathbb{F}_2$. Het rekenen met zulke rijtjes is goed te doen; optellen werkt gewoon coördinaatsgewijs, dus bijvoorbeeld $1001 + 1111 = 0110$. En het vermenigvuldigen gebruikt het Maplecommando

```
f:=a0+a1*X+a2*X^2+a3*X^3: g:=b0+b1*X+b2*X^2+b3*X^3:
Rem(f*g,X^4+X+1,X) mod 2;
```

waarmee je bijvoorbeeld ziet dat $1001 * 1111 = 0111$.

Kies nu een getal ℓ met $1 \leq \ell < 2^m$. Onder de ruimte $\mathcal{P}_\ell(\mathbb{F}_{2^m})$ verstaan we per definitie alle veeltermen $f(X)$ met coëfficiënten in \mathbb{F}_{2^m} , en graad $\leq \ell$.

Nummer nu de elementen van \mathbb{F}_{2^m} als

$$\mathbb{F}_{2^m} = \{\alpha_1, \alpha_2, \dots, \alpha_{2^m}\}.$$

De Reed-Solomon code bij onze \mathbb{F}_{2^m} en ℓ bestaat dan per definitie uit alle rijen

$$(f(\alpha_1), f(\alpha_2), \dots, f(\alpha_{2^m})),$$

waarbij f de ruimte $\mathcal{P}_\ell(\mathbb{F}_{2^m})$ doorloopt.

Omdat elke $f(\alpha_i)$ een element uit \mathbb{F}_{2^m} is, wordt deze weergegeven als een rijtje van m nullen en enen. Er staan in een codewoord 2^m zulke rijtjes achter elkaar, dus we krijgen een binaire code met lengte $m2^m$. Merk verder op, dat deze codes lineair zijn. Immers, als we een rij met coördinaten $f(\alpha_i)$ hebben, en een andere rij met coördinaten $g(\alpha_i)$, dan hoort de som hiervan precies bij de veelterm $f + g$, dus die zit ook in de code.

We beperken ons nu tot onze \mathbb{F}_{16} en proberen dan iets te zeggen over de dimensie en de minimale afstand van de zo verkregen codes.

De eerste opmerking die daarbij gemaakt kan worden, is dat $\mathcal{P}_\ell(\mathbb{F}_{16})$ een lineaire ruimte is over het lichaam \mathbb{F}_{16} . En wel eentje met dimensie $\ell + 1$ over \mathbb{F}_{16} , want $1, X, \dots, X^\ell$ is een basis. Noemen we onze code C , dan is C opgevat als deel van \mathbb{F}_{16}^{16} precies het beeld van de afbeelding

$$\psi : \mathcal{P}_\ell(\mathbb{F}_{16}) \rightarrow \mathbb{F}_{16}^{16}$$

gegeven door $\psi(f) = (f(\alpha_1), f(\alpha_2), \dots, f(\alpha_{16}))$. Deze ψ is een lineaire afbeelding tussen lineaire ruimten over \mathbb{F}_{16} . Een f in de kern van ψ is een polynoom met de eigenschap dat $f(\alpha) = 0$ voor elke $\alpha \in \mathbb{F}_{16}$. Dat zou betekenen dat f een veelvoud is van $\prod_{i=1}^{16} (X - \alpha_i)$, dus ofwel $f = 0$, of f heeft graad minstens 16. Omdat wij aannemen dat $\text{graad}(f) \leq \ell < 16$, volgt dat $f = 0$. Conclusie: als lineaire ruimte over \mathbb{F}_{16} heeft onze code een dimensie die gelijk is aan de dimensie van $\mathcal{P}_\ell(\mathbb{F}_{16})$, en die is $\ell + 1$.

Hieruit is vervolgens ook de dimensie over \mathbb{F}_2 te bepalen. Dat kan door te tellen: onze ruimte heeft $16^{\ell+1} = 2^{4\ell+4}$ elementen, dus moet de dimensie over \mathbb{F}_2 wel $4\ell + 4$ zijn. Een andere manier om tot ditzelfde antwoord te komen is door op te merken, dat als $v_1, \dots, v_{\ell+1}$ een basis is voor de code als lineaire ruimte over \mathbb{F}_{16} , dan is de verzameling $v_i, xv_i, x^2v_i, x^3v_i$ met $i = 1, \dots, \ell + 1$ een basis voor dezelfde code als lineaire ruimte over \mathbb{F}_2 .

Tenslotte proberen we iets over de minimale afstand d te zeggen. Het constante polynoom 1 zit in $\mathcal{P}_\ell(\mathbb{F}_{16})$, en voor iedere $\alpha \in \mathbb{F}_{16}$ neemt dit de waarde $1 = 1000 \in \mathbb{F}_{16}$ aan. Kortom, dit levert een codewoord $10001000 \dots 1000$ met Hammingnorm 16. Dus $d \leq 16$.

Om een ondergrens voor d te vinden merken we verder op, dat een polynoom $f \neq 0$ in $\mathcal{P}_\ell(\mathbb{F}_{16})$ hoogstens ℓ nulpunten in \mathbb{F}_{16} kan hebben. Dus als we $f(\alpha_1)$ tot en met $f(\alpha_{16})$ opschrijven, dan zijn tenminste $16 - \ell$ van deze elementen uit \mathbb{F}_{16} niet gelijk aan 0000. Dat betekent dat het bijbehorende codewoord Hammingnorm $\geq 16 - \ell$ heeft, dus $d \geq 16 - \ell$.

Als er echt een element met Hammingnorm $16 - \ell$ in onze code zou zitten, dan hoort dit bij een polynoom met precies ℓ nulpunten in \mathbb{F}_{16} (dus een polynoom van graad ℓ), en voor elk van de $16 - \ell$ andere punten in \mathbb{F}_{16} neemt dit polynoom de waarde 1000 of 0100 of 0010 of 0001 aan. Echter, een polynoom van graad ℓ kan niet vaker dan ℓ keer eenzelfde waarde aannemen, dus hieruit volgt $\ell \geq 4$.

Voor $\ell = 1$ hebben we dus een $[64, 8, 16]$ -code. Voor $\ell = 2$ is het een $[64, 12, d]$ -code met $d = 15$ of $d = 16$. Echter, $d = 15$ is ook in dit geval niet mogelijk. Hadden we namelijk een polynoom f dat een element in de code met Hammingnorm 15 zou opleveren, dan hebben we bij $\ell = 1$ al gezien dat dit polynoom niet graad ≤ 1 kan hebben. Het zou dus moeten gaan om een polynoom van graad 2. Dit heeft $n \leq 2$ nulpunten in \mathbb{F}_{16} , en verder zijn er $m \leq 8$ elementen van \mathbb{F}_{16} waar het polynoom een van de waarden 1000 of 0100 of 0010 of 0001 kan aannemen. De overige $16 - n - m$ elementen van \mathbb{F}_{16} leveren elementen van \mathbb{F}_{16} op waarin minstens twee keer een 1 voorkomt. Kortom, f geeft ons een codewoord met Hammingnorm $\geq m + 2(16 - n - m) = 32 - m - 2n \geq 20$. Conclusie: voor $\ell = 2$ hebben we een $[64, 12, 16]$ -code.

Voor grotere ℓ is het precies bepalen van d lastiger. Zelfs met onze ruwe ondergrens zien we echter al dat het hier om vrij goede codes gaat: bijvoorbeeld voor $\ell = 7$ krijgen we een 32-dimensionale code van lengte 64, met de eigenschap dat elk tweetal codewoorden op minstens $16 - 7 = 9$ plaatsen van elkaar verschilt.

Met Maple zou je als volgt in zo'n code kunnen rekenen.

```
alias(x=RootOf(_Z^4+_Z+1)):
f:=proc(X) X^5+X^4+X+x+x^2 end: H:=0:
for a from 0 to 1 do for b from 0 to 1
do for c from 0 to 1 do for d from 0 to 1 do
g:=simplify(f(a+b*x+c*x^2+d*x^3)) mod 2):
H:=H+subs(x=1,g)
od od od od:
H;
```

Deze regels laten zien dat een zeker codewoord, corresponderend met het polynoom $X^5 + X^4 + X + x^2 + x$ van graad 5, een Hammingnorm gelijk aan 18 heeft.

6.2.3 Rijndael

In 1997 riep de Amerikaanse NIST (National Institute of Standards and Technology) cryptografen uit de hele wereld op, om een ontwerp voor een nieuwe encryptiestandaard bij hen in te dienen. De oude 'Digital Encryption Standard' (DES), inmiddels tientallen jaren gebruikt door onder meer de Amerikaanse regering maar ook door duidenden bedrijven en instellingen uit de hele wereld, was nodig aan vervanging toe. Uit de voorgestelde ontwerpen zou dan een nieuwe 'Advanced Encryption Standard' worden gekozen.

In december 2001 werd officieel het systeem "Rijndael", ontworpen door de Belgische cryptografen Vincent Rijmen en Joan Daemen, uitgeroepen tot de nieuwe standaard. Met ingang van 26 mei 2002 versleutelen Amerikaanse overheidsinstellingen hun gevoelige informatie en e-mail met behulp van Rijndael. Op

<http://ratchkov.com/vpn/aes/aes.html>

lees je hier meer over. Strikt genomen is dit systeem geen fouten-verbeterend coderingssysteem, dus eigenlijk zou het beter in Hoofdstuk 3 passen. Maar omdat we nu toch met binaire codes bezig zijn, behandelen we het hier.

Rijndael kent 'bits', oftewel elementen van \mathbb{F}_2 , verder 'bytes', oftewel rijtjes van 8 bits, die wij steeds zullen weergeven als polynoom in $\mathbb{F}_2[x]$ van

graad ≤ 7 . Dus bijvoorbeeld de ‘byte’ 01110001 schrijven we als $x + x^2 + x^3 + x^7$. Vervolgens zijn er ‘words’, en dat zijn rijtjes van 4 bytes. Wij vatten een ‘word’ op als een polynoom in twee variabelen $f_0(x) + f_1(x)y + f_2(x)y^2 + f_3(x)y^3$, waarin de f_i bytes voorstellen. Tenslotte zijn er ‘states’, en dat zijn rijtjes (w_1, w_2, w_3, w_4) van vier ‘words’. De verzameling van alle mogelijke ‘states’ noemen we \mathcal{S} .

Het hele Rijndael-systeem kan worden opgevat als een codering (‘encoding’)

$$\epsilon : \mathcal{S} \longrightarrow \mathcal{S}.$$

Hierin hangt de afbeelding ϵ af van een van te voren afgesproken geheime sleutel (‘key’) $\kappa \in \mathcal{S}$. Om uit een versleuteld bericht uit \mathcal{S} de originele boodschap te reconstrueren, bestaat een ‘decoding’-afbeelding

$$\delta : \mathcal{S} \longrightarrow \mathcal{S}$$

met de eigenschap dat $\delta(\epsilon(s)) = s$ voor elke $s \in \mathcal{S}$.

Het versleutelen gebruikt een aantal bewerkingen op elementen van \mathcal{S} , die we nu in Maple weergeven.

```

mx:=x^8+x^4+x^3+x+1:
nx:=x^8-1:
ax:=x^4+x^3+x^2+x+1:
bx:=x^6+x^5+x+1:
ny:=y^4-1:
c:=x+y+y^2+y^3x*y^3:

mw:=proc(w) Rem(Rem(c*w,m,x)mod 2,n,y)mod 2 end proc:
m:=proc(s) [mw(s[1]),mw(s[2]),mw(s[3]),mw(s[4])]
end proc:

lb:=proc(f) Powmod(f,254,mx,x)mod 2 end proc:
tb:=proc(f) Rem(ax*lb(f)+bx,nx,x)mod 2 end proc:
tw:=proc(w) tb(coeff(w,y,0))+tb(coeff(w,y,1)*y+
tb(coeff(w,y,2)*y^2+tb(coeff(w,y,3)*y^3) end proc:
t:=proc(s) [tw(s[1]),tw(s[2]),tw(s[3]),tw(s[4])]
end proc:

r:=proc(s) [s[1],Rem(y*s[2],ny,y)mod 2,
Rem(y^2*s[3],ny,y)mod 2,
Rem(y^3*s[4],ny,y)mod 2] end proc:

```

```

a:=proc(s) m(r(t(s))) end proc:

v:=proc(k,s) [Rem(k[1]+s[1],ny,y)mod 2,
              Rem(k[2]+s[2],ny,y)mod 2,
              Rem(k[3]+s[3],ny,y)mod 2,
              Rem(k[4]+s[4],ny,y)mod 2] end proc:

sl:=array(0..10):
k:=[x^3,x^5*y,x^7*y^2,y^3]:
sl[0]:=k: for j from 1 to 10 do
          sl[j][1]:=sl[j-1][1]+
            Powmod(x,j-1,mx,x)mod 2+tw(y^3*sl[j-1][4]):
          sl[j][2]:=sl[j-1][2]+sl[j][1]:
          sl[j][3]:=sl[j-1][3]+sl[j][2]:
          sl[j][4]:=sl[j-1][4]+sl[j][3] od:

eps:=proc(s) h:=s:
      for j from 0 to 9 do h:=a(v(sl[j],h))) od:
      v(sl[10],r(t(h))) end proc;

```

Het algoritme `eps` (Rijndael) is dus in feite opgebouwd uit een aantal eenvoudige bouwstenen, namelijk verschuivingen v_ℓ over een $\ell \in \mathcal{S}$, gegeven door

$$v_k(s) = s + k$$

en verder vermenigvuldigen met y :

$$\mu_y : \mathbb{F}_2[x, y]/(y^4 - 1) \longrightarrow \mathbb{F}_2[x, y]/(y^4 - 1) : \mu_y(f) := yf$$

en

$$\lambda : \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1) \longrightarrow \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1) : \lambda(f) := f^{254}$$

en tenslotte

$$\tau : \mathbb{F}_2[x]/(x^8 - 1) \longrightarrow \mathbb{F}_2[x]/(x^8 - 1)$$

gegeven door $\tau(f) := (x^4 + x^3 + x^2 + x + 1)f + x^6 + x^5 + x + 1$. Al deze bouwstenen blijken inverteerbare afbeeldingen te zijn, en dus is het totale versleutelen dat ook.

Om een boodschap die met Rijndael versleuteld is te ontcijferen, kan je natuurlijk van alle 2^{128} ‘states’ in \mathcal{S} uitrekenen wat hun beeld is onder ϵ . In de praktijk is dit natuurlijk vrijwel onhaalbaar. Ook kan je proberen door van een paar ‘states’ het beeld uit te rekenen, de sleutel $\kappa \in \mathcal{S}$ te achterhalen.

Maar die wordt tijdens het versleutelen zo sterk meegecodeerd, dat dit niet realistisch lijkt. Kortom, men gaat er van uit dat we hiermee een heel veilig systeem in handen hebben.

6.3 Opgaven

1. Hoeveel symbolen zou je met de voor de UPC versie A genoemde regels als een ‘linkercijfer’ kunnen gebruiken?
2. Leg uit dat de voor de cijfers in UPC versie A gebruikte code *wel* een leesfout kan ontdekken, maar niet corrigeren. (Waarom is bijvoorbeeld 0111111 fout? Als er precies één foutje in zit, wat kan je dan zeggen over een mogelijke correctie?)
3. Wat zijn de lengte, de dimensie en de minimale afstand voor de code

$$C = \{(a, b, c, d, e, f) \in \mathbb{F}_2^6 \mid a + b + c = 0 = d + e + f = a + f\}?$$

4. Ga van elk van de volgende uitdrukkingen na, of het elementen zijn van de Hammingcode H , en zo niet, welke coördinaat je moet veranderen om er een element van H van te maken:

(a) $x^4 + x^5 + x^6$

(b) $x^4 + x^6$

(c) x

(d) $1 + x + x^2 + x^5$.

5. Stel dat we de Hammingcode gebruiken, en een bericht $b := v_0 + v_1x + v_2x^2 + \dots + v_6x^6$ ontvangen.

Bepaal met Maple `Rem(x^n*b, x^3+x+1, x) mod 2` voor $n = 0$. Als er in b alleen op de plek x^5 een fout was opgetreden, welk antwoord zou je hier dan krijgen? Ga na dat op dezelfde manier $n = 1$ kan worden gebruikt om een fout op de plek x^4 te vinden, en algemener $n = k$ voor een fout op plaats x^{5-k} . Hoe zit dat met $n = 6$?

6. Maak een lijstje van alle 16 elementen van de Hammingcode H , en controleer daarmee dat inderdaad elk tweetal op minimaal drie plaatsen verschilt.
7. Gebruik `alias(x=RootOf(Z^3+Z+1))` en dan verder regels als bijvoorbeeld `simplify((x+1)^5*x) mod 2`; om rekenen in \mathbb{F}_8 mogelijk te maken. Wat is bijvoorbeeld $111 * 011$, als we schrijven $a_0a_1a_2 = a_0 + a_1x + a_2x^2$?
8. Gebruik $\mathcal{P}_3(\mathbb{F}_8)$ om een (Reed-Solomon) binaire code van lengte 24 en dimensie 12 te construeren. Probeer met behulp van Maple de minimale afstand van deze code te vinden.

9. Ga na dat $x^8 + x^4 + x^3 + x + 1$ irreducibel is in $\mathbb{F}_2[x]$ en concludeer dat $F := \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ een lichaam is. Leg hiermee uit dat $\lambda : F \rightarrow F : \lambda(f) := f^{2^{54}}$ elk element $\neq 0$ uit F naar z'n inverse stuurt, en dat $\lambda \circ \lambda$ de identiteitsafbeelding is.

10. Gebruik

```
Powmod(x^4+x^3+x^2+x+1,-1,x^8-1,x)mod 2}
```

om een inverse van de in Rijndael gebruikte afbeelding **tb** te vinden.

11. Probeer de *orde* van de in Rijndael gebruikte afbeelding **tb** te bepalen. Anders gezegd, hoe vaak moet je die afbeelding op een gegeven byte toepassen om de oorspronkelijke byte weer terug te krijgen?
12. Versleutel de boodschap bestaande uit 128 nullen in Rijndael, waarbij je als sleutel eveneens een rij van 128 nullen neemt.
13. Geef een formule voor het decodeeralgoritme in Rijndael.

7 Eindopdrachten

Ter afsluiting van dit college dien je in groepen van (maximaal) twee personen een verslag over één van de hier volgende onderwerpen te schrijven. Dat maak je met behulp van \LaTeX en het dient tussen de 4 en de 10 pagina's lang te zijn. Probeer een voor medestudenten begrijpelijke tekst te schrijven, en geef duidelijk aan welke bronnen erbij zijn geraadpleegd.

7.1 de exponent van $(\mathbb{Z}/N\mathbb{Z})^*$

De kleinste $e \geq 1$ met de eigenschap, dat elke eenheid $a \bmod N \in (\mathbb{Z}/N\mathbb{Z})^*$ voldoet aan $(a \bmod N)^e = (1 \bmod N)$, heet de *exponent* van $(\mathbb{Z}/N\mathbb{Z})^*$. We noteren deze als $\epsilon(N)$.

Waarom bestaat er eigenlijk zo'n kleinste positieve $e \geq 1$? Wat is het verband met $\phi(N)$?

Bereken $\epsilon(N)$ voor een aantal kleine getallen N . Probeer op grond hiervan een formule te bedenken voor $\epsilon(p^n)$ waarin p een priemgetal is. Lukt het om deze formule (eventueel in speciale gevallen, zoals $n = 2$) te bewijzen?

Probeer uit te leggen waarom je in de theorie over het worteltrekken modulo N in ons dictaat in plaats van $\phi(N)$ ook wel $\epsilon(N)$ had kunnen nemen. Kan je voorbeelden vinden waarin $\epsilon(N)$ veel kleiner is dan $\phi(N)$?

7.2 Carmichaelgetallen

In ons dictaat staat een manier waarmee je van een gegeven getal, als je er tenminste de factorisatie in priemfactoren van kent, snel kan nagaan of het een Carmichaelgetal is. Dit kan je gebruiken om heel veel zulke getallen te construeren.

Eerste constructie: Stel dat $n > 0$ de eigenschap heeft dat zowel $6n + 1$ als $12n + 1$ als $18n + 1$ priemgetallen zijn. Leg uit dat daaruit volgt dat $(6n + 1)(12n + 1)(18n + 1)$ een Carmichaelgetal is. Probeer hiermee een aantal Carmichaelgetallen te vinden.

Tweede constructie: We moeten kwadraatvrije getallen N maken met de eigenschap, dat voor elke priemdelers p van N geldt dat $p - 1$ een deler is van $N - 1$. Dat doen we door vantevoren een getal L te kiezen, te zorgen dat we uitsluitend priemgetallen p bekijken met de eigenschap dat $p - 1$ een deler is van L , en we zoeken dan producten van zulke priemdelers zodanig dat dit product van de vorm $N = 1 + kL$ is voor zekere k . Leg uit dat als zo'n product kwadraatvrij is, dan is het een Carmichaelgetal.

Voer dit uit met $L = 36$. Welke priemenvind je? Welke producten van onderling verschillende p 's hiervan zijn van de gevraagde vorm?

Om heel veel Carmichaelgetallen te maken met deze methode, heb je een L nodig waarvoor er een heleboel priemgetallen p zijn met de eigenschap dat $p - 1$ een deler is van L . Vind dit aantal priemgetallen voor een paar getallen L van de vorm $2^a \cdot 3^b \cdot 5^2 \cdot 7^2 \cdot 11$. Je zoekt hier producten die $1 \pmod L$ zijn. Leg uit dat je daarom de priemgetallen die je hebt en die bovendien L delen niet kan gebruiken. De resterende verzameling priemenvind we \mathcal{P}_L .

Ga na dat $\mathcal{P}_{144} = \{5, 7, 13, 17, 19, 37, 73\}$. Hoeveel Carmichaelgetallen levert dit op?

Stel dat \mathcal{P}_L een deelverzameling \mathcal{G}_L bevat die de eigenschap heeft, dat elk element in $(\mathbb{Z}/L\mathbb{Z})^*$ te schrijven is als een product $\prod (p_i \pmod L)$, waarin de p_i onderling verschillende elementen uit \mathcal{G}_L zijn. Dan levert de volgende constructie een heleboel Carmichaelgetallen op: kies een willekeurig stel priemenvind uit \mathcal{P}_L die geen van alle in \mathcal{G}_L zitten. Hun product modulo L is dan een element van $(\mathbb{Z}/L\mathbb{Z})^*$, en de inverse daarvan is modulo L een product van priemenvind uit \mathcal{G}_L . Het getal dat ontstaat door het product van de gekozen priemenvind met het product van de bijbehorende priemenvind uit \mathcal{G}_L te vermenigvuldigen, is dan een Carmichaelgetal.

Probeer in termen van het aantal priemgetallen in \mathcal{P}_L en \mathcal{G}_L aan te geven hoeveel Carmichaelgetallen je zo bij een gegeven L vindt. Geef ook voorbeelden bij verschillende L .

7.3 AKS-priemtesten

Zoek met behulp van internet een precieze versie op van het AKS-algoritme. Dus, eentje waarin staat waar de gebruikte exponent r aan moet voldoen, en welke a 's je precies moet gebruiken. Ga na hoe groot N minimaal ongeveer moet zijn om ervoor te zorgen dat het aantal rekenstappen bij AKS onder dat van een naïeve priemtest blijft.

Onderzoek wat er gebeurt wanneer je AKS in Maple probeert te implementeren.

7.4 Carmichaelpolynomen

We hebben Carmichaelgetallen gedefinieerd als getallen N waarvoor geldt dat ze niet priem zijn en toch geldt $(a \pmod N)^N = (a \pmod N)$ voor elke a . De exponent N hier is precies het aantal elementen van de ring $\mathbb{Z}/N\mathbb{Z}$ waarin we werken. Een analoge definitie zou je kunnen geven in de polynoomring $\mathbb{F}_p[X]$: een monisch polynoom f van graad $d \geq 1$ noem je een Carmichaelpolynoom

als f niet irreducibel is, terwijl toch geldt $(a \bmod f)^{p^d} = a \bmod f$ voor elk polynoom $a \in \mathbb{F}_p[X]$.

Maak voorbeelden, en probeer net zoals we dat voor Carmichaelgetallen gedaan hebben ook theorie over Carmichaelpolynomen te ontwikkelen.

7.5 worteltrekken modulo p^n

We hebben tijdens het college voor het behandelen van het rsa-systeem theorie gezien waarmee uit $(a \bmod N)^k$, als k een grootste gemene deler 1 heeft met $\phi(N)$, weer $(a \bmod N)$ kan worden teruggevonden. Als $N = p^n$ en k is geen veelvoud van het priemgetal p , dan is er nog een andere manier om k -de machts wortels te trekken.

Ten eerste, als $(a \bmod p^n)^k = b \bmod p^n$, laat zien dat dan ook geldt $(a \bmod p)^k = b \bmod p$. Anders gezegd: wil $b \bmod p^n$ een k -de macht zijn, dan moet zeker gelden dat $b \bmod p$ een k -de macht is.

Omgekeerd, stel dat we van $b \bmod p$ al een k -de machts wortel $a \bmod p$ gevonden hebben. Probeer dan door bij a een geschikt veelvoud van p op te tellen, hier een k -de machts wortel van $b \bmod p^2$ van te maken. Vervolgens kan je proberen bij de nieuw gevonden ‘wortel’ een geschikt veelvoud van p^2 op te tellen zodat het een k -de machts wortel van $b \bmod p^3$ wordt, enzovoort.

Leg uit dat dit werkt, en geef een (Maple)-algoritme ervoor.

Kan je voor k oneven en niet deelbaar door 5 ook een algoritme maken dat k -de machts wortels modulo 10^n trekt?

Ook het geval $k = p$ kan je op deze manier beschouwen, door te beginnen met een k -de machts wortel modulo p^2 of modulo p^3 , en die vervolgens om te werken tot een wortel modulo p^n .

Tenslotte: de hier beschreven methoden werken veel algemener dan alleen maar voor k -de machts worteltrekken. Namelijk, als $f(X) \in \mathbb{Z}[X]$ een (monisch) polynoom is en $a \in \mathbb{Z}$ zo dat $f(a) \bmod p = 0 \bmod p$, dan kan je onder ‘gunstige’ omstandigheden hieruit een nulpunt $a + kp \bmod p^2$ van $f(X)$ modulo p^2 vinden en zo verder modulo hogere machten p^n . Probeer dit in meer detail uit te werken.

7.6 button madness

Het spel ‘button madness’ wordt gespeeld op een $n \times n$ ruitjesbord, waarop sommige velden wit zijn en de overige zwart. De bedoeling van het spel is dat alle velden wit worden. Hiertoe kan je een vakje aanwijzen, en dan verandert dit vak plus de onder-, boven-, linker- en rechterbuur ervan van kleur. Als je een vak aan bijvoorbeeld de bovenrand aanwijst, dan verspringt ook het vak

aan de onderrand in dezelfde kolom van kleur, enzovoort. Bij elk aangewezen vakje, als tenminste $n \geq 3$, veranderen dus 5 vakken van kleur.

Met zoekprogramma's zijn op internet wel versies van dit spel te vinden. Onze vraag is, te onderzoeken of het bij gegeven n inderdaad wel mogelijk is om de stand met alle vakken wit te bereiken.

Daartoe maken we een wiskundig model van het spel. Neem de ring $R_n := \mathbb{F}_2[X, Y]/(X^n - 1, Y^n - 1)$. Schrijf $x := X \bmod (X^n - 1, Y^n - 1)$ en $y := Y \bmod (X^n - 1, Y^n - 1)$. Elke $f \in R_n$ is dus te schrijven als een som van termen $x^i y^j$ met $0 \leq i, j \leq n - 1$.

Als we de vakjes op ons speelbord weergeven door paren (i, j) met $0 \leq i, j \leq n - 1$, waarbij $i = 0, 1, \dots, n - 1$ aangeeft in welke kolom het vakje ligt en evenzo geeft j het rijnummer aan, dan kan je een stand van het bord weergeven als een $f \in R_n$, waarbij f een term $x^i y^j$ bevat precies dan als (i, j) een zwart vakje op het bord is.

Leg uit dat x en y eenheden zijn in R_n , en dus dat x^{-1} en y^{-1} bestaan. Gegeven een speelstand $f \in R_n$, betekent het selecteren van (a, b) op het bord om samen met z'n 4 burens van kleur te worden veranderd, dat de stand f wordt overgevoerd in $f + x^a y^b u$, waarin $u = 1 + x + x^{-1} + y + y^{-1}$.

Leg uit dat elke stand in de geheel witte stand kan worden overgevoerd precies dan, als $u \in R_n^*$.

Probeer met behulp van de computer voor een aantal waarden n te bepalen of u een eenheid is.

7.7 factoriseren met $x \bmod (x^2 + ax + 1)$

Een nadeel van de Pollard $p-1$ methode is, dat deze vooral werkt om getallen N te factoriseren die een priemdeeler p hebben zodat $p-1$ een product van allemaal kleine priemgetallen is. We geven hier een variant die ook voor bepaalde andere getallen N werkt.

Neem de ring $(\mathbb{Z}/N\mathbb{Z})[X]/(X^2 + \bar{a}X + 1)$, voor een $\bar{a} = a \bmod N$ uit $\mathbb{Z}/N\mathbb{Z}$. Schrijf $x := X \bmod (X^2 + \bar{a}X + 1)$. Ga na dat dit een eenheid is in deze ring. Een hoge macht x^k is dan te schrijven in de vorm $x^k = \bar{a}_k x + \bar{b}_k$ voor zekere $a_k, b_k \in \mathbb{Z}$. Een mogelijke factor van N is dan $\text{ggd}(a_k, N)$. En levert dit niks op dan kan je a en eventueel k variëren.

Test dit algoritme. Probeer met name wat Mersennegetallen die niet priem zijn, en een aantal getallen van de gedaante $(10^{2n+1} - 1)/9$ met $n > 2$ te ontbinden. Kan je, zoals dat bij Pollard $p-1$ het geval is, een 'voorkeur' voor een bepaald soort priemdelers ontdekken? Let daarbij ook op $p+1$.

Probeer, zoals we dat voor Pollard $p-1$ gedaan hebben, uit te leggen waarom dit algoritme enige kans van slagen heeft.

7.8 opnieuw: de kwadratische zeef

Bij de kwadratische zeef methode voor het factoriseren van getallen zijn tijdens het college alleen maar voorbeelden met een heel kleine ‘factorbasis’ behandeld.

Probeer zelf meer ‘serieuze’ voorbeelden te maken. Kies daarvoor als factorbasis de verzameling van alle priemgetallen kleiner dan 100. Lukt het om daarmee bijvoorbeeld $2^{64} + 1$ en $2^{128} + 1$ te factoriseren?

Een mogelijk probleem van deze methode is, dat je soms te weinig kwadraten modulo N hebt die je volledig in termen van je factorbasis kan factoriseren. Daar kan je wat aan doen door meer polynomen te gebruiken. De standaardmanier gebruikte alleen $f(x) = (x + s)^2 - N$, waarin het gehele getal s in de buurt van de wortel uit N ligt. Maar je kan ook een s_2 in de buurt van $\sqrt{(2N)}$ nemen, en dan kijken naar $f_2(x) = (x + s_2)^2 - 2N$, of iets analoogs met een s_3 in de buurt van de wortel uit $3N$, enzovoort. Experimenteer met dit idee...

7.9 de factorisatie van $x^p - 1$

Neem $p > 2$ een priemgetal en $f = X^p - 1$. Voor elk priemgetal q kunnen we dan f ontbinden in $\mathbb{F}_q[X]$.

Doe dit voor $p = 3$ en een aantal priemmen q . Wat zijn de graden van de irreducibele factoren die je krijgt? Zie je regelmaat? Kan je een verklaring bedenken?

Probeer dezelfde vraag voor een veel grotere priem p te beantwoorden. Hint: bereken voor de priemmen $q \neq p$ die je gebruikt de orde van q mod p in $(\mathbb{Z}/p\mathbb{Z})^*$.

7.10 nulpunten vinden in \mathbb{F}_p

Gegeven een monisch polynoom $f \in \mathbb{F}_p[X]$. Als f een nulpunt $a \in \mathbb{F}_p$ heeft, dan is $X - a$ een deler van f en dus (ga na!) van $\text{ggd}(f, X^p - X)$. Als f precies één nulpunt in \mathbb{F}_p heeft, dan vinden we dat op deze manier. Dit is moeilijker wanneer f meerdere nulpunten heeft.

Geef van beide bovenstaande mogelijkheden voorbeelden, ook voor vrij grote priemgetallen p .

Om ook in het geval van meerdere nulpunten deze te vinden, kan je $X^p - X$ als een product schrijven. Bijvoorbeeld als

$$X^p - X = (X + a) \cdot ((X + a)^{(p-1)/2} - 1) \cdot ((X + a)^{(p-1)/2} + 1),$$

voor een willekeurige $a \in \mathbb{F}_p$. Probeer uit te leggen waarom deze factorisaties correct zijn.

Vervolgens kan je een nulpunt van f proberen te vinden door hierboven een factor te kiezen en daarvan de ggd met f te bepalen. Schrijf hiervoor een algoritme en test dit voor een aantal grote priemmen p en sommige f die een product zijn van lineaire factoren. Bijvoorbeeld

$$f = (X - 1/4)(X - 1/9)(X - 1/16),$$

waarbij je p ongeveer 10^{10} neemt...

7.11 repunits

‘Repunits’ zijn getallen van de vorm $r_n := (10^n - 1)/9$. Leg uit dat dit gehele getallen zijn; kan je verklaren waaraan ze hun naam te danken hebben? Geef zoveel mogelijk voorwaarden waaronder r_n geen priemgetal is. Maak daarvoor in het bijzonder een tabel en zoek naar regelmaat in die tabel. Geef bijvoorbeeld alle n waarvoor r_n deelbaar is door 7. Dezelfde vraag met ‘7’ vervangen door 3, door 11, door 13, en door een ander priemgetal p met $13 < p < 30$.

Probeer een aantal r_n te vinden die priem zijn.

7.12 het voetbalpool probleem

Neem een $n > 1$ vast, en bekijk rijtjes $a = (a_1, \dots, a_n)$ waarin elke $a_i \in \mathbb{F}_3$. Zo’n rijtje vatten we op als een mogelijke uitslag van n gespeelde (voetbal)wedstrijden: $a_i = \bar{0}$ betekent dat bij de i -de wedstrijd de thuisclub verliest, $a_i = \bar{1}$ houdt in dat bij die i -de wedstrijd gelijk wordt gespeeld en $a_i = \bar{2}$ betekent winst voor de thuisclub in de i -de wedstrijd.

We willen nu vantevoren een heleboel zulke mogelijke rijtjes opschrijven, en wel zo, dat wat vervolgens de uitslagen ook zijn, we hebben altijd wel een rijtje opgeschreven die hoogstens op één plek afwijkt van de echte uitslag. De vraag is dan, hoeveel zulke rijtjes we minimaal moeten opschrijven om hieraan te voldoen.

Probeer voor een aantal getallen n dit minimale aantal te vinden. In het tijdschrift Nieuw Archief voor Wiskunde, 4de serie Deel 16, pp. 173–177 vind je veel meer informatie over dit probleem.

7.13 de binaire Golay code

Schrijf in $\mathbb{F}_2[X]$ het polynoom $X^{23} - 1$ als een product $(X - 1) \cdot m_1 \cdot m_2$, voor zekere monische polynomen m_1 en m_2 die beide graad 11 hebben. De

binaire Golay code kan beschreven worden als de deelverzameling (het ideaal) in $\mathbb{F}_2[X]/(X^{23} - 1)$ bestaande uit alle $f \cdot m_1 \bmod (X^{23} - 1)$, waarin f de hele $\mathbb{F}_2[X]$ doorloopt.

Probeer uit te vinden hoeveel elementen deze code heeft. Probeer ook elementen $g \bmod (X^{23} - 1)$ in deze code te vinden met $1 \leq \text{graad}(g) \leq 22$ met de eigenschap dat in g zo weinig mogelijk machten van X voorkomen. Wat is het minimale aantal machten dat hierbij mogelijk is?

7.14 Rijndael

In ‘Rijndael’ is het van belang, dat de sleutel $\kappa \in \mathcal{S}$ geheim is. Onderzoek, in hoeverre het mogelijk is om uit het feit dat je van elke input-boodschap de Rijndael-output kan bepalen, toch die sleutel te vinden.

Implementeer verder het systeem, en schrijf in het bijzonder ook een decodeeralgoritme (uitgaande van het gegeven dat je de sleutel kent).

Index

A

Abel 16
abelse groep 16
Adleman, Leonard M. 38
Advanced Encryption Standard 81
afgeleide 61
aftrekken 6
Agarwal, M 56
AKS-algoritme 56
Alford, W. (Red) 50
ASCII 72

B

baby-step giant-step 23
Bahr, Friedrich 61
basis 74
Berlekamp, E. 46
Berlekamp algoritme 64
binaire code 73
binaire schrijfwijze 22
breuken 4
button madness 88

C

Cantor, D.G. 65
Carmichael, Robert D. 50
Carmichaelgetal 50
Carmichaelpolynoom 87
chinese reststelling 33
cipher challenge 38
code van type $[n, k, d]$ 74
coderen 71
commutatieve groep 16
commutatieve ring 7
complexe getallen 5
cryptografie 38
cyclische code 75

D

Daemen, J. 81
decryptie 38
deling met rest 24

dimensie 74

E

ECM 65
eenhedengroep 16
eenheid 8
eenheidselement 15
eindig lichaam 47
encryptie 38
Euclides 24
Euclidische algoritme 27
Euclidische ring 24
Euler 20
Euler phi functie 39
exponent van $(\mathbb{Z}/N\mathbb{Z})^*$ 86

F

factorbasis 67
factoring 12
Fermat 20
Fermat's kleine stelling 49
Fermat-test 49
Franke, Jens 61

G

gehele getallen 5
Golay code 92
graad 10
Granville, A. 50
groep 15
grootste gemene deler 26

H

Hamilton, Sir W.R. 8
Hamming, R.W. 74
Hamming code 76
Hammingnorm 74

I

ideaal 11
ideaal voortgebracht door 11
inverse 9
irreducibel 46

K

Kayal, N 56
kleine stelling van Fermat 49
Kleinjung, Thorsten 61
kopcoëfficiënt 46
kwadraatvrij 51
kwadratische zeef 65

L

Lehmer, D.H. 54
lengte 73
Lenstra jr., H.W. 65
lichaam 8
lineaire binaire code 74
lineaire ruimte 74
Lucas, E. 54
Lucas-Lehmer test 54

M

matrixring 7
meervoudige factor 62
Mersenne, M. 53
Mersennegetal 53
Mersennepriem 53
Miller, Gary L. 53
Miller-Rabin test 53
minimale afstand 74
monisch 46
Montgomery, P.L. 65
MPQS 65

N

natuurlijke getallen 5
NFS 65
niet-commutatieve groep 16
Nieuw Archief voor Wiskunde 91
nuldeler 10

O

ontbinden 46
optellen 5
optellen modulo I 13
orde 16

P

PGP (pretty good privacy) 38

Pinch, R.G.E. 50
Pollard, J.M. 65
Pollard $p - 1$ 65
polynoomring 9
polynoomring in meer variabelen 10
Pomerance, C. 50
productring 32
pseudopriem 50

Q

quaternionen 8
quotientring 12

R

Rabin, Michael O. 53
rationale getallen 4
Reed, I.S. 78
Reed-Solomon codes 78
reële getallen 5
repunits 91
te Riele, Herman 60
ring 5
ring zonder nuldelers 10
Rivest, Ronald L. 38
Rijmen, V. 81
Rijndael 81
RSA 38
RSA- n 60

S

Saxena, N 56
scalaire vermenigvuldiging 74
Selfridge, John 59
Shamir, Adi 38
Shanks, D. 23
Solomon, G. 78
sterke pseudopriem 53
streepjescode 71

T

tegengestelde 6

U

Universal Product Code 71
UPC versie A 71

V

vermenigvuldigen 5
vermenigvuldigen modulo I 13
voetbalpool probleem 91

W

Wagstaff, Sam 59
Wang, Xiaoyun 38
worteltrekken modulo N 40

Z

Zassenhaus, H. 65