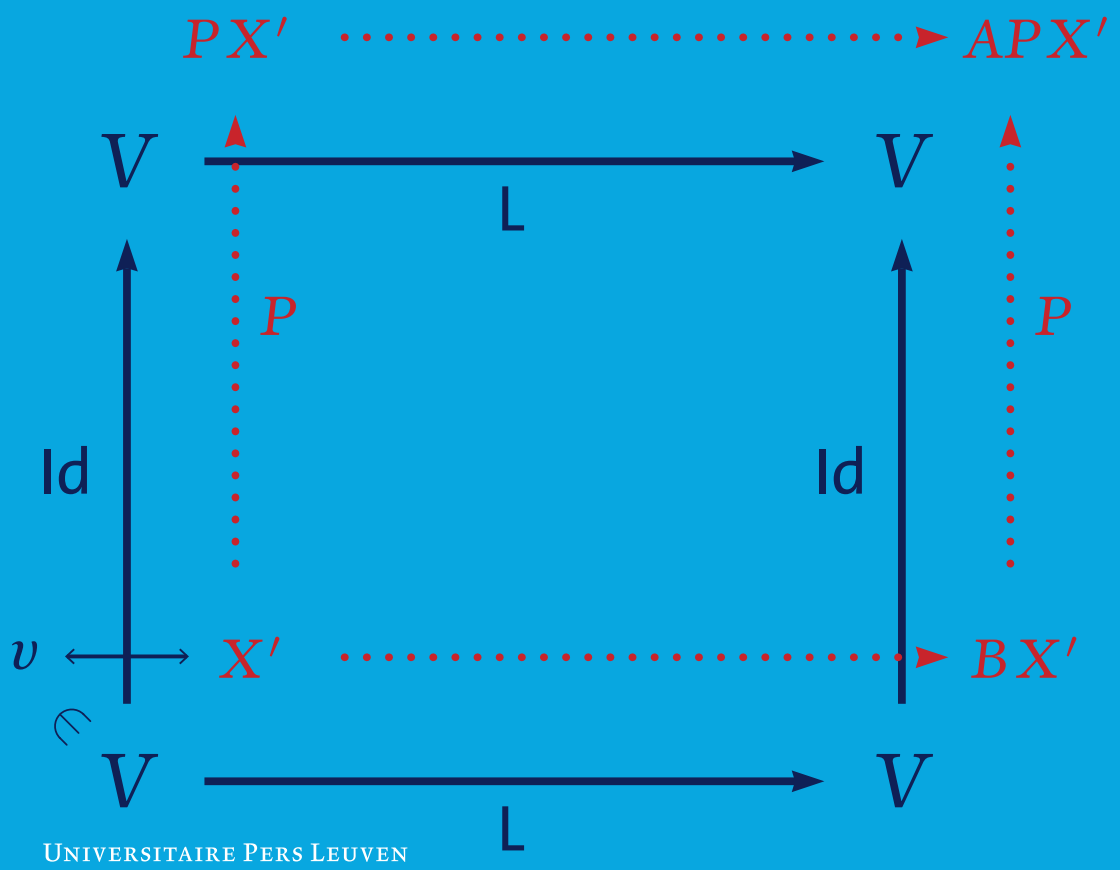
PAUL IGODT & WIM VEYS

lineaire algebra

Derde editie



UNIVERSITAIRE PERS LEUVEN

Extra online module bij Lineaire algebra, derde editie, Paul Igodt, Wim Veys, ISBN 9789462703148, © Universitaire Pers Leuven 2022

Eerste druk, 2011
Tweede herwerkte druk, 2015
Derde herwerkte druk, 2022

ISBN 978 94 6270 314 8
D / 2022/ 1869 / 4
NUR: 921

Ontwerp omslag: André Klijsen

HOOFDSTUK 9

Toemaatjes: uitbreidingen

9.1 Lineaire codes en lineaire afbeeldingen

In deze paragraaf wordt gewerkt met vectorruimten met coëfficiënten in een eindig veld $\mathbb{Z}_p$. Indien je hiermee niet vertrouwd bent, raden we je aan eerst het toemaatje 'Wat is een veld?' bij Hoofdstuk 3 door te nemen.

Voor een geheel getal a noteerden we in die paragraaf 3.7 de restklasse van a modulo p als $\overline{a}$. Voor eenvoud van notatie noteren we hier die restklasse gewoon als a.

9.1.1 Situering

In een studiegebied als codeertheorie houdt men zich ondermeer bezig met het omzetten van informatie (denk: boodschappen) in een meer formele taal of vorm, met de bedoeling hiermee op de een of andere manier 'toegevoegde waarde' te creëren. Deze toegevoegde waarde kan verband houden met het afschermen van de informatie voor ongewenste toegang, met het authenticeren of identificeren van een persoon of een product, of met het beschermen van de informatie tijdens allerlei transmissies (denk: uitlees- en schrijfoperaties, verzenden van signalen doorheen een bepaald medium, ...).

De toepassing die we hier belichten beoogt voornamelijk het (technisch) beschermen van de informatie tegen fouten bij transmissie. Denk bijvoorbeeld aan het doorsturen van een foto genomen door een ruimtetuig op de planeet Mars, het uitlezen door een zeer snelle processor die miljoenen operaties per seconde uitvoert van het geheugen van een computer, of het afspelen van een CD of DVD. De aard van de foutjes bij de transmissie kan afhangen van de specifieke situatie die zich voordoet; bijvoorbeeld kan door een kras op de CD alle informatie over een strook van een millimeter beschadigd zijn.

Richard Hamming (Chicago, 1915 - Monterey, 1998)

Hamming studeert aan de Universiteiten van Chicago en Nebraska en behaalt in 1942 een doctoraat in de wiskunde aan de Universiteit van Illinois te Urbana-Champaign. In 1945 wordt hij een medewerker in het Manhattan Project, een geheime operatie in Los Alamos met het oog op de ontwikkeling van een atoombom. Kort daarna reeds, in 1946, gaat hij

aan de slag in het departement wiskunde van de befaamde Bell Telephone Laboratories in de staat New Jersey. Daar leert hij Claude Shannon kennen, de grondlegger van de moderne informatietheorie.

Hammings naam zal voor altijd verbonden zijn aan de theorie van foutendetecterende en foutencorrigerende codes, een domein waarin hij baanbrekend onderzoek verrichtte. Begrippen zoals de Hammingafstand en een familie codes die naar hem genoemd worden, behoren tegenwoordig tot de basisvorming aan de universiteit. Hammingcodes zijn ook nauw verbonden met de correcte werking van computergeheugens.

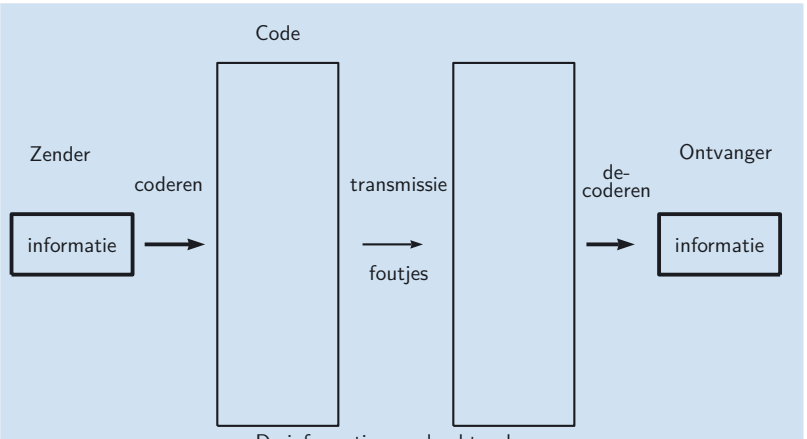
In 1968 ontving Hamming de befaamde Alan Turing Award*, vandaag gezien als de hoogste onderscheiding in de informatica.*

Het is hierbij belangrijk een onderscheid te maken tussen foutendetectie enerzijds, en foutencorrectie anderzijds. Hammingcodes zijn op deze beide aspecten actief. Ze worden nog steeds frequent gebruikt in diverse informatica- of computersystemen, onder andere bij het uitlezen van en wegschrijven in diverse computergeheugens. Hamming mag terecht gezien worden als een van de grondleggers van de algebraïsche coderingstheorie. Zonder deze theorie zou de betrouwbaarheid van vele communicatie- en computersystemen heel wat te wensen overlaten.

Het gezond-verstand-idee dat we aan het werk zien bij coderingssystemen voor foutendetectie en -correctie bestaat erin de feitelijke informatie aan te vullen met extra gegevens. Het lijkt even alsof je overtollige zaken toevoegt, en dus de communicatie vertraagt of extra belast, maar in werkelijkheid kan een intelligente aanpak ook effectieve baat opleveren. Je kunt het vergelijken met het in een context plaatsen van een woord of een zin. Als er dan toch een foutje optreedt, zal de betekenis van de boodschap sneller kunnen gerecupereerd worden, juist omwille van die context.

Een voorbeeld helpt allicht: als door een minieme fout in de communicatie het woord "groetjes" omgezet wordt in "groefjes", dan kan er weliswaar een glimlach ontstaan, maar ook twijfel over wat juist bedoeld werd. Staat er echter "Veilig aangekomen, vele groefjes", dan zal de ontvanger ongetwijfeld de boodschap ondubbelzinnig begrijpen.

Schematisch kan een en ander als volgt worden voorgesteld.



De informatie-overdrachtcyclus

9.1.2 Een inspirerend voorbeeld

Het hierna volgend voorbeeld ontleenden we aan Jack H. van Lint, die baanbrekend werk verrichtte in de codeertheorie. Het is een dermate mooi en sprekend voorbeeld, dat we bij wijze van spreken van Lint zelf (zie [7]) aan het woord laten.

Veronderstel dat we een verhaal van $600\,000$ *tekens willen opslaan; denk aan een boek met* 200 *virtuele pagina's die elk* 3000 *tekens bevatten. Op elke pagina wordt een deeltje van het verhaal opgeslagen.*

Om dit uit te voeren stellen we de letters a *tot en met* z *van ons alfabet voor door de getallen* 1 *tot en met* 26. *Een* spatie *wordt voorgesteld door* 0, *en we gebruiken de gehele getallen* 27, 28, 29 *en* 30 *respectievelijk voor de leestekens* punt, komma, puntkomma *en* dubbelpunt.

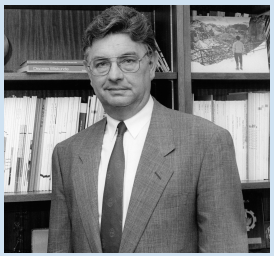
Een technicus informeert ons dat de technologie om de tekens op te slaan helaas niet optimaal werkt; meer zelfs, elk teken op een (virtuele) pagina wordt met een kans gelijk aan 0.1% *fout weggeschreven. Gemiddeld gezien zullen er dus op elke pagina drie fouten ontstaan, al bij al geen puik resultaat.*

We onderzoeken of en hoe we met behulp van een of ander codeersysteem de correctheid van deze opdracht kunnen verbeteren. In een eerste poging verdelen we de rij van alle tekens in groepjes van vier. Bij elk groepje van vier tekens voegen we twee extra tekens toe. Zo verkrijgen we een groepje van zes tekens uit $\mathbb{Z}_{31}$, *dat we voorstellen als* $(a_0, a_1, a_2, a_3, a_4, a_5)$; *hierbij zijn* a_0 *en* a_1 *de twee extra tekens en zijn* a_2, a_3, a_4, a_5 *de*

vier oorspronkelijke tekens. Om a_0 en a_1 te bepalen spreken we af dat, voor elk groepje van vier tekens a_2, a_3, a_4, a_5 in $\mathbb{Z}_{31}$, moet voldaan zijn aan

$$\begin{cases} a_0 + a_1 + a_2 + a_3 + a_4 + a_5 = 0 \\ a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 = 0. \end{cases}$$

Jacobus Hendricus (Jack) van Lint (Bandoeng (Indonesië), 1932 - Eindhoven, 2004)



©Familie van Lint

Jack van Lint studeert wiskunde en natuurkunde aan de Universiteit Utrecht, waar hij in 1955 afstudeert en in 1957 promoveert. Hij wordt reeds op 26-jarige leeftijd hoogleraar aan de Technische Universiteit Eindhoven. Daar werkt hij meer dan veertig jaar, eerst rond getaltheorie, maar later vooral rond combinatoriek en coderingstheorie.

Hij is auteur van heel wat toonaangevende publicaties, waaronder meerdere boeken, vaak echte standaardwerken voor het betreffende vakgebied, zoals Introduction to Coding Theory *en* A course in combinatorics.

Als expert in codeertheorie en adviseur bij het Philips NatLab te Eindhoven is van Lint vanaf 1985 nauw betrokken bij de ontwikkeling van de CD.

Jack van Lint was rector van de T.U. Eindhoven (1991-1996) en ijverde, als enthousiast sporter, voor goede sportvoorzieningen. Zelf was hij een getalenteerd zwemmer. Nu draagt het universitair zwembad zijn naam en is er nog jaarlijks een internationaal Prof. van Lint voetbaltornooi. Hij is in binnen- en buitenland geëerd met een groot aantal onderscheidingen. Hij was vier keer uitgenodigd spreker op het vierjaarlijkse International Congress of Mathematicians *(ICM), een waarlijk uitzonderlijke prestatie.*

Een voorbeeld helpt om dit goed te begrijpen. Veronderstel dat we het woord "code" aantreffen in de tekst die we zullen opslaan. Dit betekent het viertal $(a_2, a_3, a_4, a_5) = (3, 15, 4, 5)$. We zoeken nu a_0 en a_1 zodat voldaan is aan de gegeven vergelijkingen. Uit de tweede vergelijking berekenen we dat $a_1 = 1$, waarmee we via de eerste vergelijking besluiten dat $a_0 = 3$. Het woord "code" is dus omgezet in een nieuw, gecodeerd woord "cacode".

Veronderstel nu dat we in de op die manier gecodeerde tekst een woord "xgfopt" aantreffen. Wanneer we de eerste twee tekens vergeten, lezen we het Nederlandse woord "fopt". Dit is een bestaand Nederlands woord, maar is het ook een correct bedoeld woord in onze tekst? In getallen omgezet staat dit codewoord voor het zestal $(24, 7, 6, 15, 16, 20)$. Deze getallen tellen op tot $78 \equiv 26$ modulo 31. Ze voldoen dus niet aan de eerste

vergelijking en bijgevolg kan dit zestal onmogelijk het resultaat zijn van het coderen van een groepje van vier tekens. Omdat de kans zeer groot is dat er slechts in één teken een fout is opgetreden, ontstaat een sterk vermoeden dat precies één van de tekens a_i moet vervangen worden door $a_i + 5$, opdat de eerste vergelijking zou voldaan zijn.

De fout, e genoteerd, in het teken a_i is dus $e = -5$. In de tweede vergelijking wordt elk teken a_i met i vermenigvuldigd, en dus wordt de fout e omgezet in $i.e = -5i$. Anderzijds, als we de tweede vergelijking voor dit zestal nagaan, vinden we

$$a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 = 7 + 12 + 45 + 64 + 100 = 228 \equiv 11 \text{ modulo } 31.$$

In $\mathbb{Z}_{31}$ moet dan $-5i = 11$, of dus $i = -11/5$, wat neerkomt op $i = 4$. Het is dus het teken a_4 dat fout is; in plaats van $a_4 = 16$ is de correcte waarde $a_4 = 21$. We vinden dus niet het woord "fopt", maar het woord "fout" wanneer we de correctie doorvoeren. En inderdaad, ook dit is een correct Nederlands woord.

We waren dus in staat om een fout die optreedt in juist één positie te corrigeren. Maar hebben we nu ook echt baat bij heel deze codeer- en decodeeroperatie, die toch vrij veel extra werk vergt? Immers, per vier tekens creëerden we er twee extra, zodat het totaal aantal tekens voor dit verhaal opgelopen is tot maar liefst $900\,000$. Willen we daarenboven op elke virtuele pagina hetzelfde deel van het verhaal opslaan als daarnet, dan moeten er nu niet 3000 maar 4500 tekens op één pagina opgeslagen worden. Dit kan slechts door als het ware een compacter lettertype te gebruiken. Op de koop toe bereikt ons nu een bericht van de technicus dat zijn technologie helaas meer fouten maakt met dergelijk compact opslagformaat; er blijkt zelfs een tweemaal zo grote kans op een fout te zijn bij elk teken. Gemiddeld zal elke (virtuele) pagina dus maar liefst negen fouten tellen. Was onze codeerinspanning nu wel echt de moeite waard?

Het is natuurlijk het eindresultaat na decodering (en bijgevolg na de eventuele correctie van een foutje) dat telt. Dit noopt tot een nieuwe berekening. Elk zestal tekens wordt correct gedecodeerd zolang het hoogstens één fout bevat. De kans dat er in zo een zestal meer dan één fout is opgetreden, is gegeven als

$$\textit{de kans op twee fouten} \;+\; \textit{de kans op drie fouten} \;+\ldots,$$

wat gelijk is aan

$$\sum_{j=2}^{6} \left(\begin{array}{c} 6 \\ j \end{array} \right) (0.002)^j (0.998)^{6-j} \approx 0.00006.$$

Dus ongeveer 6 op $100\,000$ keer zal een zestal tekens twee of meer fouten bevatten. We kunnen dus verwachten dat dit niet meer dan tien keer zal gebeuren voor het volledige verhaal. Onze codeerinspanning wordt, met andere woorden, beloond met een reductie van het aantal te verwachten opslagfouten tot hoogstens tien voor het volledige verhaal

(in plaats van drie per virtuele pagina)! De inspanning om te coderen en te decoderen, inderdaad een heuse inspanning, drijft de nauwkeurigheid van het uit te voeren werk substantieel op.

Het hier geïllustreerde codeersysteem is een voorbeeld van een lineair codeersysteem. We leggen in de volgende paragraaf uit waarom dat zo is. De extra symbolen die aan een woord worden toegevoegd, worden gevonden door een stelsel eerstegraadsvergelijkingen op te lossen, weliswaar in het veld $(\mathbb{F}, +, .) = (\mathbb{Z}_{31}, +, .)$.

9.1.3 Lineaire codes en lineaire afbeeldingen

In het algemeen, als $(\mathbb{F}, +, \cdot)$ een veld is, is $(\mathbb{F}, \mathbb{F}^n, +)$ een vectorruimte. Eerdere voorbeelden hiervan waren reeds de vectorruimten $(\mathbb{R}, \mathbb{R}^n, +)$ en $(\mathbb{C}, \mathbb{C}^n, +)$. We kunnen die hier nu aanvullen met de nieuwe reeks voorbeelden $(\mathbb{Z}_p, \mathbb{Z}_p^n, +)$, en dit voor elk priemgetal p en elk natuurlijk getal $n \geqslant 1$. In de vorige paragraaf beschouwden we de woorden als 4-tallen uit $\mathbb{Z}_{31}$, met andere woorden als elementen van $(\mathbb{Z}_{31})^4$ dat we liever kort noteren als $\mathbb{Z}_{31}^4$.

De elementen (vectoren) van de vectorruimte $\mathbb{F}^n$ zijn de n-tallen $(x_1, x_2, \ldots, x_n)$ waarbij elke $x_i \in \mathbb{F}$. De vectorruimten $(\mathbb{Z}_p, \mathbb{Z}_p^n, +)$ zijn dus eindige verzamelingen, en precies daarom erg geschikt voor toepassingen in de informatica.

Een vrij natuurlijk model voor het coderen van informatie werkt als volgt. Fixeer een eindig veld $\mathbb{F}$. Een 'blok' informatie of *boodschap* bestaat uit k elementen $c_1, c_2 \ldots, c_k$ van $\mathbb{F}$. Het is dus een vector van $\mathbb{F}^k$. Dit blok wordt gecodeerd door er $n-k$ 'overtollige' elementen $c_{k+1}, \ldots, c_n$ aan toe te voegen, die berekend worden in termen van $c_1, c_2 \ldots, c_k$. Het resultaat is een *codewoord* $c_1, c_2 \ldots, c_n$, met andere woorden een vector van $\mathbb{F}^n$.

Een eenvoudige (maar zoals blijkt nuttige en krachtige) manier om dit concreet te doen bestaat erin $c_{k+1}, \ldots, c_n$ te bepalen als lineaire combinaties van $c_1, c_2 \ldots, c_k$. We belanden zo in de context van lineaire algebra: beschouw de boodschappen en de codewoorden als elementen van respectievelijk de vectorruimten $\mathbb{F}^k$ en $\mathbb{F}^n$, en het coderen als een lineaire afbeelding $\mathbb{F}^k \to \mathbb{F}^n$. De verzameling van codewoorden, de *code* C, is dan het beeld van deze lineaire afbeelding. Merk op dat een dergelijk type lineaire afbeelding automatisch injectief is.

Vooraleer te formaliseren geven we een voorbeeld.

Voorbeeld 9.1 Veronderstel dat we binair werken, dit wil zeggen met het veld $\mathbb{F} = \mathbb{Z}_2$, en met boodschappen van lengte 4. Neem nu de lineaire afbeelding

$$L : \mathbb{Z}_2^4 \to \mathbb{Z}_2^7 : (c_1, c_2, c_3, c_4) \mapsto (c_1, c_2, c_3, c_4, c_2 + c_3 + c_4, c_1 + c_3 + c_4, c_1 + c_2 + c_3).$$

De geassocieerde code C, dit wil zeggen het beeld $\mathrm{Im}(L)$, is dan de vierdimensionale deelruimte $\mathbb{Z}_2^7$ bestaande uit de codewoorden $(c_1, c_2, c_3, c_4, c_5, c_6, c_7)$ die voldoen aan

$$\begin{aligned} c_5 &= c_2 + c_3 + c_4, \\ c_6 &= c_1 + c_3 + c_4, \\ c_7 &= c_1 + c_2 + c_3. \end{aligned}$$

Er zijn twee klassieke manieren om dit voor te stellen in matrixvorm. De code C wordt voortgebracht door de beelden van de standaardbasis van $\mathbb{Z}_2^4$; we schrijven deze vier beelden in de rijen van een matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix},$$

die we een *voortbrengende matrix*van C noemen. De code C is dus de rijruimte van de matrix G. (Merk op dat G de getransponeerde is van de matrix van L ten opzichte van de twee standaardbasissen.)

Anderzijds is C ook de oplossingsverzameling van het homogene stelsel bepaald door de matrix

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

Inderdaad, kijk naar de voorschriften voor c_5, c_6, c_7 in termen van c_1, c_2, c_3, c_4 en besef dat $1 = -1$ in $\mathbb{Z}_2$. Anders gezegd is C dus de nulruimte van de matrix H.

Merk op dat de matrix G 'links begint met een eenheidsmatrix' en de matrix H 'rechts eindigt met een eenheidsmatrix'. Wat is hiervoor de reden?

Codes zoals in het vorige voorbeeld, waarbij de eerste k coördinaten bij het coderen niet veranderen, worden systematische codes genoemd. Maar er is geen reden om ons hiertoe te beperken. We voeren nu eerst algemene lineaire codes in, en komen dan terug op de systematische codes.

Definitie 9.2

Zij $\mathbb{F}$ een eindig veld. Een (n,k)-**lineaire code** over $\mathbb{F}$ is een k-dimensionale deelruimte C van de vectorruimte $(\mathbb{F}, \mathbb{F}^n, +)$. De elementen van C noemen we de **codewoorden**; n noemen we de *lengte* van de codewoorden of van de code, k de *dimensie* van de code, en $\frac{k}{n}$ de *informatieverhouding* van de code. In het bijzonder geval dat $\mathbb{F} = \mathbb{Z}_2$ spreken we van een **binaire code**.

Opmerking 9.3 Laten we eerst iets opmerken over de notaties. De elementen van een lineaire code (de codewoorden) zijn dus n-tallen. Hiervoor gebruiken we klassiek de n-talnotatie; een codewoord ziet er dan uit als $(x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$. Vaak, en zeker wanneer er geen verwarring mogelijk is, zie je ook de notatie $x_1x_2x_3 \ldots x_n$, waarbij de haakjes en de komma's vergeten worden.

Anderzijds zullen we, in de context van matrixvermenigvuldiging, codewoorden meestal moeten interpreteren als kolommatrices (kolommen). Om dit laatste eenvoudiger te kunnen noteren, beschouwen we (consequent met de eerdere notaties) *bij matrixvermenigvuldiging $v \in \mathbb{F}^m$ als een kolom(matrix) en $v^T \in \mathbb{F}^m$ als een rij(matrix).*

Uiteraard geldt bij een (n, k)-lineaire code dat $k \leq n$. In het bijzonder is $\mathbb{F}^k$ op een natuurlijke manier een deelruimte van $(\mathbb{F}, \mathbb{F}^n, +)$ via de natuurlijke inclusie

$$i : \mathbb{F}^k \to \mathbb{F}^n : (x_1, x_2, \ldots, x_k) \mapsto (x_1, x_2, \ldots, x_k, 0, 0, \ldots, 0).$$

Formeel is $\mathrm{Im}(i)$ dus een (n, k)-lineaire code. Meer algemeen, en *dit is een cruciaal inzicht*, geldt dat elke lineaire afbeelding $L : \mathbb{F}^k \to \mathbb{F}^n$ die injectief is (met andere woorden waarvoor $\dim_{\mathbb{F}} \mathrm{Ker}(L) = 0$) een (n, k)-lineaire code $\mathrm{Im}(L)$ over F bepaalt. Het volstaat dus dergelijke injectieve lineaire afbeeldingen L te bepalen. Dit komt op hetzelfde neer als $(n \times k)$-matrices A zoeken met rang k. De code is dan $\mathrm{Im}(L_A)$.

Ook is voor een lineaire afbeelding $L' : \mathbb{F}^n \to \mathbb{F}^\ell$ waarvoor $\dim_{\mathbb{F}} \mathrm{Ker}(L') = k$ die kern natuurlijk een (n, k)-lineaire code. Deze beschouwingen geven aanleiding tot de twee klassieke manieren om een algemene lineaire code te beschrijven door een matrix (als in Voorbeeld 9.1).

Definitie 9.4

Zij $\mathbb{F}$ een eindig veld en C een (n, k)-lineaire code over $\mathbb{F}$. Een **voortbrengende matrix**of **generatormatrix**voor C (Engels: generator matrix) is een $(k \times n)$-matrix G over $\mathbb{F}$ waarvan de rijen C voortbrengen, met andere woorden met C als rijruimte. Een **controlematrix** of **pariteitscontrolematrix** van C (Engels: check matrix, parity check matrix) is een $((n - k) \times n)$-matrix H over $\mathbb{F}$ zodat C de nulruimte is van H.

Merk op dat G en H allebei maximale rang hebben, respectievelijk k en $n - k$. Aan de hand van deze matrices bepalen we als volgt of een vector $v \in \mathbb{F}^n$ al dan niet een codewoord is:

1. $v \in C \Leftrightarrow H \cdot v = 0$;
2. $v \in C$ als en slechts als v het beeld is van een vector (boodschap) $b \in \mathbb{F}^k$ onder de (injectieve) lineaire afbeelding $\mathbb{F}^k \to \mathbb{F}^n$ met als matrix G^T ten opzichte van de

standaardbasissen. Anders gezegd: $v \in C$ als en slechts als er een $b \in \mathbb{F}^k$ bestaat zodat

$$v = G^T \cdot b \quad (\Leftrightarrow \; v^T = b^T \cdot G\,).$$

Definitie 9.5

Zij $\mathbb{F}$ een eindig veld. Een systematische code over $\mathbb{F}$ is een (n,k)-lineaire code over $\mathbb{F}$ met een controlematrix van de vorm

$$H = \begin{pmatrix} * & * & \cdots & * & 1 & 0 & \cdots & 0 \\ * & * & \cdots & * & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ * & * & \cdots & * & 0 & 0 & \cdots & 1 \end{pmatrix} = \begin{pmatrix} A & \mathbb{I}_{n-k} \end{pmatrix} \in \mathbb{F}^{(n-k)\times n}.$$

Hierbij is A een $((n-k)\times k)$-matrix over $\mathbb{F}$.

Voorbeeld 9.6

1. Neem $\mathbb{F} = \mathbb{Z}_2$. Controleer dat $C = \{(0,0,0),(0,0,1),(1,0,0),(1,0,1)\}$ een $(3,2)$-lineaire (binaire) code is.

2. Neem $\mathbb{F} = \mathbb{Z}_5$. Beschouw de matrix

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 2 \\ 1 & 2 & 0 \\ 3 & 4 & 1 \end{pmatrix}$$

en verifieer dat deze matrix rang 3 heeft. De lineaire afbeelding $L_A : \mathbb{Z}_5^3 \to \mathbb{Z}_5^5 : X \mapsto A \cdot X$ induceert de $(5,3)$-lineaire code $\mathrm{Im}(L_A)$ over $\mathbb{Z}_5$.

3. Zij $\mathbb{F}$ een willekeurig eindig veld. Beschouw de matrix $B = (1\;1\;1\;1\;1)^T$ en $L_B : \mathbb{F} \to \mathbb{F}^5 : x \mapsto B \cdot x = (x,x,x,x,x)^T$. Een dergelijke code $\mathrm{Im}(L_B)$ wordt een herhalingscode of repetitiecode genoemd. Het is een $(5,1)$-lineaire code over $\mathbb{F}$. De codewoorden worden gevormd door het eerste element nog vier keer te herhalen. De informatieverhouding is $\frac{1}{5}$.

4. Geef een voortbrengende matrix en een controlematrix voor de codes hierboven en voor het eenvoudig codeersysteem uit het inspirerend voorbeeld in paragraaf 9.1.2.

Lemma 9.7

Veronderstel dat C een systematische (n,k)-lineaire code is over het eindig veld $\mathbb{F}$ met controlematrix H van de vorm $\begin{pmatrix} A & \mathbb{I}_{n-k} \end{pmatrix}$. Dan is de $(k \times n)$-matrix

$$G = \begin{pmatrix} 1 & 0 & \cdots & 0 & * & * & \cdots & * \\ 0 & 1 & \cdots & 0 & * & * & \cdots & * \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & * & * & \cdots & * \end{pmatrix} = \begin{pmatrix} \mathbb{I}_k & -A^T \end{pmatrix}$$

een voortbrengende matrix van C.

BEWIJS. We moeten aantonen dat $v \in \mathbb{F}^n$ een codewoord is als en slechts als v tot de rijruimte van G behoort.

Neem een vector v in de rijruimte van G. Er bestaat dus een vector $b \in \mathbb{F}^k$, zó dat $b^T \cdot G = v^T \in \mathbb{F}^n$. Dus geldt dat $v^T = (b^T, -b^T \cdot A^T)$ en dus $v = (b, -A \cdot b)$. Bijgevolg is

$$H \cdot v = A \cdot b + \mathbb{I}_{n-k} \cdot (-A \cdot b) = A \cdot b - A \cdot b = 0 \in \mathbb{F}^{n-k}$$

en behoort v tot C.

Omgekeerd observeren we gewoon dat de rang van G, dus de dimensie van de rijruimte van G, duidelijk k is. We toonden zopas aan dat die rijruimte een deelruimte is van C. Omdat C ook dimensie k heeft, moet dus gelden dat die rijruimte gelijk is aan C (wegens Stelling 3.47(2).

∎

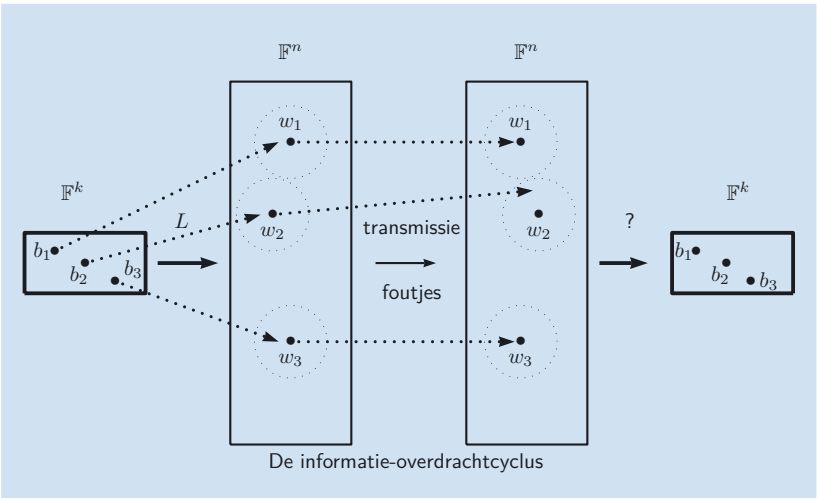
Merk op dat C hier ontstaat als beeld van een lineaire afbeelding $\mathbb{F}^k \to \mathbb{F}^n$ die een k-tal afstuurt op een n-tal *met dezelfde eerste k coördinaten*. De matrix van deze lineaire afbeelding ten opzichte van de twee standaardbasissen is G^T. Omwille van de vorm van G spreken we hier van een standaard generatormatrix.

Over lineaire codes, de eigenlijke probleemstelling en het gebruik voor foutendetectie en foutencorrectie vind je een extra downloadbare module op de webpagina van het boek: `http://www.lineairealgebra.be`.

Scan deze code om die pagina te openen:

9.1.4 De eigenlijke probleemstelling

We grijpen terug naar het schema met de informatie-overdrachtcyclus en passen het aan als volgt: boodschappen b_1, b_2, b_3 worden gecodeerd tot codewoorden w_1, w_2 en w_3. Tijdens de transmissie kunnen foutjes optreden. Het is derhalve niet zeker dat, aan de kant van de ontvanger, ook de codewoorden w_1, w_2 en w_3 ontvangen worden. Het zullen wel woorden van dezelfde lengte zijn, maar mogelijks verschillend van de verzonden codewoorden.



De informatie-overdrachtcyclus

De eigenlijke opdracht waarvoor we komen te staan, formuleren we als volgt: *Kunnen we een L vinden,* waarvoor de codewoorden w_1, w_2, ...(in $\mathrm{Im}(L)$) redelijk 'ver' uit elkaar liggen, zodanig dat als er tijdens de transmissie een kleine fout optreedt, we in staat zijn die fout te ontdekken en eventueel zelfs ze te herstellen? We willen dus fouten detecteren, en zo mogelijk ook corrigeren. En, meer zelfs, kunnen we ook een L vinden zodat we op de koop toe redelijk snel kunnen decoderen?

Alhoewel reeds nauwkeuriger, noopt deze vraagstelling tot een wiskundige precisering van enkele begrippen. Immers, wat betekent het dat codewoorden 'dicht bij' of 'ver van' elkaar liggen? En op basis van welk principe zullen we een eventuele fout corrigeren? Deze concepten komen nu aan bod.

9.1.5 Hammingafstand en Hamminggewicht

Bij de figuur hiervoor hadden we het reeds over codewoorden die al of niet 'ver' uit elkaar liggen. Dit intuïtief idee kunnen we als volgt nauwkeurig formuleren.

Definitie 9.8

Zij $\mathbb{F}$ een eindig veld. De **Hammingafstand** tussen twee vectoren van $\mathbb{F}^n$ is het aantal componenten waar die vectoren van elkaar verschillen. Het **Hamminggewicht** van een vector in $\mathbb{F}^n$ is het aantal niet-nul componenten van die vector. Het **Hamminggewicht** van een lineaire code is het minimum van de Hamminggewichten van alle niet-nul vectoren uit die code.

Voor $u, v \in \mathbb{F}^n$ noteren we $d_H(u, v)$ voor hun Hammingafstand. We schrijven $w_H(u)$ voor het Hamminggewicht van een vector u en analoog $w_H(C)$ voor het Hamminggewicht van een code C. Voor $u, v \in \mathbb{F}^n$ is duidelijk

$$d_H(u, v) = w_H(u - v) = w_H(v - u).$$

Bijgevolg geldt, als C een lineaire code is, dat

$$\begin{aligned} w_H(C) &= \min\{w_H(u) \mid 0 \neq u \in C\} \\ &= \min\{d_H(u, v) \mid u, v \in C \text{ en } u \neq v\}. \end{aligned}$$

Voorbeeld 9.9

1. Het idee van Hammingafstand moeten we, strikt gezien, niet voorbehouden voor vectoren uit $\mathbb{F}^n$. Immers, als we bijvoorbeeld denken aan de woorden van lengte vijf uit het woordenboek Nederlands, dan kunnen we evengoed stellen dat

   $$d_H(\text{tafel}, \text{stoel}) = 3$$

   omdat in deze beide woorden op juist drie plaatsen de letters verschillen.

2. Beschouw in $\mathbb{Z}_{31}^{10}$ de vectoren

   $$u = (7, 3, 0, 22, 8, 0, 30, 0, 9, 5) \quad \text{en} \quad v = (11, 3, 0, 1, 8, 0, 30, 0, 0, 5).$$

   Duidelijk zijn $w_H(u) = 7$, $w_H(v) = 6$ en $d_H(u, v) = 3$.

3. Als tijdens een informatie-overdrachtcyclus een storing optreedt, zó dat er in een codewoord één component wijzigt, dan ontvangen we een woord dat op Hammingafstand 1 van het verstuurde codewoord ligt. Zoiets noemen we een enkelvoudige fout.

Opdracht 9.10 Verifieer dat het Hamminggewicht van de code in Voorbeeld 9.1 gelijk is aan 3.

Lemma 9.11

Zij C een (n, k)-lineaire code over een eindig veld $\mathbb{F}$. De Hammingafstand bepaalt een metriek op C; dit wil zeggen dat $d_H : C \times C \to \mathbb{R} : (u, v) \mapsto d_H(u, v)$ een metriek is. In het bijzonder voldoet de Hammingafstand dus aan de driehoeksongelijkheid, met andere woorden geldt voor alle u, v, w in C dat $d_H(u, w) \leq d_H(u, v) + d_H(v, w)$.

BEWIJS. Het bewijs van deze eigenschappen is eenvoudig. Het is hierbij belangrijk te realiseren dat $d_H(u, v)$ in feite het aantal posities telt dat je moet wijzigen om de vector u te veranderen tot de vector v. Met deze omschrijving volgt de driehoeksongelijkheid vrij direct.

∎

Op basis van welk principe zullen we ontvangen woorden decoderen? Een frequent gehanteerd concept hiervoor is het dichtste-buur-principe.

Definitie 9.12

Zij C een (n,k)-lineaire code over het eindig veld $\mathbb{F}$. Als in een informatie-overdrachtcyclus een vector $v \in \mathbb{F}^n$ ontvangen wordt, gaan we ervan uit dat het codewoord dat oorspronkelijk verstuurd werd, het meest dichtbij gelegen codewoord w (in C) was, indien deze **dichtste buur** uniek is. We decoderen v dan als w. Indien er geen unieke dichtste buur onder de codewoorden is, wordt v niet gedecodeerd. Dit principe noemen we het **dichtste-buur-decoderingsprincipe** (Engels: nearest neighbour of maximum likelihood) .

Voorbeeld 9.13 Veronderstel dat $C = \{(0,0,0),(1,1,1)\}$. Dit is de $(3,1)$-binaire repetitiecode. Indien ten gevolge van een transmissie de ontvanger het woord $(1,0,1)$ ontvangt, dan weet hij dat dit geen geldig codewoord is. Het codewoord dat het dichtst bij dit woord ligt is $(1,1,1)$ en bijgevolg beslist de ontvanger dat dit het effectieve verstuurde woord was.

Er is een interessant verband tussen het Hamminggewicht van een lineaire code en de capaciteit ervan om fouten te corrigeren, gebruik makend van het dichtste-buur-decoderingsprincipe. We formuleren dit hier.

Lemma 9.14

Zij C een lineaire code over het eindig veld $\mathbb{F}$. Als het Hamminggewicht $w_H(C)$ gelijk is aan $2t+1$, dan kunnen op basis van het dichtste-buur-decoderingsprincipe tot maximaal t fouten gecorrigeerd worden.

Bewijs. Als het Hamminggewicht van de code C gelijk is aan $2t+1$, wil dat zeggen dat alle codewoorden onderling minimaal op afstand $2t+1$ van elkaar liggen. Veronderstel nu dat we een codewoord w versturen en dat er tijdens de transmissie maximaal t fouten optreden. Dan zal het ontvangen woord, zeg v, maximaal op afstand t van w liggen, en bijgevolg minstens op afstand $t+1$ van elk ander codewoord. De dichtste buur van v is dus duidelijk w. We zullen v decoderen als w, en zo worden de opgetreden fouten allemaal gecorrigeerd.

∎

Opmerking 9.15 Het Hamminggewicht van een lineaire code kan natuurlijk ook even zijn, zeg gelijk aan $2t$. Een analoge redenering leert ons dat er dan maximaal $t-1$ fouten kunnen gecorrigeerd worden.

9.1.6 Foutendetectie, foutencorrectie en Hammingcodes

De controlematrix is een belangrijk instrument om fouten te detecteren bij de informatie-overdracht. Immers, na het verzenden van een codewoord kan de ontvanger, aan de hand van de controlematrix, snel en eenvoudig controleren of het woord dat hij ontvangt, zeg v, effectief ook een codewoord is. Het volstaat $H \cdot v$ uit te rekenen; als dat niet de nulvector oplevert, dan is er tijdens de transmissie een fout opgetreden.

Laat ons nu even veronderstellen dat er tijdens het verzenden van een codewoord juist één fout is opgetreden. Dit betekent dat op precies één plaats in het codewoord een coördinaat wijzigde. In het geval van een binaire code betekent dit dat een 0 is veranderd in een 1, of vice versa. Veronderstel dat de fout op positie i in het codewoord gebeurde.

We noteren, zoals voorheen, de standaardbasis van $\mathbb{F}^n$ met $e_1 = (1, 0, \ldots, 0)^T, e_2 = (0, 1, 0, \ldots, 0)^T, \ldots, e_n = (0, 0, \ldots, 0, 1)^T$. Dan is een codewoord $w \in C$, waarin op positie i juist één fout optreedt, veranderd in een woord $v = w + \lambda e_i$ (met $\lambda \in \mathbb{F} \setminus \{0\}$).

Als we dit woord v ontvangen en we berekenen $H \cdot v$, dan vinden we

$$H \cdot v = H \cdot (w + \lambda e_i) = H \cdot w + H \cdot \lambda e_i = \lambda\,(H \cdot e_i).$$

Nu is $H \cdot e_i$ precies de i-de kolom van de controlematrix H. Als er slechts één fout is opgetreden, en we vermenigvuldigen het ontvangen woord met de controlematrix, dan vinden we met andere woorden een veelvoud van een welbepaalde kolom in de controlematrix als resultaat. Bovendien vertelt het nummer van die kolom, waarvan we het veelvoud als resultaat vinden, op welke plaats in v de fout is opgetreden. En de coëfficiënt λ leert ons hoe de correcte waarde op die positie in v moet zijn.

Toch kunnen er complicaties optreden. Inderdaad, als er in de controlematrix H gelijke kolommen staan of, meer algemeen, kolommen die gelijk zijn op een veelvoud met een coëfficiënt na, dan kunnen we niet met zekerheid beslissen op welke positie de fout is opgetreden. (Een speciaal geval hiervan is als er een nulkolom in H staat; dan kunnen we eventueel $H \cdot v = 0$ vinden, ook al is v geen codewoord.) Maar dit is de enige moeilijkheid. Indien in H geen enkele kolom een veelvoud is van een andere kolom, dan kunnen we, zoals hierboven uitgelegd, met behulp van H enkelvoudige fouten corrigeren.

Samengevat hebben we de volgende interessante stelling aangetoond.

Stelling 9.16

Zij C een (n, k)-lineaire code over $\mathbb{F}$ met controlematrix H. Veronderstel dat we met een informatieoverdrachtcyclus werken waarin slechts enkelvoudige fouten optreden. Dan geldt dat we alle fouten die optreden tijdens de overdrachtcyclus kunnen corrigeren met behulp van de informatie uit $H \cdot v$ als en slechts als in de controlematrix H geen enkele kolom een scalair veelvoud is van een andere kolom.

De familie lineaire codes die bekend staan als de Hammingcodes zijn van een type waarop deze stelling van toepassing is. In het binaire geval, dit wil zeggen over het veld $\mathbb{F} = \mathbb{Z}_2$, worden ze als volgt gedefinieerd.

Definitie 9.17

Een binaire Hammingcode is een lineaire code waarvan de kolommen in de pariteitscontrolematrix H precies alle niet-nul vectoren zijn in $\mathbb{Z}_2^m$, voor een zekere $m \geqslant 2$. Zo'n code heeft dus lengte $n = 2^m - 1$ en dimensie $k = 2^m - m - 1$.

In zo'n matrix H zijn elke twee kolommen lineair onafhankelijk, precies omdat ze allemaal verschillend zijn en we over $\mathbb{Z}_2$ werken.

Voorbeeld 9.18

1. Neem $m = 2$. Dan vinden we bijvoorbeeld

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{en} \quad G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}.$$

   In dit geval heb je een $(3, 1)$-lineaire code, die in feite gewoon een repetitiecode is.

2. Neem $m = 3$. Er zijn 2^3 binaire woorden van lengte 3 (equivalent: er zijn 2^3 elementen in $\mathbb{Z}_2^3$). Er zijn dus zeven niet-nul vectoren; we moeten een matrix vormen met deze zeven vectoren in de kolommen, in een bepaalde volgorde. De matrix H in Voorbeeld 9.1 is precies zo'n matrix, die ook nog in systematische vorm is:

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

Binaire Hammingcodes hebben de volgende belangrijke eigenschap.

Stelling 9.19

Zij C een binaire Hammingcode. Dan is het Hamminggewicht van C gelijk aan 3.

BEWIJS. Veronderstel dat C bepaald wordt door de controlematrix H. Dus geldt dat $C = N(H)$, de nulruimte van H. Voor een vector v die op afstand 1 van een codewoord ligt, geldt dat $H \cdot v$ gelijk is aan een kolom uit de matrix H. Voor een vector v die op afstand 2 van een codewoord ligt, geldt dat $H \cdot v$ gelijk is aan een lineaire combinatie van twee kolommen uit H. Omdat alle kolommen in H onderling verschillend zijn (per definitie), en omdat we werken over $\mathbb{Z}_2$, betekent dit dat een lineaire combinatie van twee

kolommen uit H nooit de nulrij oplevert. Bijgevolg is de kleinste afstand tussen twee codewoorden in een Hammingcode minstens 3.

∎

Binaire Hammingcodes zijn bijgevolg in staat om enkelvoudige fouten te corrigeren. Bovendien doen ze dat op een zeer eenvoudige en efficiënte manier.

We illustreren dit even met de $(7,4)$-lineaire Hammingcode uit Voorbeeld 9.1. Stel dat een boodschap $b \in \mathbb{Z}_2^4$ gecodeerd wordt als het codewoord $w \in C \subset \mathbb{Z}_2^7$, en dat na transmissie $v = 0111010$ ontvangen wordt. We veronderstellen hierbij dat er hoogstens één fout gebeurd is bij de transmissie. Wat is de boodschap b?

Hiervoor berekenen we

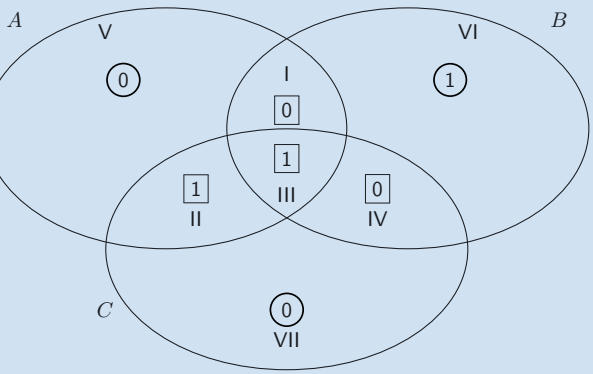
$$H \cdot v = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Dit is de vierde kolom van H; er is dus een fout gebeurd, en wel op de vierde plaats. Het codewoord was dus $w = 0110010$ en de boodschap is $b = 0110$.

In dit voorbeeld kan je het vormen van een codewoord ook heel mooi visueel weergeven met een Venn-diagramschema, en wel als volgt. Het schema toont drie verzamelingen A, B en C in algemene positie. Zo ontstaan zeven gebieden, genummerd met Romeinse cijfers I, …, VII.

Als we een binaire boodschap van lengte 4 willen coderen, bijvoorbeeld opnieuw 0110, dan plaatsen we de cijfers hiervan in de gebieden I, II, III en IV (zie $\boxed{0}$, …). Vervolgens berekenen we in de verzamelingen A, B en C het resterende cijfer, dit wil zeggen voor de gebieden V, VI en VII. We doen dat op zo een wijze dat in elk van de drie verzamelingen het aantal 1-tjes een even getal is. Voor het woord 0110 resulteert dit in een extra drietal 010 (zie $\textcircled{0}$, …), zodat het codewoord dat we verkrijgen gelijk is aan 0110010.

Veronderstel nu dat dit codewoord verstuurd wordt, en dat er hierbij juist één fout optreedt. Hanteer nu dezelfde grafische werkwijze, en plaats het ontvangen woord in dit verzamelingenschema. Ontdek de verzamelingen waar de pariteit niet meer in orde is. Dit laat zien welk cijfer in het ontvangen woord fout is.



De binaire Hamming-(7,4)-code: grafisch

Opdracht 9.20 Beschouw de $(15, 11)$-lineaire (binaire) Hammingcode C met controlematrix

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} A & \mathbb{I}_4 \end{pmatrix} \in \mathbb{Z}_2^{4\times 15}$$

en standaard generatormatrix

$$G = \begin{pmatrix} \mathbb{I}_{11} & -A^T \end{pmatrix} \in \mathbb{Z}_2^{11\times 15}.$$

Er is een boodschap $b \in \mathbb{Z}_2^{11}$ gecodeerd als $w \in C \subset \mathbb{Z}_2^{11}$, en dit wordt na transmissie $v = 100111010010010$. Bereken, in de veronderstelling dat er hoogstens één fout gebeurd is bij de transmissie, het verzonden codewoord w, en lees af wat de boodschap b is.

Opmerking 9.21 Met de vorige Hammingcode moeten er, om één fout te kunnen verbeteren na het doorzenden van een boodschap van elf bits, slechts vier 'overtollige' bits toegevoegd worden. De informatieverhouding is hier $\frac{11}{15}$, wat vrij hoog is. Met de naïeve repetitiecode zijn er voor een boodschap van elf bits *tweeëntwintig* 'overtollige' bits nodig!

9.2 Op naar de stelling van Jordan

Zoals we reeds zagen, blijkt helaas niet elke lineaire transformatie L diagonaliseerbaar te zijn. Soms situeert het probleem zich ter hoogte van de karakteristieke veelterm φ_L, die bijvoorbeeld geen of onvoldoende nulpunten heeft. Een andere keer is de karakteristieke veelterm volledig ontbindbaar, maar is er een verschil tussen algebraïsche en meetkundige multipliciteit voor minstens één eigenwaarde.

Men kan zich daarom terecht afvragen wat, meer algemeen, de best mogelijke eenvoudige matrixvorm voor een lineaire transformatie is. Voor de volledigheid brengen we hier nog enkele belangrijke resultaten.

Het eerste resultaat dat we hier wensen onder de aandacht te brengen, stelt dat op een eindigdimensionale vectorruimte, en onder redelijke voorwaarden, elke lineaire transformatie kan voorgesteld worden met behulp van een bovendriehoeksmatrix. We noemen dit het triangulariseren van een lineaire transformatie.

Stelling 9.22 (Triangularisatiestelling voor lineaire transformaties)

Veronderstel dat $L : V \to V$ een lineaire transformatie is van de n-dimensionale vectorruimte V, zó dat de karakteristieke veelterm φ_L volledig ontbindt als product van eerstegraadsfactoren. Dan bestaat er een basis voor de vectorruimte V, ten opzichte van dewelke de matrixvoorstelling van L bovendriehoeks is.

Bewijs. We geven een bewijs, voornamelijk in matrixtaal, en per inductie op de dimensie van V. Zij eerst $\dim V = 1$. Dan is de matrix van L een (1×1)-matrix, en uiteraard is die bovendriehoeks. Er is met andere woorden niets aan te tonen in dit bijzondere geval.

Laat ons nu veronderstellen dat de stelling correct is voor vectorruimten van dimensie kleiner dan of gelijk aan $n - 1$. In matrixtaal wil dit zeggen: we veronderstellen dat elke vierkante matrix, van afmeting kleiner dan of gelijk aan $n - 1$ en met een karakteristieke veelterm die volledig ontbindt als product van eerstegraadsfactoren, gelijkvormig is met een bovendriehoeksmatrix. Deze veronderstelling is onze inductiehypothese.

Zij nu V een vectorruimte van dimensie n en $L : V \to V$ een lineaire transformatie met volledig ontbindbare karakteristieke veelterm φ_L. Omdat deze veelterm volledig ontbindt, heeft hij dus zeker minstens één nulpunt (dit wil zeggen een eigenwaarde van L), en evenzo een bijhorende eigenvector (verschillend van de nulvector!). Laat ons λ noteren voor deze eigenwaarde en v voor een bijhorende eigenvector. De verzameling $\{v\}$ kan uitgebreid worden tot een basis $\beta = \{v, \ldots\}$ van V. Ten opzichte van deze basis verkrijgen we een

matrixvoorstelling A van L van de vorm

$$A = \begin{pmatrix} \lambda & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{pmatrix} \overset{\text{noteer}}{=} \begin{pmatrix} \lambda & * \\ 0 & B \end{pmatrix},$$

waarin B de $((n-1)\times(n-1))$-matrix

$$\begin{pmatrix} a_{22} & \dots & a_{2n} \\ \vdots & & \vdots \\ a_{n2} & \dots & a_{nn} \end{pmatrix} \text{ is.}$$

Hieruit volgt dat

$$\varphi_L(X) = \det(X\mathbb{I}_n - A) = \det \begin{pmatrix} X-\lambda & -* \\ 0 & X\mathbb{I}_{n-1} - B \end{pmatrix} = (X-\lambda)\varphi_B(X).$$

Omdat φ_L volledig ontbindt in eerstegraadsfactoren, is dat dus ook het geval voor φ_B. We kunnen bijgevolg de inductiehypothese inzetten voor de matrix B, en concluderen dat B gelijkvormig is met een bovendriehoeksmatrix. Dit wil zeggen dat er een inverteerbare $((n-1)\times(n-1))$-matrix Q bestaat, zó dat $Q^{-1}\cdot B\cdot Q$ bovendriehoeks is.

Vorm nu de $(n\times n)$-matrix

$$P = \begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix},$$

waarbij 0 staat voor een rij of kolom met $n-1$ keer het getal 0. Ook dit is een inverteerbare matrix, en bovendien geldt dat

$$P^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & Q^{-1} \end{pmatrix}.$$

We berekenen

$$\begin{aligned} P^{-1}\cdot A\cdot P &= \begin{pmatrix} 1 & 0 \\ 0 & Q^{-1} \end{pmatrix}\cdot\begin{pmatrix} \lambda & * \\ 0 & B \end{pmatrix}\cdot\begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix} \\ &= \begin{pmatrix} \lambda & * \\ 0 & Q^{-1}\cdot B \end{pmatrix}\cdot\begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix} = \begin{pmatrix} \lambda & *' \\ 0 & Q^{-1}\cdot B\cdot Q \end{pmatrix} \end{aligned}$$

en zien in dat deze laatste matrix bovendriehoeks is. Bijgevolg is A gelijkvormig met een bovendriehoeksmatrix; dat is precies wat we moesten aantonen.

∎

Over de resultaten die leiden tot de algemeen geldende stelling van Jordan die de eenvoudigste matrixvorm van lineaire transformaties beschrijft, brengen we aanvullend materiaal op de webpagina van het boek: `http://www.lineairealgebra.be`.

Scan deze code om die pagina te openen:

Verwijzend naar de Hoofdstelling van de Algebra die waarborgt dat elke veelterm met complexe coëfficiënten volledig ontbindbaar is (over $\mathbb{C}$), verkrijgen we nu deze conclusie.

Gevolg 9.23

Op een eindigdimensionale complexe vectorruimte $(\mathbb{C}, V, +)$ is elke lineaire transformatie triangulariseerbaar.

In de verdere studie van de lineaire algebra kan men nog een hele stap verder gaan in het vereenvoudigen van de matrixvorm van een lineaire transformatie. Dit leidt ons tot een tweede resultaat, bijzonder sterk, mooi en zeer algemeen, en bekend als de stelling van Jordan. Het bewijs ervan maakt deel uit van een meer gevorderde theorie en zou ons binnen het kader van dit boek te ver leiden. Een geïnteresseerde lezer vindt ongetwijfeld goede aanvullende informatie in bijvoorbeeld hoofdstuk 8 van Kwak en Hong [3] of hoofdstuk 7 van Friedberg, Insel en Spence [1].

Camille Jordan (Lyon, 1838 - Parijs, 1922)
Jordan studeert aan de École Polytechnique en werkt een tijd lang als ingenieur. Later wordt hij professor aan diezelfde École Polytechnique en ook aan het zeer befaamde Collège de France. In zijn tijd was hij de belangrijkste onderzoeker in de groepentheorie. Zijn Traité des substitutions *(1870) over permutatiegroepen was lange tijd een standaardwerk. Zijn naam is ook verbonden aan de stelling van Jordan-Hölder over de compositiefactoren van een groep.*

Naast de Jordanmatrix is er ook een Jordankromme. *Dit is een vlakke kromme die topologisch equivalent is aan de cirkel. Jordan bewees de intuïtief duidelijke maar moeilijk aan te tonen stelling dat elke dergelijke kromme het vlak in twee samenhangende delen verdeelt.*

We formuleren hier bondig de meest essentiële begrippen en de prachtige stelling van Jordan. Tegelijk introduceren we ook de gangbare terminologie.

Definitie 9.24

Een $(k \times k)$-matrix met elementen in $\mathbb{R}$ of $\mathbb{C}$ noemen we een **Jordanblok** (soms ook een Jordan-k-blok) als die van de vorm

$$\begin{pmatrix} \lambda & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda & 1 & & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & 0 & & \lambda & 1 & 0 \\ 0 & 0 & & 0 & \lambda & 1 \\ 0 & 0 & \dots & 0 & 0 & \lambda \end{pmatrix}$$

is, waarbij λ willekeurig is. We noteren $J_k(\lambda)$ voor een dergelijke matrix.

Met het begrip Jordanblok kunnen we nu komen tot een nog meer algemene definitie.

Definitie 9.25

Een $(n \times n)$-matrix A noemen we een **Jordanmatrix** als A van de volgende vorm is:

$$A = \begin{pmatrix} J_{k_1}(\lambda_1) & 0 & \dots & 0 \\ 0 & J_{k_2}(\lambda_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & J_{k_m}(\lambda_m) \end{pmatrix}.$$

Merk op dat A als het ware bloksgewijs wordt voorgesteld en, uiteraard, dat $k_1 + k_2 + \cdots + k_m = n$.

Voorbeeld 9.26 Enkele voorbeelden helpen om deze begrippen te verduidelijken.

- De matrices $J_2(5) = \begin{pmatrix} 5 & 1 \\ 0 & 5 \end{pmatrix}$ en $J_3(-1) = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix}$ zijn respectievelijk voorbeelden van een Jordan-2-blok en een Jordan-3-blok.
- Elke (1×1)-matrix, zoals bijvoorbeeld $\begin{pmatrix} 3 \end{pmatrix}$, is een Jordanblok, in dit geval genoteerd als $J_1(3)$.

- De matrix $J_2(0) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ is een Jordan-2-blok.
- De matrix

$$A = \left(\begin{array}{ccc|ccc} 2 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{array}\right),$$

  met extra horizontale en verticale lijnen ter visuele ondersteuning, is een Jordanmatrix waarin we een 1-blok $J_1(2)$, een 2-blok $J_2(2)$ en een 3-blok $J_3(5)$ aantreffen.
- Elke diagonaalmatrix is een Jordanmatrix, met uitsluitend 1-blokken. Anders gezegd, matrices die wij Jordanmatrices noemen zijn een veralgemening van (de meer bijzondere) diagonaalmatrices.
- Zoals reeds blijkt uit de matrix A hierboven, hoeven de λ_i in de definitie van een Jordanmatrix niet verschillend te zijn. Voor eenzelfde eigenwaarde λ kunnen in de Jordanmatrix verschillende Jordanblokken optreden.

Stelling 9.27 (Stelling van Jordan)

Zij $L : V \to V$ een lineaire transformatie van de eindigdimensionale complexe vectorruimte $(\mathbb{C}, V, +)$. Dan bestaat er een basis β voor V, zó dat de matrixvoorstelling van L ten opzichte van die basis een Jordanmatrix is. De getallen op de diagonaal in die Jordanmatrix zijn de eigenwaarden van L, elk met hun algebraïsche multipliciteit optredend. Op de volgorde van de Jordanblokken na is deze Jordanvorm uniek bepaald door L. Een dergelijke basis voor V noemen we een **Jordanbasis**.

Extra online module bij Lineaire algebra, derde editie, Paul Igodt, Wim Veys,
ISBN 9789462703148, © Universitaire Pers Leuven 2022

Bibliografie

[1] S.H. Friedberg, A.J. Insel and L.E. Spence, *Linear Algebra*, Pearson Education Inc., New Jersey, 2003.

[2] R. Penrose, *A generalized inverse for matrices*, Math. Proc. Cambridge Phil. Soc., vol. 51, nr. 3, pp. 406-413, 1955.

[3] J.H. Kwak and S. Hong, *Linear Algebra*, Birkhäuser, Boston, 2004.

[4] L. Page, S. Brin, R. Motwani and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report, Stanford InfoLab, 1998.

[5] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

[6] K. van Harn en P.J. Holewijn, *Markov-ketens in discrete tijd*, Epsilon Uitgaven, Utrecht, 2003.

[7] J.H. van Lint, *Mathematics and the Compact Disc. John Bernoulli Lecture 1998.* Nieuw Archief voor Wiskunde, nr. 3, pp. 183-190, 1998.

Wat hebben wiskundigen te maken met de Eiffeltoren?

Zoek bij je volgende bezoek aan Parijs een manier om de buitenkant van de eerste verdieping van de Eiffeltoren te bestuderen.

Daarop staan in gouden letters de namen van tweeënzeventig beroemde Fransen ('Les 72 savants'), vooral wetenschappers en ingenieurs.

Achttien daarvan zijn wiskundigen, waaronder Cauchy, Fourier, Lagrange en Legendre. Ze zijn gekozen wegens hun verdiensten, en ook omdat hun naam niet meer dan twaalf letters bevat.

Index