Kapitel 12

EN KORT INLEDNING TILL GRUPPKODER

I många kommunikationssystem översätter man information till följder av nollor och ettor. Antag att man vill sända två meddelanden A och B. Det enklaste sättet är att översätta:

$$\begin{aligned} A &\longmapsto 0, \\ B &\longmapsto 1. \end{aligned}$$

Överföringen sker med hjälp av t ex ledningar eller radiovågor eller på något annat sätt. Resultatet kan bli att beroende på störningar i kommunikationskanalen nollan förvandlas till en etta eller tvärtom. Finns det någon möjlighet att skydda sig mot en sådan störning? En möjlig lösning är att upprepa A och B till exempel två gånger dvs

$$\begin{aligned} A &\longmapsto 00, \\ B &\longmapsto 11. \end{aligned}$$

Om den mottagna sekvensen är nu 01 eller 10 så kan man konstatera att det har inträffat ett fel. Med andra ord kan man upptäcka ett fel. Låt oss gå vidare och upprepa A och B tre gånger dvs

$$\begin{aligned} A &\longmapsto 000, \\ B &\longmapsto 111. \end{aligned} \tag{12.1}$$

Situationen har förbättrats avsevärt. Om det inträffar högst ett fel i A eller B så får man följande sekvenser av signaler:

$$\begin{aligned} A &\longmapsto 000, 100, 010, 001, \\ B &\longmapsto 111, 011, 101, 110. \end{aligned}$$

Nu kan man inte bara upptäcka högst ett fel utan också korrigera det. Om man nämligen har högst ett fel i A så får man en sekvens ur övre raden, däremot ger högst ett fel i B alltid en sekvens ur nedre raden. Detta betyder att högst ett fel i A kan aldrig leda till en sekvens som är ett resultat av högst ett fel i B. Om man får en sekvens ur övre raden och man antar att det har inträffat högst ett fel så kan man korrekt avläsa meddelandet som A. På samma sätt kan man sluta sig till B om man får en sekvens ur nedre raden.

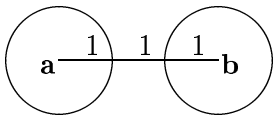
Detta är det enklaste exemplet på en felkorrigerande kod. Rent allmänt kan man beskriva situationen på följande sätt: Man har en mängd av meddelanden X och en metod att översätta dessa meddelanden till sekvenser av 0 och 1. Låt C vara mängden av alla kodord. C innehåller sekvenser av 0 och 1. Man brukar skriva $\mathbb{Z}_2$ för att beteckna mängden bestående av 0 och 1. Då skriver man $\mathbb{Z}_2^n$ för att beteckna mängden av alla sekvenser $(a_1, a_2, \ldots, a_n)$ av 0 och 1 av längden n. Man säger också att $\mathbb{Z}_2^n$ är **mängden av binära vektorer av längden** n. T ex består $\mathbb{Z}_2^3$ av följderna $000, 001, 010, 011, 100, 101, 110, 111$ (för enkelhets skull skriver vi här och i fortsättningen *abc*.. i stället för $(a, b, c, ...)$ om detta inte leder till missförstånd). Mera formellt kan man också säga att en kod är en funktion

$$X \longrightarrow C \subseteq \mathbb{Z}_2^n.$$

som mot olika element (=meddelanden) i X ordnar olika vektorer. Men enklast är att betrakta en kod som en delmängd till $\mathbb{Z}_2^n$. Därför antar vi följande definition:

(12.2) Definition. Med en **kod** menar man en godtycklig delmängd C till $\mathbb{Z}_2^n$. □

Vad är det som gör att koden C i vårt första exempel (12.1) kan korrigera 1 fel? Svaret är att högst ett fel i ett av kodorden inte kan sammanblanda den resulterande vektorn med de vektorer som man får då högst ett fel inträffar i ett annat kodord. Hur kan man uttrycka den egenskapen i matematiska termer ? Man kan säga att två olika kodord måste skilja sig på minst tre olika ställen. Detta är just den förutsättning som garanterar att ett fel i det ena kodordet inte kan ge upphov till en vektor som är ett resultat av ett fel i ett annat kodord. Man kan försöka föreställa sig situationen geometriskt så att kodorden är punkter och alla vektorer som man kan få ur ett kodord då högst ett fel inträffar bildar en cirkel med centrum i kodordet och med radien 1:



Olika cirklar kan inte överlappa för att garantera att varje vektor som skiljer sig från ett kodord på högst ett ställe skall kunna återföras på just detta kodord dvs på cirkelns centrum. Lite mera formellt kan man definiera avståndet mellan två vektorer i $\mathbb{Z}_2^n$:

(12.3) Definition. Låt $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ och $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ vara vektorer i $\mathbb{Z}_2^n$. Talet

$$d(\boldsymbol{a},\boldsymbol{b}) = \text{antalet } i \text{ sådana att } a_i \neq b_i$$

kallas **avståndet** mellan $\boldsymbol{a}$ och $\boldsymbol{b}$. Vi skall beteckna med $d(C)$ det minsta avståndet mellan två olika kodord i C, dvs $d(C) = \min d(\boldsymbol{a},\boldsymbol{b})$ då $\boldsymbol{a},\boldsymbol{b} \in C$ och $\boldsymbol{a} \neq \boldsymbol{b}$. □

Man säger också att $d(\mathbf{a},\mathbf{b})$ är **Hammingsavståndet** mellan **a** och **b**. Det var R.W. Hamming som år 1950 publicerade den första intelligenta konstruktionen av felkorrigerande koder och på det sättet startade den algebraiska kodningsteorin. Vårt första exempel (12.1) är en så kallad **repetitionskod** dvs man upprepar varje meddelande ett antal gånger (här 3 gånger). Den metoden är väl-känd (och beprövad av varje lärare), men den är tidskrävande och dyrbar. Hamming konstruktion visar att felkorrigering kan förverkligas på ett mycket mera effektivt sätt. Hammingkoder är enkla och mycket vanliga i olika datorsystem där de används för felkorrigering.

Innan vi går vidare låt oss kort sammanfatta våra resultat. **En 1-felkorrigerande kod är en mängd av vektorer C i $\mathbb{Z}_2^n$ sådan att avståndet mellan olika kodord i C är minst lika med 3 dvs $d(C) \geq 3$.**

Hur kan man konstruera koder med den egenskapen dvs med $d(C) \geq 3$? Låt oss betrakta en matris bestående av nollor och ettor t ex

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

Betrakta också alla vektorer $x_1x_2x_3x_4x_5$ i $\mathbb{Z}_2^5$ som satisfierar det linjära ekvationssystem vars koefficienter bildar raderna i den matrisen. Vi söker alltså alla lösningar till ekvationssystemet:

$$\begin{array}{ccccccccccc} x_1 & + & x_2 & + & x_3 & & & & & = & 0 \\ x_1 & + & & & & & x_4 & & & = & 0 \\ & & x_2 & + & & & & & x_5 & = & 0. \end{array}$$

Vi vill hitta alla lösningar som är sekvenser av 0 och 1. Additionen och multiplikationen av 0 och 1 är inte de vanliga utan binära dvs våra räkneoperationer följer följande lagar:

$+$	0	1
0	0	1
1	1	0

$*$	0	1
0	0	0
1	0	1

På det sättet är t ex $1 + 1 = 0$ dvs $1 = -1$. Det intressanta är att även i detta fall gäller alla formella räknelagar kända för vanlig addition och multiplikation av vanliga tal dvs man har

associativiteten, kommutativiteten för både addition och multiplikation, distributiviteten för multiplikation med avseende för addition osv. Låt oss lösa ekvationssystemet! Vi får lätt att

$$\begin{aligned} x_3 &= x_1 + x_2 \\ x_4 &= x_1 \\ x_5 &= x_2. \end{aligned}$$

(Observera att minustecken kan ersättas med plustecken i den binära aritmetiken!) Nu kan vi välja för x_1 och x_2 helt godtyckliga värden 0 och 1. Då får vi motsvarande värden för x_3, x_4 och x_5. Resultatet är följande:

x_1	x_2	x_3	x_4	x_5
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

.

Nu har vi faktiskt konstruerat en 1-felkorrigerandekod. Vi kan använda den koden för att sända 4 meddelanden, säg A, B, C, D dvs

(12.4)

		x_1	x_2	$\longmapsto$	x_1	x_2	x_3	x_4	x_5
A	=	0	0	$\longmapsto$	0	0	0	0	0
B	=	0	1	$\longmapsto$	0	1	1	0	1
C	=	1	0	$\longmapsto$	1	0	1	1	0
D	=	1	1	$\longmapsto$	1	1	0	1	1

Det är lätt att kontrollera avstånden mellan olika kodord och konstatera att $d(C) = 3$. Den konstruktionen är redan en liten framgång. Om vi använder den naiva kodningsmetod som garanterar att ett fel kan korrigeras dvs om vi använder repetitionskoden så har vi följande översättning:

A	=	00	$\longmapsto$	00	00	00
B	=	01	$\longmapsto$	01	01	01
C	=	10	$\longmapsto$	10	10	10
D	=	11	$\longmapsto$	11	11	11

Kodorden har alltså längden 6. Kodorden i koden (12.4) har längden 5. Om antalet signaler är stort kan vinsten vara märkbar. Vi skall diskutera den aspekten närmare om en stund då vi konstruerar Hammingkoder.

Hur kan man rent allmänt konstruera liknande koder? Vad är det som gör att matrisen $\mathbf{H}$ ger upphov till en kod som är bättre än repetitionskoden ? En mycket viktig egenskap hos den sista koden har en stor betydelse i samband med kodkonstruktioner:

(12.5) Definition. Man säger att en kod är **linjär** eller en **gruppkod** om summan av två godtyckliga kodord också är ett kodord. Kodorden summeras som vektorer dvs om $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ och $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ så är

$$\mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n).$$

□

Den nyss konstruerade koden är linjär. Den egenskapen är enkel att kontrollera direkt, men man kan säga rent allmänt att summan av två lösningar till ett linjärt ekvationssystem med höger-leden lika med 0 också är en lösning till ett sådant system. Man kan lätt inse det direkt, men man kan också använda sig av matrisbeteckningar:

(12.6) Sats. *Om* $\mathbf{H}$ *är en binär matris med* n *kolonner så bildar alla lösningar* $\mathbf{x} \in \mathbb{Z}_2^n$ *till ekvationen* $\mathbf{H}\boldsymbol{x} = \mathbf{0}$ *en gruppkod**.

Bevis. Om $\mathbf{Ha} = \mathbf{0}$ och $\mathbf{Hb} = \mathbf{0}$ så är $\mathbf{H}(\mathbf{a} + \mathbf{b}) = \mathbf{Ha} + \mathbf{Hb} = \mathbf{0}$. □

Matrisen $\mathbf{H}$ brukar kallas **paritetsmatrisen** eller **kontrollmatrisen** av koden bestående av alla lösningar till $\mathbf{H}\boldsymbol{x} = \mathbf{0}$. Om matrisen $\mathbf{H}$ har k rader så väljer man ofta den så att de sista k kolonnerna bildar enhetsmatrisen (med k rader och k kolonner). Då säger man att matrisen $\mathbf{H}$ är **normaliserad**. Det är en fördel att ha en normaliserad matris därför att man då hittar lösningarna till ekvationen $\mathbf{H}\boldsymbol{x} = \mathbf{0}$ mycket enkelt (se t ex övning (12.2)).

Omvändningen av den sista satsen är också sann:

(12.7) Sats. *Varje gruppkod* $C \subseteq \mathbb{Z}_2^n$ *består av alla lösningar till en matrisekvation* $\mathbf{H}\boldsymbol{x} = \mathbf{0}$, *där* $\mathbf{H}$ *är en binär matris med* n*-kolonner.*

Detta påstående är inte svårt att bevisa, men det blir mycket enklare att göra det senare i kursen då vi återkommer till kodningsteorin.

För gruppkoder kan man relativt lätt undersöka avstånden mellan olika kodord dvs beräkna $d(C)$.

(12.8) Definition. Med **vikten** av $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ menas

$$w(\boldsymbol{a}) = \text{antalet } i \text{ sådana att } a_i \neq 0.$$

Med **vikten av en kod** C menar man den minsta vikten av nollskilda kodord dvs $w(C) = \min w(\mathbf{a})$ då $\boldsymbol{a} \neq \mathbf{0}$. □

***Observera att i den texten uppfattas vektorer alltid som kolonnvektorer då de multipliceras med matriser.**

Det visar sig att $d(C) = w(C)$ om koden C är linjär. För koden (1.4) konstaterar man med ett ögonkast att minimum av $w(\mathbf{a})$ är just 3 då $\mathbf{a} \neq \mathbf{0}$. Vi visar denna egenskap helt allmänt, men först antecknar vi några enkla samband mellan avståndet och vikten:

(12.9) Sats. *Låt $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{Z}_2^n$. Då gäller:*

(a) $d(\boldsymbol{a}, \boldsymbol{b}) = w(\boldsymbol{a} - \boldsymbol{b})$,

(b) $w(\boldsymbol{a} + \boldsymbol{b}) \leq w(\boldsymbol{a}) + w(\boldsymbol{b})$,

(c) $d(\boldsymbol{a}, \boldsymbol{b}) \leq d(\boldsymbol{a}, \mathbf{c}) + d(\mathbf{c}, \boldsymbol{b})$.

Bevis. Låt $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ och $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$. (a) följer omedelbart ur definitionerna av d och w: $a_i \neq b_i$ är ekvivalent med $a_i - b_i \neq 0$. Om $a_i + b_i \neq 0$ så är $a_i \neq 0$ eller $b_i \neq 0$. Detta bevisar (b). Nu är:

$$d(\boldsymbol{a}, \boldsymbol{b}) = w(\boldsymbol{a} - \boldsymbol{b}) = w(\boldsymbol{a} - \mathbf{c} + \mathbf{c} + \boldsymbol{b}) \leq w(\boldsymbol{a} - \mathbf{c}) + w(\mathbf{c} - \boldsymbol{b}) = d(\boldsymbol{a}, \mathbf{c}) + d(\mathbf{c}, \boldsymbol{b}),$$

vilket visar (c). □

Nu kan vi visa likheten mellan minimalavståndet och vikten i gruppkoder:

(12.10) Sats. *Låt C vara en linjär kod. Då är $w(C) = d(C)$.*

Bevis. Låt $w(C) = w(\boldsymbol{a})$. Då är

$$w(C) = w(\boldsymbol{a}) = d(\boldsymbol{a}, \mathbf{0}) \geq d(C).$$

Låt $d(C) = d(\boldsymbol{a}, \boldsymbol{b})$. Då är

$$d(C) = d(\boldsymbol{a}, \boldsymbol{b}) = w(\boldsymbol{a} - \boldsymbol{b}) \geq w(C).$$

ty $\boldsymbol{a} - \boldsymbol{b} \in C$. Alltså är $d(C) = w(C)$. □

När vi väljer en matris $\mathbf{H}$ som skall ge en kod som korrigerar 1 fel måste vi se till att $d(C) = w(C) \geq 3$. Detta betyder att det inte kan finnas lösningar till ekvationssystemet med vikten 1 eller 2. Vad betyder det att en lösning har vikten 1? Vi kan tänka oss en matris

$$\begin{matrix} 0 & 0 & \ldots & 1 & \ldots & 0 \end{matrix}$$
$$\begin{bmatrix} * & * & \ldots & * & \ldots & * \\ * & * & \ldots & * & \ldots & * \\ \vdots & \vdots & & \vdots & & \vdots \\ * & * & \ldots & * & \ldots & * \end{bmatrix}$$

där varje * betyder 0 eller 1. Om en vektor med exakt en etta satisfierar alla ekvationer som svarar mot raderna i den matrisen så måste kolonnen under ettan bestå av enbart nollor. Med andra ord är vikten av alla kodord $\neq 0$ minst 2 om det inte finns en nollkolonn i matrisen $\mathbf{H}$. Låt oss nu formulera ett lämpligt villkor som garanterar att det inte finns kodord av vikten 2. Om det finns ett sådant kodord med exakt två ettor:

$$\begin{array}{c} \begin{matrix} 0 & 0 & \dots & 1 & \dots & 0 & \dots & 1 & \dots & 0 \end{matrix} \\ \begin{bmatrix} * & * & \dots & a & \dots & * & \dots & a' & \dots & * \\ * & * & \dots & b & \dots & * & \dots & b' & \dots & * \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ * & * & \dots & x & \dots & * & \dots & x' & \dots & * \end{bmatrix} \end{array}$$

så måste summan av kolonnerna under de två ettorna vara lika med nollkolonnen. Med andra ord är vikten av alla kodord skild från 2 om summan av två godtyckliga kolonner inte är nollkolonnen. Detta villkor kan formuleras enklare. Vi har $a + a' = 0$ exakt då $a = a'$ (ty $a' = -a'$). Summan av två kolonner är alltså nollkolonnen exakt då dessa kolonner är lika. Nu kan vi formulera våra slutsatser.

(12.11) Sats. *En binär matris* $\mathbf{H}$ *definierar en kod vars vikt är minst* 3 *då och endast då alla kolonner i* $\mathbf{H}$ *är olika och ingen av dem är nollkolonnen.*

Med den kunskapen kan vi nu konstruera Hammingkoderna. Tag t ex matrisen

$$\mathbf{H_3} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Kolonnerna i $\mathbf{H_3}$ är alla möjliga sekvenser av tre stycken 0 och 1 utom nollsekvensen. Den kod som består av alla lösningar till motsvarande ekvationssystem är just en Hammingkod. Låt oss skriva ut alla kodord. Ekvationssystemet ser ut så här:

$$\begin{aligned} x_4 + x_5 + x_6 + x_7 &= 0, \\ x_2 + x_3 + x_6 + x_7 &= 0, \\ x_1 + x_3 + x_5 + x_7 &= 0. \end{aligned}$$

Man får lätt

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7, \\ x_2 &= x_3 + x_6 + x_7, \\ x_4 &= x_5 + x_6 + x_7. \end{aligned} \tag{12.12}$$

så att x_3, x_5, x_6, x_7 kan väljas godtyckligt som 0 eller 1, medan x_1, x_2 och x_4 därefter kan beräknas. På så sätt har man 16 kodord:

(12.13)

x_3	x_5	x_6	x_7	$\longmapsto$	x_3	x_5	x_6	x_7	x_1	x_2	x_4
0	0	0	0	$\longmapsto$	0	0	0	0	0	0	0
0	0	0	1	$\longmapsto$	0	0	0	1	1	1	1
0	0	1	0	$\longmapsto$	0	0	1	0	0	1	1
0	0	1	1	$\longmapsto$	0	0	1	1	1	0	0
0	1	0	0	$\longmapsto$	0	1	0	0	1	0	1
0	1	0	1	$\longmapsto$	0	1	0	1	0	1	0
0	1	1	0	$\longmapsto$	0	1	1	0	1	1	0
0	1	1	1	$\longmapsto$	0	1	1	1	0	0	1
1	0	0	0	$\longmapsto$	1	0	0	0	1	1	0
1	0	0	1	$\longmapsto$	1	0	0	1	0	0	1
1	0	1	0	$\longmapsto$	1	0	1	0	1	0	1
1	0	1	1	$\longmapsto$	1	0	1	1	0	1	0
1	1	0	0	$\longmapsto$	1	1	0	0	0	1	1
1	1	0	1	$\longmapsto$	1	1	0	1	1	0	0
1	1	1	0	$\longmapsto$	1	1	1	0	0	0	0
1	1	1	1	$\longmapsto$	1	1	1	1	1	1	1.

Icke-oväntat får man en kod som korrigerar ett fel. Den koden är en av de mest kända och vanliga i samband med olika datortillämpningar. För att testa den låt oss anta att man vill sända 10 000 signaler och ha möjligheten att korrigera ett fel. En vanlig repetitionskod kräver då att man upprepar varje signal 3 gånger dvs man måste sända 30 000 signaler. Hur använder man Hammingkoden? Man kan dela de 10 000 signalerna i 2 500 paket om 4 signaler vart. Därefter kan man översätta varje sekvens av 4 signaler till en sekvens av 7 signaler i enlighet med vår konstruktion dvs enligt (12.13). Man får sammanlagt 17 500 signaler. På det sättet kan man korrigera ett fel och samtidigt sända bara 7 500 extra signaler (mot 20 000 för repetitionskoden).

Kodorden i koden (12.13) är vektorer $(a, b, c, d, a+b+d, a+c+d, b+c+d)$ där vi har betecknat $x_3 = a, x_5 = b, x_6 = c, x_7 = d$ och räknat ut de sista koordinaterna enligt (12.12). a, b, c, d kallas ofta **informationssymboler** – de svarar entydigt mot olika meddelanden. De sista tre koordinaterna utgör **kontrollsymboler** (eller **checksymboler**). De finns för att möjliggöra felkorrigering. En liknande situation är väl-känd från våra personnummer – den sista siffran är en kontrollsiffra som gör det möjligt att ibland upptäcka ett felaktigt personnummer.

Hammingkoder kan naturligtvis konstrueras för en godtycklig kolonnlängd. Med andra ord kan man för varje n definiera Hammingmatrisen $\mathbf{H}_n$ vars kolonner är alla möjliga $\neq \mathbf{0}$ vektorer i $\mathbb{Z}_2^n$ (tidigare har vi använt $\mathbf{H}_3$). Rent allmänt är antalet kolonner i matrisen $\mathbf{H}_n$ lika med $2^n - 1$. Ett praktiskt sätt att generera alla kolonner är att skriva talen 1 till $2^n - 1$ som binära tal. En sådan matris definierar en kod som har $2^n - n - 1$ informationssymboler och n kontrollsymboler. Det finns nämligen exakt n kolonner som innehåller precis en etta – dessa kolonner motsvarar kontrollsymbolerna, däremot de övriga variablerna kan väljas godtyckligt. Eftersom antalet kolonner är $2^n - 1$ så är antalet informationssymboler $(2^n - 1) - n$. En sådan allmän konstruktion (för godtyckliga n) gavs av M.J.E. Golay samma år då Hamming publicerade koden för $n = 3$ (se T.M. Thompsons bok "From error-correcting codes through

sphere packings to simple groups", The Carrus Mathematical Monographs, MAA, 1983, för en mycket intressant diskussion av den tidiga kodningsteorins historia).

Efter Golays och Hamming grundläggande arbeten utvecklades den algebraiska kodningsteorin mycket intensivt. Under 50- och 60-talet konstruerades många nya klasser av högeffektiva algebraiska koder. Hammingkoderna korrigerar 1 fel. För praktiska tillämpningar är det ofta tillräckligt. Men det finns situationer då man vill ha bättre koder t ex vid överföring av bilder på stora avstånd. Därför betraktar man koder som rättar större antal fel.

(12.14) Definition. Man säger att en kod C **detekterar** t fel om ett mottaget ord ses inte vara ett kodord när högst t fel har inträffats vid sändningen. Koden **korrigerar** t fel om även när högst t fel inträffats det mottagna ordet kan avkodas till ordet som sänts. □

(12.15) Sats. *En kod C detekterar t fel om $d(C) > t$ och korrigerar t fel om $d(C) > 2t$.*

Bevis. Låt $d(C) > t$. Antag att man sänder $\boldsymbol{a}$ och tar emot $\hat{\boldsymbol{a}}$ varvid antalet fel i kanalen är högst t. Då är $d(\boldsymbol{a}, \hat{\boldsymbol{a}}) \leq t$ så att $\hat{\boldsymbol{a}}$ inte är ett kodord eller $\hat{\boldsymbol{a}} = \boldsymbol{a}$. Det är inte möjligt att $\hat{\boldsymbol{a}}$ är ett annat kodord, ty avståndet mellan olika kodord är minst $t + 1$. Mottagaren kan alltså upptäcka att högst t fel inträffats vid sändningen ($\hat{\boldsymbol{a}}$ är ett kodord betyder inga fel; $\hat{\boldsymbol{a}}$ är inte ett kodord betyder att det föreligger minst 1 och högst t fel).

Låt nu $d(C) > 2t$ och antag som tidigare att man sänder ut $\boldsymbol{a}$, tar emot $\hat{\boldsymbol{a}}$ och antalet fel i kanalen är högst t så att $d(\boldsymbol{a}, \hat{\boldsymbol{a}}) \leq t$. Nu kan man påstå att $\boldsymbol{a}$ är det kodord som ligger närmast $\hat{\boldsymbol{a}}$. Hade det funnits ett annat kodord $\boldsymbol{b}$ sådant att $d(\hat{\boldsymbol{a}}, \boldsymbol{b}) \leq t$ så skulle det innebära att

$$d(\boldsymbol{a}, \boldsymbol{b}) \leq d(\boldsymbol{a}, \hat{\boldsymbol{a}}) + d(\hat{\boldsymbol{a}}, \boldsymbol{b}) \leq 2t,$$

vilket strider mot antagandet $d(C) > 2t$. □

Huvudprincipen vid avkodning i kodningsteorin är förutsättningen att om den mottagna följden är $\hat{\boldsymbol{a}}$ och ett kodord $\boldsymbol{a}$ ligger närmast $\hat{\boldsymbol{a}}$ så avkoderar man $\hat{\boldsymbol{a}}$ som $\boldsymbol{a}$. Den principen följer sunt förnuft men den bekräftas även av beräkningar av sannolikheten – om $\hat{\boldsymbol{a}}$ är den mottagna följden och $d(\boldsymbol{a}, \hat{\boldsymbol{a}}) < d(\boldsymbol{b}, \hat{\boldsymbol{a}})$, där $\boldsymbol{a}, \boldsymbol{b}$ är kodord så är sannolikheten att det ursprungliga kodordet var $\boldsymbol{a}$ större än sannolikheten att det var $\boldsymbol{b}$ (den beräkningen är mycket enkel). Villkoret $d(C) > 2t$ i satsen ovan garanterar alltså att man kan genomföra avkodning i enlighet med avkodningsprincipen ovan om antalet fel vid sändningen är $\leq t$.

Hur kan man konstruera koder som korrigerar större antal fel? Vi skall begränsa oss till linjära koder därför att koder utan algebraisk struktur (med en algebraisk strukturen menar vi att summan av två kodord är ett kodord igen) är betydligt svårare att hantera.

Lika enkelt som i sats (12.11) kan vi konstatera att en konstruktion av en linjär kod som rättar t fel innebär en konstruktion av en matris $\mathbf{H}$ med följande egenskap:

(12.16) Sats. *Alla lösningar till ekvationen $\mathbf{H}\boldsymbol{x} = \mathbf{0}$, där $\mathbf{H}$ är en matris, bildar en kod som korrigerar t fel om $\mathbf{H}$ saknar nollkolonner och summan av högst $2t$ kolonner aldrig ger en nollvektor.*

Observera att summan av två kolonner är en nollvektor exakt då dessa kolonner är lika.

Bevis. Som i sats (12.11) har man $w(\boldsymbol{x}) \geq 2t+1$ då $\mathbf{H}\boldsymbol{x} = \mathbf{0}$ och $\boldsymbol{x} \neq \mathbf{0}$ därför att likheten $\mathbf{H}\boldsymbol{x} = \mathbf{0}$ innebär att summan av de kolonner som svarar mot nollskilda koordinater i $\boldsymbol{x}$ är lika med nollvektorn. Alltså är vikten av koden bestående av alla vektorer $\boldsymbol{x}$ minst $2t+1$, vilket innebär att koden rättar t fel. □

Tyvärr är det inte så lätt att konstruera matriser $\mathbf{H}$ med den egenskap som krävs i satsen – Hammingsmatriserna är ett sällsynt undantag. Men det finns många andra mycket intressanta konstruktioner. I senare delen av kursen kommer vi att bekanta oss med en sådan konstruktion som leder till en mycket viktig klass av s.k. BCH-koder. I detta kapitel stannar vi dock vid Hammingkoder och några enkla konstruktioner av gruppkoder i samband med övningar.

Vi skall avsluta detta kapitel med två praktiska problem i samband med kodning och avkodning. Det är inte svårt att lösa matrisekvationer $\mathbf{H}\boldsymbol{x} = \mathbf{0}$, men om matrisen $\mathbf{H}$ är stor kan problemet vara besvärligt. Det finns en mycket enklare metod som gör det möjligt att alstra linjära koder. Låt $\mathbf{G}$ vara en binär $(n \times k)$-matris. Om $\boldsymbol{a}$ är en vektor i $\mathbb{Z}_2^k$ så är $\mathbf{G}\boldsymbol{a}$ en vektor i $\mathbb{Z}_2^n$. På det sättet får man en linjär kod i $\mathbb{Z}_2^n$. Man säger att matrisen $\mathbf{G}$ är en **generatormatris** för den koden.

(12.17) Sats. *Låt $\mathbf{G}$ vara en binär $(n \times k)$-matris. Då bildar alla vektorer $\mathbf{G}\boldsymbol{a}$ med $\boldsymbol{a} \in \mathbb{Z}_2^k$ en linjär kod C i $\mathbb{Z}_2^n$.*

Bevis. Om $\mathbf{G}\boldsymbol{a}, \mathbf{G}\boldsymbol{b} \in C$ så $\mathbf{G}\boldsymbol{a} + \mathbf{G}\boldsymbol{b} = \mathbf{G}(\boldsymbol{a}+\boldsymbol{b}) \in C$. □

(12.18) Exempel. (a) Låt

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Kodorden är $\mathbf{G}\boldsymbol{a}$ där $\boldsymbol{a} = (a_1, a_2)$, $a_i \in \mathbb{Z}_2$ dvs:

$$\begin{array}{rcl} 00 & \mapsto & 000, \\ 01 & \mapsto & 011, \\ 10 & \mapsto & 101, \\ 11 & \mapsto & 110. \end{array}$$

(b) Låt

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Nu är kodorden $\mathbf{G}\boldsymbol{a}$, där $\boldsymbol{a} = (a_1, a_2, a_3)$, $a_i \in \mathbb{Z}_2$ dvs

$$\begin{array}{llll} 000 & \mapsto & 000000 & 100 \mapsto 100110 \\ 001 & \mapsto & 001101 & 101 \mapsto 101011 \\ 010 & \mapsto & 010011 & 110 \mapsto 110101 \\ 011 & \mapsto & 011110 & 111 \mapsto 111000 \end{array}$$

□

(12.19) Anmärkning. Man kan visa utan större problem att varje gruppkod har en generatormatris. Se vidare övning 12.9 som visar hur man konstruerar en generatormatris $\mathbf{G}$ då paritetsmatrisen $\mathbf{H}$ är given. Låt oss observera att generatormatrisen ger en mycket enkel möjlighet att skriva ut kodorden – alla kodord fås som produkter $\mathbf{G}\boldsymbol{a}$. Om man t ex vill alstra koden med hjälp av en dator behöver man endast lagra matrisen $\mathbf{G}$. Låt oss också observera att om

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{bmatrix},$$

där $\mathbf{g}_i = (a_{1i}, a_{2i}, \dots, a_{ni})$ så är $\mathbf{G}\boldsymbol{a} = a_1\mathbf{g}_1 + a_2\mathbf{g}_2 + \ldots + a_k\mathbf{g}_k$ då $\boldsymbol{a} = (a_1, a_2, \dots, a_k)$. Detta innebär att koden med generatormatrisen $\mathbf{G}$ består av alla linjärkombinationer av kolonnerna i matrisen $\mathbf{G}$. Om kolonnerna i denna matris är linjärt oberoende över $\mathbb{Z}_2$ dvs en linjärkombination $\mathbf{G}\boldsymbol{a} = a_1\mathbf{g}_1 + a_2\mathbf{g}_2 + \ldots + a_k\mathbf{g}_k = \mathbf{0}$ endast då $a_1 = a_2 = \cdots = a_k = 0$, så säger man att matrisen $\mathbf{G}$ genererar en kod av dimensionen k. En sådan kod består av 2^k vektorer därför att alla vektorer $\mathbf{G}\boldsymbol{a}$ då $\boldsymbol{a} \in \mathbb{Z}_2^k$ är olika. En gruppkod $C \subseteq \mathbb{Z}_2^n$ kallar man för en (k, n)-kod. Ofta betraktar man matriser $\mathbf{G}$ i vilka de första k raderna utgör en $(k \times k)$-enhetsmatris. Då säger man att matrisen $\mathbf{G}$ är **normaliserad**. Det är en fördel att ha en normaliserad matris därför att kodorden $\mathbf{G}\boldsymbol{a}$ har då informationssymbolerna på de första k platserna och kontrollsymbolerna (checksymbolerna) på de sista $n - k$.

□

I praktiska sammanhang är det oftast mycket viktigt att kunna relativt lätt genomföra avkodning. Hammingkoderna är mycket enkla att hantera just i detta avseende. Innan vi visar hur man kan genomföra avkodningen av dessa koder låt oss ägna några ord åt det allmänna avkodningsproblemet.

Antag att man man har en linjär kod C som korrigerar t fel och består av alla lösningar till $\mathbf{H}\boldsymbol{x} = \mathbf{0}$. Antag vidare att det verkligen inträffar högst t-fel vid sändningen. Antag att man sänder ett kodord $\boldsymbol{a}$ och tar emot en vektor $\hat{\boldsymbol{a}}$, där $\hat{\boldsymbol{a}} = \boldsymbol{a} + \boldsymbol{\varepsilon}$. Vi skall kalla $\boldsymbol{\varepsilon}$ för **felvektor**. Om vi känner denna vektor då kan vi genomföra avkodning på ett mycket enkelt sätt:

(12.20) $$\boldsymbol{a} = \hat{\boldsymbol{a}} + \boldsymbol{\varepsilon}\,,$$

ty $\hat{\boldsymbol{a}} + \boldsymbol{\varepsilon} = \boldsymbol{a} + \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} = \boldsymbol{a}$ (tänk på det att $\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} = \mathbf{0}$). Hur kan man bestämma felvektorn med ledning av $\hat{\boldsymbol{a}}$?. Vi skall visa att man kan göra det genom att beräkna $\mathbf{H}\hat{\boldsymbol{a}}$. Vi har

$$\mathbf{H}\hat{\boldsymbol{a}} = \mathbf{H}(\boldsymbol{a} + \boldsymbol{\varepsilon}) = \mathbf{H}\boldsymbol{\varepsilon},$$

ty $\mathbf{H}\boldsymbol{a} = \mathbf{0}$ eftersom $\boldsymbol{a}$ är ett kodord.

(12.21) Definition. Om C är en linjär kod bestående av alla lösningar till $\mathbf{H}\boldsymbol{x} = \mathbf{0}$ och $\hat{\boldsymbol{a}}$ är den mottagna vektorn då $\boldsymbol{a}$ har sänts så kallas vektorn $\mathbf{H}\hat{\boldsymbol{a}}$ för **felsyndrom**. □

Observera att felsyndrom endast beror på felvektorn $\boldsymbol{\varepsilon}$. Det visar sig att högst t fel i den mottagna vektorn ger en möjlighet att rekonstruera kodordet med hjälp av felsyndromet:

(12.22) Sats. *Låt C vara en linjär kod som korrigerar t fel. Om vid transmission med hjälp av C inträffar högst t fel så ger olika felvektorer olika felsyndrom.*

Detta innebär att man t ex kan tabulera alla felsyndrom som svarar mot olika felvektorer av vikt högst t. Ur en sådan tabell kan man avläsa felvektorn med ledning av felsyndromet. På det sättet kan man genomföra felkorrigering (dvs avkodning) enligt (12.20).

Bevis. Om $\boldsymbol{\varepsilon} \neq \boldsymbol{\varepsilon}'$ är två olika felvektorer, så måste felsyndromen $\mathbf{H}\boldsymbol{\varepsilon}$ och $\mathbf{H}\boldsymbol{\varepsilon}'$ vara olika ty likheten $\mathbf{H}\boldsymbol{\varepsilon} = \mathbf{H}\boldsymbol{\varepsilon}'$ ger $\mathbf{H}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}') = \mathbf{0}$, vilket är omöjligt därför att vikten av $\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}' \neq \mathbf{0}$ är högst $2t$. □

(12.23) Exempel. Nu kan vi visa hur man kan genomföra felkorrigering med hjälp av Hammingkoder. Vi skall begränsa oss till koden $\mathbf{H}_3$ som vi betraktade tidigare. Felvektorerna är 0000000 (inget fel alls — den mottagna vektorn är ett kodord) och alla vektorer i $\mathbb{Z}_2^7$ med exakt en etta och sex nollor. Låt oss beräkna felsyndromen $\mathbf{H}_3\boldsymbol{\varepsilon}$:

$\boldsymbol{\varepsilon}$:		$\mathbf{H}_3\boldsymbol{\varepsilon}$:
0000000	$\longmapsto$	000
1000000	$\longmapsto$	001
0100000	$\longmapsto$	010
0010000	$\longmapsto$	011
0001000	$\longmapsto$	100
0000100	$\longmapsto$	101
0000010	$\longmapsto$	110
0000001	$\longmapsto$	111

Nu observerar vi att felsyndromen är helt enkelt kolonnerna i matrisen $\mathbf{H}_3$ – felet i 1:a koordinaten ger första kolonnen i denna matris, felet i 2:a koordinaten ger andra kolonnen osv. För att korrigera eventuella fel multipliceras den mottagna vektorn med $\mathbf{H}_3$. Om man får nollvektorn är transmissionen korrekt – den mottagna vektorn är ett kodord. Om man får k-te kolonnen i matrisen $\mathbf{H}_3$ är felet i k-te koordinaten och man måste addera 1 till denna för att få ett kodord. □

Tyvärr är avkodningsproblemet långt ifrån lika enkelt för andra kodklasser.

Varje kod C har viktiga parametrar – längden av kodorden, deras antal och antalet fel som koden korrigerar. Om $C \subseteq \mathbb{Z}_2^n$ så består C av en del av alla 2^n vektorer. Om C är linjär och har generatormatrisen med k linjärt oberoende kolonner så har koden 2^k kodord, där $k \leq n$. Dess dimension är då lika med k.

(12.24) Definition. Låt C vara en kod i $\mathbb{Z}_2^n$ av vikten $w(C) = d$ med $|C|$ kodord. Talet $R = \frac{\log_2 |C|}{n}$ (R = "rate") kallas **hastigheten** av koden och talet $\frac{d}{n}$ **relativa vikten** av koden. Om koden är linjär av dimensionen k så är $R = \frac{k}{n}$ (ty $|C| = 2^k$). □

(12.25) Anmärkning. Talen

$$x(C) = \frac{w(C)}{n}, \qquad\qquad y(C) = \frac{\log_2 |C|}{n}$$

ligger bägge i intervallet [0,1]. När man konstruerar koder är man ofta intresserad av att dessa två tal är så nära 1 som möjligt. Ju närmare 1 desto fler fel korrigerar koden och desto mer meddelenden kan den överföra. Men det finns en klar motsättning mellan dessa strävanden. En ganska intensiv matematisk forskning är inriktad på en undersökning av mängden av alla punkter $(x(C), y(C))$ i kvadraten $[0,1] \times [0,1]$ som svarar mot alla möjliga koder. Låt oss observera att för Hammingkoderna är $x(C_n) = 1/(2^n - 1)$ och $y(C_n) = (2^n - n - 1)/(2^n - 1)$. Alltså $(x(C_n), y(C_n)) \to (0,1)$ då $n \to \infty$. Det var först i mitten av 60-talet då en rysk matematiker V.D.Goppa konstruerade sekvenser av koder C_n sådana att både $x(C_n)$ och $y(C_n)$ konvergerar mot en punkt (x, y) med $xy \neq 0$. Dessa problem har en mera teoretisk karaktär och sysselsätter många matematiker. De kräver ofta mycket djupa kunskaper i ämnet. □

Historiska anmärkningar. Som början av kodningsteorin kan man betrakta arbeten av tre amerikanska matematiker: C.E. Shannon ("A mathematical theory of communication" från 1948), M.J.E. Golay ("Notes on digital coding" från 1949) och R.W. Hamming ("Error detecting and error correcting codes" från 1950). Shannons arbete lade grunden för matematisk informationsteori, och även om hans huvudresultat tillhör s k statistisk kodningsteori (i den spelar sannolikhetslära viktigare roll än algebra) hade det en mycket stor betydelse för utvecklingen av den algebraiska kodningsteorin. Shannon och Hamming sysslade med telekommunikation vid Bell Laboratories i New Jersey (USA), däremot ledde Golays väg till koder från spektrografi. Kodningsteorin skapades alltså direkt från början som en matematisk teori med

syfte att lösa mycket konkreta tekniska problem. Under de år som har gått från 1948 publicerades flera tusen väsentliga bidrag till den algebraiska kodningsteorin, och den matematiska apparat som man använder för att lösa dess problem omfattar nu mycket avancerade teorier som ofta har en ganska abstrakt karaktär. Samtidigt utvidgas tillämpningsområdena av den algebraiska kodningsteorin, speciellt i samband med utvecklingen av datatekniken.

ÖVNINGAR

12.1. Kodera tre meddelanden A, B, C, så att minimiavståndet mellan kodorden blir

(a) 3, (b) 4, (c) 5.

12.2. Låt $\mathbf{H}$ vara en kontrollmatris för en kod C. Skriv ut alla kodord och bestäm antalet fel som koden korrigerar då:

$$\text{(a) } \mathbf{H} = \begin{bmatrix} 101100 \\ 110010 \\ 011001 \end{bmatrix}, \qquad \text{(b) } \mathbf{H} = \begin{bmatrix} 1010 \\ 1101 \end{bmatrix}, \qquad \text{(c) } \mathbf{H} = \begin{bmatrix} 011000 \\ 110100 \\ 010010 \\ 100001 \end{bmatrix}.$$

12.3. Skriv ut alla kodord samt bestäm vikten och hastigheten för den kod som har generatormatrisen $\mathbf{G}$ då

$$\text{(a) } \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad \text{(b) } \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \text{(c) } \mathbf{G} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\text{(d) } \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \text{(e) } \mathbf{G} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Hur många fel detekterar och korrigerar dessa koder?

12.4. Låt $\boldsymbol{a}, \boldsymbol{b}, \mathbf{c}, \in \mathbb{Z}_2^n$. Visa att

(a) $d(\boldsymbol{a} + \mathbf{c}, \boldsymbol{b} + \mathbf{c}) = d(\boldsymbol{a}, \boldsymbol{b})$.

(b) $w(\boldsymbol{a} + \boldsymbol{b}) \geq |w(\boldsymbol{a}) - w(\boldsymbol{b})|$.

12.5. För $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_2^n$, $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$, $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ definierar man

$$\boldsymbol{ab} = (a_1 b_1, a_2 b_2, \ldots, a_n b_n).$$

Visa att $w(\boldsymbol{a} + \boldsymbol{b}) = w(\boldsymbol{a}) + w(\boldsymbol{b}) - 2w(\boldsymbol{ab})$.

12.6. Låt $C \subseteq \mathbb{Z}_2^n$ vara en kod. Man definierar en ny kod $C' \subseteq \mathbb{Z}_2^{n+1}$ så att för $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in C$ är $\boldsymbol{a'} = (a_1, a_2, \ldots, a_n, s) \in C'$, där $s = a_1 + a_2 + \cdots + a_n$.

Exempel. Om $0 \mapsto 000$, $1 \mapsto 111$ är koden C, så är C' definierad så att $0 \mapsto 0000$ och $1 \mapsto 1111$. Detta innebär att man förlänger orden i C med en symbol som är lika med

summan av de föregående (modulo 2). Som vi ser i detta exempel ökar vikten av den nya koden med 1, så att den upptäcker ett fel mer än C'.

(a) Konstruera koden C' för koderna i övning 12.3. Lägg märke till de situationer då man får en kod med större vikt och till de då man inte har någon vinst.

(b) Visa att om C är en gruppkod, så är C' en gruppkod.

(c) Konstruera en matris som genererar C' då $\mathbf{G}$ är en matris som genererar C.

12.7. Konstruera en kod $C \subseteq \mathbb{Z}_2^6$ som upptäcker 3 fel.

12.8. Låt $C \subseteq \mathbb{Z}_2^n$ vara en gruppkod.

(a) Visa att alla kodord i C med jämn vikt bildar också en gruppkod.

(b) Visa att antingen har alla kodord i C en jämn vikt, eller hälften av kodorden har jämn vikt och den andra hälften har udda vikt.

12.9. (a) Låt $C \subseteq \mathbb{Z}_2^n$ vara en gruppkod bestående av alla lösningar till ekvationen $\mathbf{Hx} = \mathbf{0}$, där $\mathbf{H}$ är en $(k \times n)$-matris. Antag att $\mathbf{H}$ är normaliserad dvs $\mathbf{H} = [\mathbf{P}\ \mathbf{E}]$, där $\mathbf{E}$ är $(k \times k)$-enhetsmatrisen och $\mathbf{P}$ är en $(k \times (n-k))$-matris. Visa att matrisen

$$\mathbf{G} = \begin{bmatrix} \mathbf{E} \\ \mathbf{P} \end{bmatrix}$$

är en generatormatris för koden C. Visa samtidigt att om $\mathbf{G}$ genererar koden så är $\mathbf{H}$ en kontrollmatris för denna kod.

12.10. Bestäm generatormatriser för koderna i övning 12.2.

12.11. Bestäm kontrollmatriser för koderna i uppgift 12.3.

12.12. Låt C vara en kod med en $(n-k \times n)$-kontrollmatris $\mathbf{H}$ och en $(n \times k)$-generatormatris $\mathbf{G}$.

(a) Hur förändras koden då man kastar om raderna eller kolonnerna i kontrollmatrisen? Vad händer då man adderar en linjärkombination av några rader till en annan rad?

(b) Besvara samma frågor som i (a) om generatormatrisen ordet "rad" byts ut mot ordet "kolonn".

Anmärkning. Man skall observera att att alla dessa operationer leder till oväsentliga förändringar av koden.

(c) Anta att koden C har dimensionen k. Visa att med hjälp av operationerna i (a) (respektive (b)) kan $\mathbf{H}$ (respektive $\mathbf{G}$) överföras på en normaliserad form (man säger att $\mathbf{H}$ och $\mathbf{G}$ kan **normaliseras**).

12.13. Låt $C \subseteq \mathbb{Z}_2^7$ vara koden genererad av matrisen

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

(a) Normalisera $\mathbf{G}$.

(b) Bestäm en kontrollmatris och vikten av koden.

(c) Bestäm alla felsyndrom för felvektorer av vikten ≤ 1.

(d) Avkodera följande följder: 0011010, 0001111, 0101100.

(e) Lös samma uppgifter (a) – (c) för koderna i övningen 12.2.

12.14. **Perfekta koder.** En gruppkod kallas **perfekt** om till varje vektor $\mathbf{x} \in \mathbb{Z}_2^n$ existerar exakt ett kodord vars avstånd från $\mathbf{x}$ är högst t. Visa att Hammingkoderna är perfekta.

Anmärkning. Om en kod $C \subseteq Z_2^n$ är perfekt och har dimensionen k, så består den av 2^k kodord. För varje kodord finns

$$1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}$$

olika vektorer som ligger på ett avstånd $\leq t$ från kodordet. Alltså har man identiteten:

$$2^k[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}] = 2^n$$

Den likheten är svår att uppfylla vilket medför att perfekta koder är sällsynta[†]. Det finns bara tre typer av sådana koder[‡]: Hammingkoderna, koderna $C \subseteq \mathbb{Z}_2^{2r+1}$ bestående av $00\ldots0$ och $11\ldots1$ (man upprepar 0 respektive 1 ett udda antal gånger) och en mycket viktig (12, 23)-kod som korrigerar 3 fel. Den konstruerades av M.J.E. Golay år 1949. För (12,23)-Golay koden har man:

$$2^{12}[1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}] = 2^{23}$$

12.15. Låt $C \subseteq \mathbb{Z}_2^n$ vara en gruppkod med 2^k kodord som korrigerar 1 fel. Låt $n = k + r$. Visa att $k \leq 2^r - 1 - r$.

[†]Men även om den är uppfylld så garanterar den inte existensen av en perfekt kod (t ex då $m = 78$, $n = 90$, $t = 2$)

[‡]Satsen bevisades av J.H. van Lint (1971) och A. Tietäväinen (1973).