SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Lineära koder och Reed-Muller koder

av

Magnus Weiderling

2012 - No 14

Lineära koder och Reed-Muller koder

Magnus Weiderling

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Rikard Bögvad

2012

Sammanfattning

Den här uppsatsen handlar om lineära koder, varav det sista kapitlet ägnas helt åt Reed-Muller koder.

När man hör ordet koder kanske man först och främst tänker på kryptering men här är målet bra och säker överföring av information. Koderna används antingen för att informationen ska vara möjlig att skicka eller spara (t.ex. ettor och nollor) eller för att förhindra förlust/förändring av data.

I den enklaste modellen för *kommunikationsöverföring*, vare sig man är i matematikens värld eller utanför, behövs en sändare, en *kommunikationskanal* och en mottagare. Här intresserar vi oss för kommunikationskanalen. Det ska ses som ett allmänt begrepp som kan symbolisera flera olika situationer, exempelvis ett samtal, en datakabel eller en trådlös överföring. Det som är typiskt för att vi ska ha nytta av att använda koder är att det på ett eller annat sätt kan uppstå fel.

Den här uppsatsen riktar in sig på hur man kan koda sitt meddelande innan det passerar kommunikationskanalen för att mottagaren ska kunna få maximalt med information om det sändaren skickade, med så korta meddelanden som möjligt. Som så ofta annars är man nämligen till viss del tvungen att välja mellan tillförlitlighet och tid (kostnad).

Uppsatsen tar upp grundläggande begrepp och algebra i kodteorin inklusive kodning och avkodning. Teorin eller materialet kan självklart angripas på flera olika sätt och i litteraturen finns ofta olika namn för samma matematiska begrepp. I stora drag följer min framställning ordningsföljd och terminologi i boken Coding Theory - A first Course [5]. Speciellt studeras Reed-Muller koder som har många tillämpningar.

Abstract - Linear codes and the Reed-Muller codes

A simple model of communication, both within and around the subject of mathematics, consists of a *transmitter*, a *communication channel* and a *receiver*. I will particularly direct my focus to the communication channel. In real life it translates into many different things such as phone calls, data cables or wireless transmissions. What is typical is that in one way or another, errors can occur. That is, the message sent is different from what is received. We usually call these disturbances for noise.

This essay focuses on how to encode messages before they pass the communication channel so that the receiver is able to get the maximum amount

of information. As in so many other cases, there is a trade-off between security and time (cost).

We addresses the basic algebra and fundamental concepts of coding theory including encoding and decoding. Special interest is given to the old Reed-Muller code which has many applications.

Förord

Denna uppsats utgör ett examensarbete i matematik om 15 högskolepoäng vid Stockholms universitet. Arbetet med uppsatsen har varit mycket roligt och lärorikt. Jag vill framförallt tacka min handledare Professor Rikard Bøgvad på Stockholms Universitet för idéer kring genomförandet och stort stöd under arbetets gång.

Magnus Weiderling, Aktuarieprogrammet vid Stockholms Universitet, september 2011

Innehåll

1 Inledning

Hanteringen av information och data blir allt viktigare samhället. Två problem som hör samman med *överföring* (och lagring) av information är dels hur man ska skydda informationen från obehöriga och dels hur man undviker att den ändras av misstag (eller försvinner). Jag ska koncentrera mig på det senare.

Vid överföring av information finns normalt en *sändare*, en *mottagare* och en *kommunikationskanal* däremellan. När överföringen är av sådan typ att data kan förstöras på vägen säger man att den är utsatt för *brus*. Principer för dataöverföring är mycket lika de för *datakomprimering*.

När det gäller att få ett meddelande korrekt överfört är en mycket enkel förbättring att skicka all information två gånger. Men eftersom det tar dubbelt så lång tid (och kostar mer) är det inte särskilt effektivt. Istället försöker man hitta bättre sätt att koda data innan man skickar den. Genom att göra det på ett genomtänkt sätt kan man faktiskt nå fördelar som att koden automatiskt upptäcker och rättar eventuella fel som uppstår.

Vi kan titta på ett enkelt exempel. Digital överföring är i grunden ettor och nollor, det vill säga *binär*. Låt oss därför anta att vi bara kan skicka ettor och nollor och att vi har en sändare som vill skicka några bokstäver eller ett ord. För att klara av överföringen behöver vi koda vårt meddelande.

Vi tänker oss en enkel kod så här

Gul	=	00
Röd	=	01
Blå	=	10
Grön	=	11

Antag att vi vill skicka "Gul" och skickar därför 00 men på vägen till mottagaren blir meddelandet ändrat och mottas som 01. Mottagaren kommer inte upptäcka att något blev fel utan tolkar det som "Röd".

Ett enkelt sätt att förhindra detta är att lägga till en siffra till varje kod. Till exempel på detta sätt

Gul	=	000
Röd	=	011
Blå	=	101
Grön	=	110

Om man återigen vill skicka "Gul" och precis ett tecken blir fel så kommer mottagaren få koden 001, 010 eller 100. Eftersom dessa inte betyder något kommer mottagaren veta att något blivit fel och kan be om en *omsändning*. Koden *detekterar* alltså ett fel (men inte två). För att kunna rätta fel behövs ytterligare tecken i koden.

De flesta dataenheter för masslagring (hårddiskar etc) använder någon typ av *felrättande kod* för att förhindra förlust av data.

Ett vardagsexempel på en variant av feldetekterande kod är svenska personnummer. Av de tio siffrorna är den sista en kontrollsiffra. Har man ändrat en av de första nio siffrorna, men inte den tionde blir personnumret i regel ogiltigt och felet upptäcks och kan rättas till.

Den här uppsatsen är indelad i fyra huvudkapitel. I det första av dessa kapitel, med nummer 2, tar jag upp grundläggande kodbegrepp. Nästa kapitel koncentras till de viktiga lineära koderna. Kapitel 4 handlar om kodernas egenskaper. Kapitel 5 behandlar Reed-Muller koden.

2 Definitioner

2.1 Kodanvändning

Problemet som kodteorin försöker lösa består i att uttrycka mycket information med få tecken. Det är ganska naturligt att tänka sig en överföringskostnad eller tidsfördröjning per tecken som gör att man vill hålla ner antalet.

Från exemplet i inledningen inser man att det är en fördel om de kodord man använder är så olika varandra som möjligt. Vad det innebär och hur olika de behöver vara kommer att besvaras längre fram. Vi ska också titta på matematiken som gör detta möjligt och ta reda på om det finns andra sätt att förbättra koden på något annat sätt.

I mina exempel har jag valt att begränsa mig till *binära koder* (se vidare Exempel 1). Dels för att det blir tydligare, dels för att de är så pass vanliga. Vid behov kan man enkelt utvidga antalet symboler.

2.2 Kodalfabet

När man skriver en text på svenska finns det 29 bokstäver att välja på. Detta är att likna vid vårt *kodalfabet* och bokstäverna är i det fallet våra *kodsymboler*. Kodsymboler används till att bilda *kodord*. Till skillnad mot i det svenska språket kommer alla ord här att vara lika långa. Närmare bestämt av längden n.

Definition 1 (Kodalfabet)**.** *Låt $A = \{s_1, s_2, \ldots, s_q\}$ vara en ändlig mängd med q element. Vi kallar mängden A för kodalfabet och dess element för kodsymboler.*

(i) Ett $q-$ärt ord av längd n i A är en följd $\boldsymbol{x} = x_1x_2\ldots x_n$ där varje $x_i \in A$ för alla i, $1 \leq i \leq n$. Ekvivalent kan $\boldsymbol{x}$ betraktas som vektorn $(x_1, \ldots, x_n)$

(ii) En $q-$är (block)kod av längd n på A är en icke-tom mängd C av $q-$ära ord, alla med samma längd n.

(iii) Ett element i koden C kallas för kodord i C.

(iv) Antal kodord i C kallas storleken av C och skrivs med $|C|$ alternativt M.

(v) Informationsgraden, $R(C)$, i C med längd n, definieras som $\frac{log_q\ |C|}{n}$

Antalet kodord i C kan, av kombinatoriska skäl, som mest vara $M = q^n$, för givna q och n. Detta kan också skrivas som $n = log_q\ M = log\ M/log\ q$.

Informationsgraden (eng. Rate) är ett mått på effektiviteten per kodsymbol i koden C. Den varierar mellan de två extremerna 0 och 1. Om varje kombinatoriskt möjligt kodord ($q-$ärt med längd n) används i koden är informationsgraden 1. Men då finns inget utrymme kvar för felkorrigering eller feldetektering. Se vidare Exempel 18.

Exempel 1. *En kod i alfabetet $A = \mathbb{F}_2 = \{0, 1\}$ kallas binär kod. Kodsymbolerna är bara 0 och 1 så $q = 2$. Två exempel på binära koder är*

$C_1 = \{00, 01, 10, 11\}$*, som har längd $n = 2$ och storlek $|C| = 4$*

$C_2 = \{000, 011, 101, 110\}$*, som har längd $n = 3$ och storlek $|C| = 4$*

En kod i $\mathbb{F}_3 = \{0, 1, 2\}$ kallas *tertiär kod* och en kod i $\mathbb{F}_4$ *kvartär kod*. Beteckningen $q-$är i Definition 1 är bara ett gemensamt namn för binär, tertiär och så vidare.

För en ändlig kropp (Galoiskropp, eng. field, Beachy [1] sid 160) med q element använder vi beteckningen $\mathbb{F}_q$. En kropp är en kommutativ ring, där varje element skiljt från 0 har en invers. Man kan visa att då q är primtal så finns en och upp till isomorfi bara en ändlig kropp $\mathbb{F}_q$ (Xing [5] sid 26). Se vidare Kapitel 3.

I Kapitel 2.1 berördes nyttan av att jämföra hur lika (eller olika) två kodord är. *Hammingavståndet* är ett sådant mått. Vi gör en definition.

Definition 2 (Hammingavstånd)**.** *Låt $\boldsymbol{x}$ och $\boldsymbol{y}$ vara ord med samma längd n i kodalfabetet A. Hammingavståndet från $\boldsymbol{x}$ till $\boldsymbol{y}$ skrivs $d(\boldsymbol{x},\boldsymbol{y})$ och är antalet tecken där $\boldsymbol{x}$ och $\boldsymbol{y}$ skiljer sig åt.*

Om $\boldsymbol{x} = x_1 \dots x_n$ och $\boldsymbol{y} = y_1 \dots y_n$ så gäller

$$d(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} d(x_i, y_i)$$

där varje x_i och y_i ses som ord av längd 1 och

$$d(x_i,y_i) = \begin{cases} 1 & \text{om } x_i \neq y_i \\ 0 & \text{om } x_i = y_i \end{cases}$$

Det minsta av alla Hammingavstånd i en kod kallas för *minimumavståndet*.

Definition 3 (minimumavstånd)**.** *För en kod C som består av minst två ord definieras minimumavståndet i C, skrivs $d(C)$, som*

$$d(C) = min\ \{d(\boldsymbol{x},\boldsymbol{y}) : \boldsymbol{x},\boldsymbol{y} \in C,\ \boldsymbol{x} \neq \boldsymbol{y}\}$$

Exempel 2. *Om vi tar koden från Exempel 1,* $C_1 = \{00, 01, 10, 11\}$*, och låter* $x = 00$ *och* $y = 11$ *så får vi att Hammingavståndet från* x *till* y *är* $d(x, y) = 1 + 1 = 2$ *och minimumavståndet i* C_1 *är* $d(C_1) = 1$

Det kommer visa sig att en kods minimumavstånd hör väl samman med dess förmåga att upptäcka och korrigera fel (se Sats 2 och 3) En kod som har längd n, storlek M och minimumavstånd d brukar kallas för $(n, M, d)-$kod. Talen n, M och d kallas tillsammans för kodens *parametrar*.

Nära förknippat med Hammingavståndet är *Hammingvikten*. Vi gör en definition av den också.

Definition 4 (Hammingvikt)**.** *Hammingvikten av* $\boldsymbol{x}$ *skrivs wt(*$\boldsymbol{x}$*) och definieras som antalet koordinater skilda från noll i* $\boldsymbol{x}$*, det vill säga*

$$wt(\boldsymbol{x}) = d(\boldsymbol{x},\boldsymbol{0})$$

Exempel 3. *För koden* $C = \{000, 010, 111\}$ *gäller att* $wt(010) = 1$

När man skickar koder är det av vikt under vilka förutsättningar de skickas. Alltså vilka regler som gäller för överföringen eller kommunikationskanalen. Ett viktigt antagande är att sannolikheten för eventuella fel i en viss sändning är oberoende av tidigare sändningar.

Definition 5 (BSC/Kommunikationskanal)**.** *En* $q-$*är symmetrisk kommunikationskanal är en minneslös kanal som har ett kodalfabet av storlek* q *så att*

(i) varje symbol som sänds har samma sannolikhet p*, där* $p < \frac{1}{2}$*, att tas emot fel.*

(ii) om en symbol mottas fel så är de $q - 1$ *möjliga felen lika sannolika.*

Speciellt gäller att den binära symmetriska kanalen (BSC), med kodalfabetet $\{0, 1\}$*, har felsannolikhet*

$$\mathrm{P}\ (1\ \textit{mottaget} \mid 0\ \textit{skickat}) = \mathrm{P}\ (0\ \textit{mottaget} \mid 1\ \textit{skickat}) = p$$

Det finns satser längre fram i den här texten som bara är giltiga om kommunikationskanalen är av BSC typ.

2.3 Maximum-likelihoodavkodning

För att en kod ska fylla sin funktion måste den kunna avkodas. Det finns flera metoder. En av dem är *Maximum-likelihoodavkodning* som går till som följer. Antag att ett kodord från en kod C skickats via en BSC och vi känner till *felsannolikheten*, p. Om ett ord M_r (received) mottas, kan vi beräkna sannolikheterna

$$\text{P}(M_r \text{ mottaget } |c \text{ skickat}) \text{ för alla kodord } c \in C$$

Maximum-likelihoodavkodning är en metod som drar slutsatsen att M_c är det som skickats om P(M_r mottaget $|M_c$ skickat) ger den högsta sannolikheten dvs

$$\text{P}(M_r \text{ mottaget } |M_c \text{ skickat}) = max\ \{\text{P}(M_r \text{ mott.}|c \text{ skick.}) \text{ för alla } \boldsymbol{c} \in C\}$$

Det finns två varianter av maximum-likelihoodavkodning. *Fullständig avkodning* och *icke-fullständig avkodning*. Med fullständig menas här att om två kodord har exakt samma sannolikhet väljs ett av dem godtyckligt vid avkodningen. I den icke-fullständiga varianten ber man istället om omsändning av meddelandet.

2.4 Hammingavkodning

En annan generell metod för avkodning är att jämföra hur mycket det mottagna orden skiljer sig från de kodord som är tillåtna i koden man använder (om metoden verkar vara samma sak som ovan beror det på Sats 1). Om det mottagna ordet är ett av de tillåtna kodorden gör man ingen korrigering. Annars antar man att det kodord som avsågs är det som ligger närmast med avseende på Hammingavståndet.

Det vill säga, om ett ord M_r är mottaget så avkodas det till det M_c som minimerar Hammingavståndet $d(M_r, M_c)$ dvs

$$d(M_r, M_c = min\{d(M_r, c) \text{ för alla } \boldsymbol{c} \in C\}$$

Även här skiljer man mellan fullständig och icke-fullständig avkodning. För en BSC och även i en del andra fall ger dessa två metoder för avkodning samma resultat vilket visas i Sats 1.

Sats 1. *För en BSC med felsannolikhet $p < \frac{1}{2}$ är maximum-likelihoodavkodning ekvivalent med Hammingavkodning.*

Bevis. Låt C beteckna den kod som används och låt M_r vara det mottagna ordet av längd n. För godtycklig vektor $\boldsymbol{c}$ av längd n och för alla $0 \leq i \leq n$ har vi,

$$d(M_r, c) = i \Leftrightarrow P\ (M_r \text{ mottaget } |\ \boldsymbol{c} \text{ skickat}) = p^i(1-p)^{n-i}$$

Eftersom $p < \frac{1}{2}$ gäller för högerledet att

$$p^0(1-p)^n > p^1(1-p)^{n-1} > ... > p^n(1-p)^0$$

Per definition gäller att maximum-likelihoodavkodning avkodar/rättar M_r till det $\boldsymbol{c} \in C$ som ger det största P(M_r mottaget$|\boldsymbol{c}$ skickat) vilket samtidigt innebär att $d(M_r, c)$ är som minst. Eller begär omsändning om icke-fullständig avkodning används (se ovan) och $\boldsymbol{c}$ inte är unik. Alltså är maximum-likelihoodavkodning ekvivalent med Hammingavkodning. □

2.5 Feldetekterande kod

Vi har tidigare nämnt feldetekterande och felrättande koder. Det är dags att definiera dessa.

Definition 6 (Feldetekterande)**.** *Låt u vara ett positivt heltal. En kod C är $u-$feldetekterande om ett kodord som drar på sig minst ett men högst u fel inte längre är ett kodord. Ett kodord är exakt $u-$feldetekterande om det är $u-$feldetekterande men inte $(u+1)-$feldetekterande.*

Exempel 4. *Den binära koden $C = \{00000, 00111, 11111\}$ är $1-$feldetekterande ty om man ändrar en kodsymbol på godtycklig position får man ett nytt ord som inte är ett kodord. C är också exakt $1-$feldetekterande eftersom den inte är $2-$feldetekterande. Det inses genom att man via två ändringar kan bilda* 00111 *från* 11111*.*

Sats 2 (Feldetektering)**.** *En kod C är $u-$feldetekterande om och endast om $d(C) \geq u+1$*

Bevis. Antag $d(C) \geq u+1$. Om $\boldsymbol{c} \in C$ och $\boldsymbol{x}$ är sådana att $1 \leq d(\boldsymbol{c},\boldsymbol{x}) \leq u$ då gäller $\boldsymbol{x} \notin C$, eftersom $u \leq d(C) - 1 < d(C)$ Alltså är C $u-$feldetekterande.

Antag nu istället att $d(C) < u+1$ dvs $d(C) \leq u$. Då finns $\boldsymbol{c}_1,\boldsymbol{c}_2 \in C$ sådana att $1 \leq d(\boldsymbol{c}_1,\boldsymbol{c}_2) = d(C) \leq u$. Det är därför möjligt att börja med $\boldsymbol{c}_1$ varefter $d(C)$ fel inträffar så att ordet blir $\boldsymbol{c}_2$, som är ett kodord i C. Alltså är C inte $u-$feldetekterande. □

2.6 Felrättande kod

Definition 7 (Felrättande)**.** *Låt v vara ett positivt heltal. En kod C är $v-$felrättande om Hammingavkodning rättar v eller färre fel. En kod C är exakt $v-$felrättande om den är $v-$felrättande men inte $(v+1)-$felrättande.*

Exempel 5. *Betrakta den binära koden $C = \{000, 111\}$. Genom Hammingavkodning ser vi att*

1. *om* 000 *är skickat och ett fel uppstår så är det mottagna ordet* $100, 010$ *eller* 001*. Det avkodas/rättas till* 000*.*
2. *om* 111 *är skickat och ett fel uppstår så är det mottagna ordet* $110, 101$ *eller* 011*. Det avkodas/rättas till* 111*.*

Ett ensamt fel rättas utan problem. Koden klarar dock inte att rätta två fel. C är således exakt $1-$felrättande.

Sats 3 (Felrättning)**.** *En kod C är $v-$felrättande om och endast om $d(C) \geq 2v+1$.*

Lite mer precist gäller att en kod med minimumavstånd d är exakt $\lfloor (d-1)/2 \rfloor$−felrättande, se Sats 9 för symbolförklaring. Nu till bevis av Sats 3.

Bevis. ($\Leftarrow$) Antag $d(C) \geq 2v+1$. Låt $\boldsymbol{c}$ vara kodordet som skickats och $\boldsymbol{x}$ kodordet som mottagits. Om v eller färre fel inträffar så är $d(\boldsymbol{c},\boldsymbol{x}) \leq v$. För vilket annat som helst $\boldsymbol{c}' \in C$, $\boldsymbol{c} \neq \boldsymbol{c}'$ har vi att

$$d(\boldsymbol{c}', \boldsymbol{x}) \geq d(\boldsymbol{c}', \boldsymbol{c}) - d(\boldsymbol{c}, \boldsymbol{x})$$

enligt triangelolikheten (Tengstrand [7] sid 295). Det gäller att

$$d(\boldsymbol{c}',\boldsymbol{c}) - d(\boldsymbol{c},\boldsymbol{x}) \geq 2v+1-v = v+1 > d(\boldsymbol{c},\boldsymbol{x})$$

Så, $\boldsymbol{x}$ kommer avkodas (rätt) till $\boldsymbol{c}$ om Hammingavkodning används. Detta visar att C är v−felrättande.

($\Rightarrow$) Antag nu att C är v−felrättande. Om $d(C) < 2v+1$ så finns skilda kodord $\boldsymbol{c},\boldsymbol{c}' \in C$ med $d(\boldsymbol{c}',\boldsymbol{c}) = d(C) \leq 2v$. Vi ska visa att det (förutsatt att $\boldsymbol{c}$ är skickat och högst v fel inträffar) kan inträffa att Hammingavkodningen antingen avkodar fel eller ger flera möjliga alternativ. Detta motsäger ju antagandet att C är v−felrättande. Alltså visas $d(C) \geq 2v+1$.

Notera att, om $d(\boldsymbol{c}',\boldsymbol{c}) < v+1$ så kan $\boldsymbol{c}$ ändras till $\boldsymbol{c}'$ genom att införa högst v fel. Dessa fel skulle passera orättade och även oupptäckta eftersom $\boldsymbol{c}' \in C$. Detta motsäger dock antagandet att C är v−felrättande. Alltså är $d(\boldsymbol{c}',\boldsymbol{c}) \geq v+1$. Utan inskränkning kan vi anta att $\boldsymbol{c}$ och $\boldsymbol{c}'$ skiljer sig på exakt de $d = d(C)$ första positionerna där $v+1 \leq d \leq 2v$. Om ordet som mottas är

$$\boldsymbol{x} = \underbrace{x_1 \ldots x_v}_{\textit{lika som } c'} \; \underbrace{x_{v+1} \ldots x_d}_{\textit{lika som } c} \; \underbrace{x_{d+1} \ldots x_n}_{\textit{lika som } c' \textit{ och } c}$$

så har vi $d(\boldsymbol{c}',\boldsymbol{x}) = d - v \leq v = d(\boldsymbol{c},\boldsymbol{x})$. Det följer att antingen $d(\boldsymbol{c}',\boldsymbol{x}) < d(\boldsymbol{c},\boldsymbol{x})$ eller $d(\boldsymbol{c}',\boldsymbol{x}) = d(\boldsymbol{c},\boldsymbol{x})$. I det första fallet får vi felaktig avkodning till $\boldsymbol{c}'$, i det andra oavgjord avkodning. □

3 Lineära koder

Lineära koder är koder där koden är vektorer i ett vektorrum över en ändlig kropp (Tengstrand [7] sid 277) och lineära kombinationer av kodord också är kodord. Om den ändliga kroppen bara har två element och alltså svarar mot heltalen modulo 2, kallas koden binär. För binära lineära koder är koeffecienterna antingen 0 eller 1 så en lineär kombination här är detsamma som summan av ett urval av kodorden. Det kan också vara bra att minnas att summan av två binära tal är lika med differensen mellan dem.

Lineära koder har bland annat effektivare metoder för avkodning än icke-lineära koder. Syftet med koden är fortfarande att förlänga sitt meddelande med ett antal extra tecken för att kunna upptäcka och även rätta fel.

Intuitivt inser man att sändning av mer information med färre möjliga tecken kräver längre kod. I texten nedan har jag begränsat mig till binära lineära koder. Även i praktiken har binära koder många fördelar eftersom de kan hanteras genom att bara skilja mellan på/av eller svart/vitt etc.

Först definierar vi några algebraiska begrepp.

3.1 Lineär algebra

Definition 8 (Lineärt oberoende)**.** *Låt V vara ett vektorrum över $\mathbb{F}_q$. En mängd vektorer $\{e_1, \ldots, e_r\}$ i V är lineärt oberoende om*

$$a_1e_1 + ... + a_re_r = 0 \Rightarrow a_1 = ... = a_r = 0$$

Mängden är lineärt beroende om den inte är lineärt oberoende.

Definition 9 ($<S>$)**.** *Låt V vara ett vektorrum över $\mathbb{F}_q$ och låt $S = \{e_1, e_2, \ldots, e_k\}$ vara en icke-tom delmängd av V. Vi säger att S spänner upp $< S >$ där*

$$< S > = \{(a_1e_1 + \cdots + a_ke_k) : a_i \in \mathbb{F}_q\}$$

Om S är en tom mängd definierar vi $< S >$ som $\{0\}$.

Exempel 6. *Om $q = 2$ så spänner $S = \{0001, 0010, 0100\}$ upp en binär lineär kod*

$$< S > = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111\}$$

Definition 10 (Bas)**.** *Låt V vara ett vektorrum över $\mathbb{F}_q$. En icke-tom delmängd $S = \{e_1, \ldots, e_k\}$ av V kallas en bas för V om $V = < S >$ och S är lineärt oberoende.*

Ett vektorrum V över en ändlig kropp $\mathbb{F}_q$ kan ha flera baser men alla innehåller samma antal element. Antalet är dimensionen av V i $\mathbb{F}_q$. För att markera vilket vektorrum som avses kan man skriva $dim\ _{\mathbb{F}_q}(V)$ (se bland annat Tengstrand [7] sid 42)

Definition 11 (Skalärprodukt och ortogonalt komplement)**.** *Låt* $\boldsymbol{v} = (v_1, \ldots, v_n)$, $\boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{F}_q^n$

(i) Skalärprodukten av $\boldsymbol{v}$ *och* $\boldsymbol{w}$ *definieras som* $\boldsymbol{v} \cdot \boldsymbol{w} = v_1 \cdot w_1 + ... + v_n \cdot w_n \in \mathbb{F}_q$

(ii) De två vektorerna $\boldsymbol{v}$ *och* $\boldsymbol{w}$ *sägs vara ortogonala om* $\boldsymbol{v} \cdot \boldsymbol{w} = 0$

(iii) Låt S *vara en icke-tom delmängd av* $\mathbb{F}_q^n$*, där* $\mathbb{F}_q^n$ *är mängden av alla vektorer av längd* n *i* $\mathbb{F}_q$*. Det ortogonala komplementet* $S^\perp$ *till* S *definieras som*

$$S^\perp = \{\boldsymbol{v} \in \mathbb{F}_q^n : \boldsymbol{v} \cdot \boldsymbol{s} = 0 \text{ för alla } \boldsymbol{s} \in S\}$$

Om S *är den tomma mängd definierar vi* $S^\perp = \mathbb{F}_q^n$

3.2 Lineär kod

Lineära koder av längd n över den ändliga kroppen $\mathbb{F}_q$ är alltså koder där summan av två kodord också är ett kodord. Varje lineär kod måste på grund av detta innehålla nollvektorn. Vi gör en definition av lineär kod.

Definition 12 (Lineär kod)**.** *Den lineära koden* C *av längd* n *över* $\mathbb{F}_q$ *är ett delvektorrum till* $\mathbb{F}_q^n$*.*

En lineär kod C av längd n och dimension k över $\mathbb{F}_q$ skrivs ofta med hakparenteser som $[n, k]-$kod eller $[n, k, d]-$kod om d (minimumavståndet se Definition 3) är känt. Ett ekvivalent skrivsätt (från Kapitel 2) med vanliga parenteser, och M istället för k, är $(n, M) = (n, q^k)-$lineär kod. Dimensionen av C fås på samma sätt som är känt för andra vektorrum (Tengstrand [7] sid 285). Alltså om vår bas är $(e_1, ..., e_k)$ kan alla element skrivas unikt som lineärkombinationer $(a_1e_1 + ... + a_ke_k)$ där a_i tillhör kroppen $\mathbb{F}_q$ med q element. Så C innehåller precis q^k element. Dimensionen för vår kod får vi genom $k = log_q q^k = log_q M$

Exempel 7. *Binär lineär kod*

$C_1 = \{000000, 100000, 010000, 110000\}$

$C_2 = \{000000, 111000, 000111, 111111\}$

För den binära koden C_1 är $q = 2$ och vi har $4 = 2^2$ kodord av längd 6. Dimensionen blir $k = 2$. Hammingvikten av kodorden som inte är noll är $1, 1$ respektive 2. Minimumavståndet är alltså 1. Felrättning är inte möjligt enligt vårt tidigare resonemang (Sats 3).

För C_2 har vi också dimension $k = 2$. Hammingvikterna är $3, 3$ respektive 6 för icke-noll kodorden. Den minsta Hammingvikten är 3 och minimumavståndet är 3. Koden är alltså $1-$felrättande. Förhållandet mellan Hammingvikt och minimumavstånd i båda dessa exempel är ingen slump, se vidare Sats 4.

Den *dubbla koden* kommer vi att använda i flera sammanhang. Det är lämpligt att definiera den här.

Definition 13. *(Dubbla koden)*
Låt C vara en linear kod i $\mathbb{F}_q^n$. Den dubbla koden till C är $C^\perp$, det ortogonala komplementet till delmängden C i $\mathbb{F}_q^n$

Följande sats kan användas för att finna minimumavståndet $d(C)$ för en linear kod C utan att behöva jämföra Hammingavstånden för alla möjliga kombinationer av kodord två och två.

Sats 4. *För en linear kod C över $\mathbb{F}_q$ gäller att $d(C) = wt(C)$, där $wt(C)$ är den minsta Hammingvikten av alla icke-noll kodord i C.*

Bevis. För godtyckliga ord $\boldsymbol{x}$ och $\boldsymbol{y}$ gäller att $d(\boldsymbol{x},\boldsymbol{y}) = wt(\boldsymbol{x}-\boldsymbol{y})$. Från definitionen av $d(C)$ (nummer 3) vet vi att det finns $\boldsymbol{x}'$, $\boldsymbol{y}' \in C$ sådana att $d(\boldsymbol{x}',\boldsymbol{y}') = d(C)$, så

$$d(C) = d(\boldsymbol{x}',\boldsymbol{y}') = wt(\boldsymbol{x}'-\boldsymbol{y}') \geq wt(C) \text{ ty } \boldsymbol{x}'-\boldsymbol{y}' \in C$$

Omvänt, det finns ett $\boldsymbol{z} \in C - \{\boldsymbol{0}\}$ sådant att $wt(C) = wt(\boldsymbol{z})$, så

$$wt(C) = wt(\boldsymbol{z}) = d(\boldsymbol{z},\boldsymbol{0}) \geq d(C)$$

□

Några fördelar med lineära koder:

(i) Lineära koder är vektorrum och kan beskrivas enbart med dess bas (se Kapitel 3.3).

(ii) Minimumavståndet i en linear kod är detsamma som den lägsta Hammingvikten av de kodord som är skilda från noll.

(iii) Kodning och avkodning av lineära koder är snabbare och enklare än för godtyckliga icke-lineära koder, vilket jag återkommer till.

3.3 Bas

Som nyss nämnts kan lineära koder, eftersom de är vektorrum, beskrivas av baser. Dessa kan med fördel skrivas i matrisform. Det finns några olika sätt att få fram en bas. Först lite matrisrepition.

När man använder sig av *successiv elimination*, även kallat *Gausselimination* (Tengstrand [7] sid 7), minns vi att följande *elementära radoperationer* är tillåtna

(i) Byta plats på två rader sinsemellan.

(ii) Multiplicera en rad med en icke-noll skalär.

(iii) Ersätta en rad med summan mellan den raden och en annan rad som man först multiplicerat med en skalär.

Antag att man har framför sig en delmängd S (här innehållande kodord) av $\mathbb{F}_q^n$ och man vill hitta basen för den lineära kod som spänns upp av S.

Börja med att bilda en matris vars rader är kodorden i S. Använd elementära radoperationer så långt det är möjligt, det vill säga tills matrisen är på *trappstegsform* (se nedan och även kallat *echelonform*) (se t.ex. Tengstrand [7] sid 7). Ta bort rader som bara innehåller nollor.

Icke-noll raderna av matrisen är nu lineärt oberoende och bildar en bas för C (se Definition 10).

Exempel 8. *Låt $q = 3$. Finn en bas för $C = < S >$, där $S = \{12101, 20110, 01122, 11010\}$. Med hjälp av successiv elimination kan vi skriva matrisen så här*

$$\begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 0 & 2 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 2 & 2 & 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Den sista matrisen (längst till höger) är på trappstegsform. Genom att ta bort den sista raden i den matrisen bildar övriga rader en bas för C.

Definition 14 (Genereringsmatris, G)**.** *En genereringsmatris för en lineär kod C är en matris G vars rader bildar en bas för C.*

Om C är en $[n, k]-$lineär kod så är genereringsmatrisen en $k \times n-$matris (alltså k rader och n kolonner). Antalet kolonner bestäms direkt av att varje kodord har n kodsymboler (eller element). Att det blir precis k rader beror på att dimensionen är k. Detta i sin tur innebär ju att $k-1$ rader (vektorer) inte räcker för att bilda en bas samtidigt som godtyckliga $k+1$ vektorer är lineärt beroende.

Definition 15 (Kontrollmatris/paritetscheckmatris, H)**.** *En paritetscheckmatris H för en lineär kod C är en genereringsmatris för den dubbla koden $C^{\perp}$.*

Om C är en $[n,k]-$lineär kod så är paritetscheckmatrisen en $(n-k)\times n-$matris.

Raderna i paritetscheckmatrisen genererar det ortogonala komplementet till en kod C. Det innebär att u är ett kodord i C om och endast om $uH^T = 0$. Matrisen är användbar till att kontrollera just detta (om m är ett kodord i C).

Definition 16 (Standardform för G och H)**.**

(i) En genereringsmatris med utseendet $(I_k|X)$, där I_k är identitetsmatrisen av storlek $k\times k$ (X är resterande del av matrisen G och symbolen $|$ är bara en tänkt skiljelinje) sägs vara på standardform.

(ii) En paritetscheckmatris på formen $(Y|I_{n-k})$ sägs vara på standardform (Y är resterande del av H).

Raderna i generingsmatrisen är lineärt oberoende. Detsamma gäller för H. För att visa att G är en genereringsmatris för en kod C räcker det att visa att alla rader i G är kodord i C och att raderna är lineärt oberoende.

Sats 5. *Låt C vara en $[n,k]-$lineär kod över $\mathbb{F}_q$ med genereringsmatris G. Då gäller att $\boldsymbol{v}\in\mathbb{F}_q^n$ tillhör den dubbla koden $C^{\perp}$ om och endast om $\boldsymbol{v}$ är ortogonal mot varje rad i G, det vill säga $\boldsymbol{v}\in C^{\perp}\Leftrightarrow \boldsymbol{v}G^T=\boldsymbol{0}$.*

Speciellt gäller, givet en $(n-k)\times n-$matris H, att H är en paritetscheckmatris för C om och endast om raderna i H är lineärt oberoende och $HG^T=\boldsymbol{0}$ (noll-matrisen).

Bevis. Låt $\boldsymbol{r}_i$ beteckna den i:te raden i G. Vi har att $\boldsymbol{r}_i\in C$ för alla $1\le i\le k$ och att varje $\boldsymbol{c}\in C$ kan skrivas som

$$\boldsymbol{c}=\lambda_1\boldsymbol{r}_1+\ldots+\lambda_k\boldsymbol{r}_k \qquad \text{där } \lambda_1,\ldots,\lambda_k\in\mathbb{F}_q$$

Om $\boldsymbol{v}\in C^{\perp}$, då är $\boldsymbol{v}\cdot\boldsymbol{c}=0$ för alla $\boldsymbol{c}\in C$. Speciellt är $\boldsymbol{v}$ ortogonal mot $\boldsymbol{r}_i$, för alla $1\le i\le k$, dvs $\boldsymbol{v}G^T=\boldsymbol{0}$.

Omvänt, om $\boldsymbol{v}\cdot\boldsymbol{r}_i=0$ för alla $1\le i\le k$ då är, för alla $\boldsymbol{c}=\lambda_1\boldsymbol{v}_1+\ldots+\lambda_k\boldsymbol{v}_k\in C$,

$$\boldsymbol{v}\cdot\boldsymbol{c}=\lambda_1(\boldsymbol{v}\cdot\boldsymbol{v}_1)+\cdots+\lambda_k(\boldsymbol{v}\cdot\boldsymbol{r}_k)=0$$

I det sista påståendet, om H är en paritetscheckmatris för C så är raderna i H lineärt oberoende per definition. Eftersom raderna i H är kodord i $C^{\perp}$ följer det från det första påståendet att $HG^T = \mathbf{0}$.

Omvänt om $HG^T = \mathbf{0}$ så visar det första påståendet att raderna i H är innehållna i $C^{\perp}$. Eftersom raderna i H är lineärt oberoende så har de möjliga kombinationerna dimensionen $(n-k)$, det vill säga samma som $C^{\perp}$. Med andra ord är H en paritetscheckmatris för C. □

Sats 6. *Om $G = (I_k|X)$ är en genereringsmatris på standardform för en kod C så är $H = (-X^T|I_{n-k})$ en paritetscheckmatris för C.*

Bevis. Vi konstaterar direkt att ekvationen $HG^T = 0$ är uppfylld. Genom att betrakta de sista $n-k$ koordinaterna i varje rad inser man att raderna i H är lineärt oberoende. Resultatet följer nu från Sats 5. Notera att $-X^T = X^T$ om X bara innehåller binära tal. □

Lite mer om hur man bestämmer en paritetscheckmatris finns i Exempel 16 längre fram.

Sats 7. *Låt C vara en lineär kod och låt H vara en paritetscheckmatris för C. Då gäller att*

(i) C har minimumavstånd $\geq d$ om och endast om $d-1$ kolonner i H är lineärt oberoende och

(ii) C har minimumavstånd $\leq d$ om och endast om H har d kolonner som är lineärt beroende.

Bevis. Låt $\boldsymbol{v} = (v_1, ..., v_n) \in C$ vara ett ord med Hammingvikt $e > 0$. Antag att icke-noll koordinaterna är på positionerna $i_1, ..., i_e$, så att $v_j = 0$ om $j \notin \{i_1, ..., i_e\}$. Låt $\boldsymbol{c}_i$ (för $1 \leq i \leq n$) beteckna den i:te kolonnen av H. Då innehåller C ett icke-noll ord $\boldsymbol{v} = (v_1, ..., v_n)$ med Hammingvikt e (vars icke-noll koordinater är $v_{i_1}, ..., v_{i_e}$) om och endast om

$$\mathbf{0} = \boldsymbol{v}H^T = v_{i_1}\boldsymbol{c}_{i_1}^T + \mathbf{...} + v_{i_e}\boldsymbol{c}_{i_e}^T$$

Detta är sant om och endast om det finns e kolonner i H (nämligen $c_{i_1}, ..., c_{i_e}$) som är lineärt beroende. Att säga att minimumavståndet i C är större eller lika med d är ekvivalent med att säga att C inte innehåller några icke-noll ord av Hammingvikt $\leq d-1$ som i sin tur är ekvivalent med att $\leq d-1$ godtyckliga kolonner i H är lineärt oberoende. Detta visar (i).

På liknande sätt, att säga att minimumavståndet i C är mindre eller lika med d är ekvivalent med att säga att C innehåller minst ett icke-noll ord med Hammingvikt $\leq d$ som i sin tur är ekvivalent med att säga att H har $\leq d$ stycken kolonner som är lineärt beroende. Vilket visar (ii). □

Exempel 9. *Låt C vara den binära lineära koden med paritetscheckmatris*

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

Som synes finns inga noll-kolonner och inte heller några kombinationer av två kolonner som man kolonnvis kan summera till $\boldsymbol{0}$, så två godtyckliga kolonner från H är lineärt oberoende. Däremot summeras kolonnerna 1,3 och 4 till 0 *och är därför lineärt beroende. Vilket ger minimumavstånd $d(C) = 3$ enligt Sats 7.*

Några lineära koder saknar genereringsmatris i sin ursprungliga kodform. Men genom permutation av tecken eller multiplikation med en skalär på utvalda positioner kan man alltid finna en ny ekvivalent kod som har en genereringsmatris.

3.4 Kodning av lineär kod

Låt C vara en $[n, k, d]-$lineär kod över den ändliga kropppen $\mathbb{F}_q$. Varje kodord i C kan representera en informationsdel så C kan representera q^k delar av information. I Kapitel 3.3 tittade vi på hur man hittar en bas för en lineär kod. När man har en sådan bas $(\boldsymbol{e}_1, ..., \boldsymbol{e}_k)$ kan varje kodord $\boldsymbol{v}$ skrivas unikt som en lineär kombination

$$\boldsymbol{v} = u_1\boldsymbol{e}_1 + ... + u_k\boldsymbol{e}_k \text{ där } u_1, ..., u_k \in \mathbb{F}_q$$

Ekvivalent kan vi låta G vara genereringsmatrisen för C vars i:te rad är vektorn $\boldsymbol{e}_i$ i den valda basen. Givet en vektor $\boldsymbol{u} = (u_1, ..., u_k) \in \mathbb{F}_q^k$ är det klart att

$$\boldsymbol{v} = \boldsymbol{u}G = u_1\boldsymbol{e}_1 + ... + u_k\boldsymbol{e}_k \text{ är ett kodord i } C$$

Omvänt, alla $\boldsymbol{v} \in C$ kan skrivas unikt som $\boldsymbol{v} = \boldsymbol{u}G$, där $\boldsymbol{u} = (u_1, ..., u_k) \in \mathbb{F}_q^k$. Alltså alla ord $\boldsymbol{u} \in \mathbb{F}_q^k$ kan kodas som $\boldsymbol{v} = \boldsymbol{u}G$. Detta kallas kodning.

Exempel 10. *Låt C vara en binär $(5, 3)-$lineär kod med genereringsmatris*

$$G_{(5,3)} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

då kodas meddelandet $\boldsymbol{u} = 101$ så här

$$\boldsymbol{v} = \boldsymbol{u}G = (101) \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} =$$

$$= 1+0+0,\ 0+0+0,\ 1+0+1,\ 1+0+0,\ 0+0+1 = 1,0,0,1,1$$

vi ser att det kodade ordet består av två tecken fler än det ursprungliga meddelandet. Det är ingen tillfällighet att man också kan få fram kodordet genom att summera rad 1 *och rad* 3 *i genereringsmatrisen.*

3.5 Avkodning av lineär kod

Vi fortsätter med två olika metoder för avkodning.

Definition 17 (Bimängd/Koset)**.** *Låt C vara en lineär kod av längd n över $\mathbb{F}_q^n$ och låt $\boldsymbol{u} \in \mathbb{F}_q^n$ vara godtycklig vektor av längd n. Vi definierar bimängden till C, som bestäms av $\boldsymbol{u}$, som mängden*

$$C+\boldsymbol{u} = \{(\boldsymbol{v}+\boldsymbol{u}) : \boldsymbol{v} \in C\} \quad (= \boldsymbol{u}+C)$$

Exempel 11. *Låt $q = 2$ och $C = \{000, 101, 010, 111\}$. Då är bimängderna*

$C + 000 = \{000, 101, 010, 111\}$

$C + 001 = \{001, 100, 011, 110\}$

$C + 010 = \{010, 111, 000, 101\}$

$C + 011 = \{011, 110, 001, 100\}$

$C + 100 = \{100, 001, 110, 011\}$

$C + 101 = \{101, 000, 111, 010\}$

$C + 110 = \{110, 011, 100, 001\}$

$C + 111 = \{111, 010, 101, 000\}$

notera att

$C + 000 = C + 010 = C + 101 = C + 111 = C$

$C + 001 = C + 011 = C + 100 = C + 110 = \mathbb{F}_2^3$ **-** C

Definition 18. *(Bimängdsledare) Kodordet med minst Hammingvikt i en bimängd kallas bimängdsledare.*

3.5.1 Hammingavkodning

Låt C vara en lineär kod. Antag att kodordet $\boldsymbol{v} \in C$ skickats och ordet $\boldsymbol{w}$ mottagits. Då dessa är olika får vi ett fel som kan skrivas $\boldsymbol{e} = \boldsymbol{w} - \boldsymbol{v} \in \boldsymbol{w}+C$. Vi ser att $\boldsymbol{w} - \boldsymbol{e} = \boldsymbol{v} \in C$ och att $\boldsymbol{e}$ och $\boldsymbol{w}$ tillhör samma bimängd.

Eftersom fel $\boldsymbol{e}$ med liten Hammingvikt är mer sannolika än fel med stor Hammingvikt fungerar avkodningen på följande sätt. Efter att ha mottagit ordet $\boldsymbol{w}$ väljer man det ord $\boldsymbol{e}$ som har minst Hammingvikt i bimängden $\boldsymbol{w}+C$ och drar slutsatsen att $\boldsymbol{v} = \boldsymbol{w} - \boldsymbol{e}$ var det skickade kodordet.

Exempel 12. *Låt $q = 2$ och $C = \{0000, 1011, 0101, 1110\}$. Avkoda det mottagna ordet $w = 1101$. Man ser visserligen direkt att 0101 är det enda ord som ligger ett fel ifrån vårt mottagna ord och därmed är mest sannolikt men vi låtsas inte om det nu utan gör den fullständiga analysen*

Först listar vi bimängderna till C

$C + 0000 = \{0000, 1011, 0101, 1110\}$
$C + 0001 = \{0001, 1010, 0100, 1111\}$ $(= C + 0100)$
$C + 0010 = \{0010, 1001, 0111, 1100\}$
$C + 1000 = \{1000, 0011, 1101, 0110\}$

$w = 1101$ *hittar vi i den fjärde raden. Ordet med minst Hammingvikt i denna rad är $u = 1000$, alltså vår bimängdsledare. Vi har att $w-u = 1101-1000 = 1101 + 1000 = 0101$ som med störst sannolikhet var det ord som skickades. Notera att detta ord står överst i samma kolonn, på raden för $C + 0000$.*

3.5.2 Syndromavkodning

Avkodningen i Kapitel 3.5.1 fungerar bra när n är litet men för stora n tjänar man på att använda *syndrom* för att avgöra från vilken bimängd det mottagna ordet kommer. Med hjälp av en paritetscheckmatris fås ett unikt syndrom för varje fel som koden kan rätta. Man vill konstruera syndromen så att syndromet av ett kodord är noll eftersom det representerar noll fel.

Definition 19. *(Syndrom) Låt C vara en $[n, k, d]-$lineär kod på $\mathbb{F}_q$ och låt H vara en paritetscheckmatris för C. För något $\boldsymbol{w} \in F_q^n$ är syndromet för $\boldsymbol{w}$ ordet $S(\boldsymbol{w}) = \boldsymbol{w}H^T \in F_q^{n-k}$*

Notera att syndromet beror på valet av paritetscheckmatrisen H. För att hålla ordning på våra syndrom använder vi en *syndromtabell*.

Vi konstruerar en syndromtabell för C för den binära lineära koden $C = \{0000, 1011, 0101, 1110\}$. En genereringsmatris är

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

Gör så här.

1. Lista alla bimängder i koden. Välj det kodord som har minst Hammingvikt i varje bimängd. Låt dessa vara våra bimängdsledare, alltså $\boldsymbol{u}$.

2. Beräkna syndromet för varje bimängdsledare $\boldsymbol{u}$, genom $S(\boldsymbol{u}) = \boldsymbol{u}H^T$

Vi listade bimängderna i Exempel 12. Orden $0000, 0001, 0010$ och 1000 väljs igen som bimängdsledare. En paritetscheckmatris för C som uppfyller $HG^T = \mathbf{0}$ (se Sats 5) är

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

Syndromtabellen får följande utseende

Bimängdsledare, $\boldsymbol{u}$	Syndrom, $S(\boldsymbol{u})$
0000	00
0001 (=0100)	01
0010	10
1000	11

Avkodningsproceduren går nu till enligt följande

(i) Beräkna $S(\boldsymbol{w})$, syndromet för det mottagna ordet $\boldsymbol{w}$. I Hammingavkodningen ovan undersöktes vilken bimängd $\boldsymbol{w}$ tillhörde genom att lista alla kombinationer. Här räcker det att beräkna syndromet för $\boldsymbol{w}$ eftersom om två ord har samma syndrom så tillhör de samma bimängd (Xing [5] sid 62)

(ii) Hitta bimängdsledaren $\boldsymbol{u}$ som hör till uträknat syndrom, det vill säga läs av i tabellen ovan.

(iii) Det skickade ordet $\boldsymbol{v}$ får vi fram genom att ta det mottagna ordet $\boldsymbol{w}$ minus vår bimängdsledare $\boldsymbol{u}$.

Exempel 13. *Låt $q = 2$ och $C = \{0000, 1011, 0101, 1110\}$. Använd syndromtabellen ovan. Antag att vi vill avkoda $\boldsymbol{w} = 1101$.*

Syndromet är $S(\boldsymbol{w}) = \boldsymbol{w}H^T = 11$. Från syndromtabellen ser vi att bimängdsledare är 1000. Alltså har vi att $\boldsymbol{v} = 1101 - 1000 = 0101$ med störst sannolikhet var det skickade kodordet. Vilket lyckligtvis är detsamma som vi fick med Hammingavkodning i Exempel 12.

4 Begränsningar

Givet en kod med parametrar n, M och d, där n är fix, så är M ett mått på kodens effektivitet och d ett mått på felrättningsförmågan. Det bästa vore om båda var så stora som möjligt men en kompromiss mellan dem är nödvändig.

Definition 20 (Relativa minimumavståndet)**.** *För en $q-$är kod C med parametrar (n, M, d) är det relativa minimumavståndet i C, $\delta(C) = (d-1)/n$.*

Det är också möjligt att definiera det relativa minimumavståndet som d/n. Men eftersom vissa formler blir enklare väljs ofta den första definitionen.

Vi ska titta på ett exempel. Från Definition 1 minns vi att informationsgraden skrivs som $R(C) = \frac{log_q M}{n}$

Exempel 14. *Betrakta repetitionskoden $C = \{\underbrace{00\ldots0}_{n}, \underbrace{11\ldots1}_{n}\}$ över $\mathbb{F}_q$.*

Det är en binär $(n, M, d) = (n, 2, n)-$lineär kod eller ekvivalent en binär $[n, k, d] = [n, 1, n]-$kod, eftersom

$$M = 2$$

$$k = dim(C) = log_q(M) = log_2(2) = 1$$

Vi får att

$$R(C) = \frac{log_2(2)}{n} = \frac{1}{n} \to 0 \text{ då } n \to \infty$$

$$\delta(C) = \frac{n-1}{n} \to 1 \text{ då } n \to \infty$$

Då denna kod har största möjliga relativa minimumavstånd har den utmärkta felrättande egenskaper. Men detta uppnås på bekostnad av effektivitet eftersom informationsgraden $R(C)$ är den lägsta möjliga.

Definition 21 (Största M)**.** *För ett givet kodalfabet A av storlek q $(q > 1)$ och givna värden på n och d låt $A_q(n,d)$ beteckna den största möjliga storleken M för vilken det existerar en $(n, M, d)-$kod på A.*

$$A_q(n, d) = max\{M : \textit{det finns en } (n, M, d)-\textit{kod på } A\}$$

Alla $(n, M, d)-$koder C som har maximal storlek $M = A_q$(n,d) kallas *optimala koder*.

Definition 22 (Största q^k)**.** *För ett givet primtal q och givna värden på n och d låt $B_q(n, d)$ beteckna den största möjliga storleken q^k för vilken det finns en $[n, k, d]-$kod över $\mathbb{F}_q$. Alltså*

$$B_q(n, d) = max\{q^k : \textit{det finns en } [n, k, d]-\textit{kod över } \mathbb{F}_q\}$$

Generellt är det svårt att bestämma exakta värden på $A_q(n,d)$ och $B_q(n,d)$. Problemet med att bestämma $A_q(n,d)$ kallas ibland för 'the main coding theory problem'.

Vi ska titta mer på dessa begränsningar och särskilt för binära koder, då $q=2$. För detta behöver vi den utökade koden till C, som skrivs $\overline{C}$. Den fås genom att lägga till en koordinat till varje kodord.

Definition 23. *För någon kod C i $\mathbb{F}_q$ är den utökade koden till C*

$$\overline{C} = \{(c_1, ..., c_n, -\sum_{i=1}^{n}(c_i)) : (c_1, ...c_n) \in C\}$$

Då $q=2$ kallas den extra koordinaten som läggs till kodordet för paritetscheck-koordinaten och det gäller i detta fall att $-\sum_{i=1}^{n}(c_i) = \sum_{i=1}^{n}(c_i)$

Exempel 15. *För den binära lineära koden $C = \{000, 111, 011, 100\}$ har vi parametrarna $[3,2,1]$. Den utökade koden $\overline{C}$ fås genom att*

kodord 1 får extra koordinat $\sum_{i=1}^{3}(c_i) = 0+0+0 = 0$

kodord 2 får extra koordinat $\sum_{i=1}^{3}(c_i) = 1+1+1 = 1$

kodord 3 får extra koordinat $\sum_{i=1}^{3}(c_i) = 0+1+1 = 0$

kodord 4 får extra koordinat $\sum_{i=1}^{3}(c_i) = 1+0+0 = 1$

vilket ger oss $\overline{C} = \{0000, 1111, 0110, 1001\}$ som är en binär lineär $[4,2,2]-$kod.

En viktig begränsning nedåt för $A_q(n,d)$ är *sfärtäckningsbegränsningen*. Man kan visa (till exempel Xing [5] sid 80) att för ett heltal $q>1$ och heltal n,d sådana att $1 \leq d \leq n$ gäller att

$$\frac{q^n}{\sum_{i=0}^{d-1}\binom{n}{i}(q-1)^i} \leq A_q(\mathrm{n,d})$$

Det finns flera begränsningar nedåt för både $A_q(n,d)$ och $B_q(n,d)$. Men vi går vidare till de övre begränsningarna eftersom de är mer intressanta. En av de vanligare är *sfärpackningsbegränsningen*, även kallad *Hammingbegränsningen*. Vi behöver två definitioner och en hjälpsats innan vi kan nå fram till Sats 9.

Definition 24. *Låt A vara ett kodalfabet av storlek q, $q>1$. För någon vektor $\boldsymbol{u} \in A^n$, dvs i A och med längd n, och vilket som helst heltal $r \geq 0$ är sfären, med mittpunkt $\boldsymbol{u}$ och radie r, mängden*

$$S_A(\boldsymbol{u}, r)) = \{\boldsymbol{v} \in A^n : d(\boldsymbol{u},\boldsymbol{v}) \leq r\}$$

Vi fortsätter med att räkna ut antalet vektorer i A^n.

Definition 25. *För ett givet heltal $q > 1$, ett positivt heltal n och ett heltal $r \geq 0$ definieras $V_q^n(r)$ till*

$$V_q^n(r) = \begin{cases} \binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + ... + \binom{n}{r}(q-1)^r & om\ 0 \leq r \leq n \\ q^n & om\ n \leq r \end{cases}$$

Sats 8. *För alla heltal $r \geq 0$ gäller att en sfär i A^n med radien r innehåller exakt $V_q^n(r)$ vektorer, där A är ett alfabet med $q > 1$ element.*

Bevis. Välj en viss vektor $\boldsymbol{u} \in A^n$. Bestäm antalet vektorer $\boldsymbol{v} \in A^n$ sådana att $d(\boldsymbol{u}, \boldsymbol{v}) = m$; alltså antalet vektorer i A^n med Hammingavstånd exakt m från $\boldsymbol{u}$. Antalet sätt man kan välja de m koordinaterna där $\boldsymbol{v}$ skiljer sig från $\boldsymbol{u}$ ges av $\binom{n}{m}$. För varje koordinat har vi $q-1$ val för den koordinaten i $\boldsymbol{v}$. Därför ges det totala antalet vektorer med avstånd m från $\boldsymbol{u}$ av $\binom{n}{m}(q-1)^m$. För $0 \leq r \leq n$ är beviset nu klart.

Då $n \leq r$ notera att $S_A(\boldsymbol{u}, r) = A^n$. Alltså innehåller den $V_q^n(r) = q^n$ vektorer. □

Sats 9 (Hammingbegränsningen)**.** *För ett heltal $q > 1$ och heltal n, d sådana att $1 \leq d \leq n$ har vi*

$$A_q(n, d) \leq \frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}(q-1)^i}$$

där $\lfloor (d-1)/2 \rfloor$ är ('floor function') det största heltal som är $\leq (d-1)/2$

Bevis. Låt $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_M\}$ vara en optimal $(n, M, d)-$kod på alfabetet A och låt antal kodsymboler i A vara q stycken så $M = A_q(n, d)$. Låt $e = \lfloor (d-1)/2 \rfloor$. Packningssfärerna $S_A(\boldsymbol{c}_i, e)$ är disjunkta och varje ord i A^n finns i högst en sfär.

Vi har alltså

$$\bigcup_{i=1}^{M} S_A(\boldsymbol{c}_i, e) \subseteq A^n$$

där unionen i vänsterledet är en disjunkt union. Eftersom $|A^n| = q^n$ och $|S_A(\boldsymbol{c}_i, e)| = V_q^n(e)$ för vilket i som helst har vi att,

$$M \cdot V_q^n(e) \leq q^n \implies A_q(n, d) = M \leq \frac{q^n}{V_q^n(e)} = \frac{q^n}{V_q^n(\lfloor (d-1)/2 \rfloor)}$$

□

Definition 26 (Perfekt kod)**.** *En $q-$är kod som i antal kodord når upp till Hammingbegränsningen, dvs en som har*

$$\frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}(q-1)^i}$$

kodord, kallas för perfekt kod.

4.1 Hammingkod

Exempel på perfekta koder är *Hammingkoder*. De har intressanta tillämpningar och är lätta att hantera.

Definition 27 (Hammingkod)**.** *Låt $r \geq 2$. En binär lineär kod av längd $n = 2^r - 1$ med paritetscheckmatris H, vars kolonner består av alla icke-noll vektorer i $\mathbb{F}_2^r$, kallas binär Hammingkod av längd $2^r - 1$ och skrivs $Ham(r, 2)$.*

Exempel 16. *$Ham(3, 2)$ är (med våra vanliga beteckningar skulle den kallats $[7, 4, 3]-$lineär kod) en Hammingkod av längd 7. Den kodar 4 tecken data till 7 tecken kod genom att lägga till 3 tecken för feldetektering/felrättning. Om man vill jämföra överföringshastigheter mellan koder, så räknas denna kods hastighet vanligen ut som $\frac{4}{7}$. Den har genereringsmatris (notera att vänstra delen av G är en enhetsmatris ty G är på standardform, se Definition 16)*

$$G_{Ham(3,2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\textit{och paritetscheckmatris } H_{Ham(3,2)} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$H_{Ham(3,2)}$ kan också fås genom att låta vänstra delen av $G_{Ham(3,2)}$, dvs I_4, bilda högra delen av $H_{Ham(3,2)}$, nu som $I_{7-4} = I_3$, samt transponera högra delen av $G_{Ham(3,2)}$ till vänstra delen av $H_{Ham(3,2)}$.

Sats 10. *För Hammingkoder gäller*

(*i*) *Alla binära Hammingkoder av given längd är ekvivalenta.*

(*ii*) *Dimensionen av $Ham(r, 2)$ är $k = 2^r - 1 - r$.*

(*iii*) *Minimumavståndet, d, i $Ham(r, 2)$ är 3, alltså är den exakt $1-$felrättande.*

(*iv*) *Binära Hammingkoder är perfekta koder.*

Bevis. (*i*) För en given längd, kan varje paritetscheckmatris fås från en annan paritetscheckmatris genom permutationer av kolonner. Direkt från definitionen följer att motsvarande binära Hammingkoder är ekvivalenta.

(*ii*) En paritetscheckmatris H för $Ham(r, 2)$ med längd $2^r - 1$ är en $r \times (2^r - 1)-$matris så dimensionen av koden är $2^r - 1 - r$ (se Definition 15)

(*iii*) Eftersom det inte finns några kolonner som är lika i H, är två godtyckliga kolonner i H lineärt oberoende. Å andra sidan innehåller H kolonnerna

$$\begin{matrix}1\\0\\.\\0\end{matrix}, \quad \begin{matrix}0\\1\\.\\0\end{matrix} \text{ och } \begin{matrix}1\\1\\.\\0\end{matrix}$$

som bildar en lineärt beroende mängd. Minimumavståndet för $Ham(r,2)$ är därför 3 enligt Sats 7. Det följer att den är $1-$felrättande enligt Sats 3.

(iv) En binär kod som uppnår Hammingbegränsningen är en perfekt kod enligt Definition 26. För $Ham(r,2)$ kan vi skriva detta som

$$|C| = \frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}(q-1)^i} = \frac{2^n}{1+n} \text{ då } q = 2 \text{ och } d = 3$$

Vi har att $n = 2^r - 1$ enligt Definition 27. Den är alltså perfekt om och endast om antalet kodord är

$$|C| = \frac{2^{2^r-1}}{1+2^r-1} = 2^{2^r-1-r}$$

vilket är detsamma som 2^k om man använder sig av resultatet i (ii). □

4.2 Avkodning av Hammingkod

Eftersom $Ham(r,2)$ är en perfekt $1-$felrättande kod finns 2^r bimängder och bimängdsledare är precis de 2^r $(= n+1)$ vektorerna av längd n och Hammingvikt ≤ 1 (alltså $0\ldots00$, $0\ldots01$, $0\ldots10$ osv upp till $n+1$ st). Låt $\boldsymbol{u}_j$ beteckna vektorn med 1 på den j:te koordinaten och 0 i övrigt. Då är syndromet av $\boldsymbol{u}_j$ helt enkelt $\boldsymbol{u}_j H^T$, dvs transponatet av den j:te kolonnen i H. Om kolonnerna i H är arrangerade i ökande ordning (binära tal) så görs avkodningen så här:

Steg 1: Då $\boldsymbol{w}$ är mottagen, beräkna syndromet $\mathrm{S}(\boldsymbol{w}) = \boldsymbol{w}H^T$
Steg 2: Om $\mathrm{S}(\boldsymbol{w}) = 0$ antag att $\boldsymbol{w}$ var det skickade kodordet.
Steg 3: Om $\mathrm{S}(\boldsymbol{w}) \neq 0$ låter vi $\mathrm{S}(\boldsymbol{w})$ representera j, för något $1 \leq j \leq 2^r - 1$. Antaget ett singelfel så ges felet av ordet $\boldsymbol{u}_j$, så det skickade ordet ges av $\boldsymbol{w}$-$\boldsymbol{u}_j = \boldsymbol{w}+\boldsymbol{u}_j$

Exempel 17. *Gör först en syndromtabell (enligt metoden sid 20) för $Ham(3,2)$ (se Exempel 16). För att avkoda $\boldsymbol{w} = 1001001$ tar vi reda på dess syndrom. Det blir $\boldsymbol{w}H^T = 010$, även H kommer från Exemplet med nr 16. Från syndromtabellen kan man läsa av att bimängdsledare är $\boldsymbol{u}_2 = 0100000$. Efter det går det att avkoda $\boldsymbol{w}$ som $\boldsymbol{w}+\boldsymbol{u}_2 = 1101001$.*

Definition 28 (Utökad Hammingkod)**.** *Den utökade binära Hammingkoden skrivs $\overline{Ham(r,2)}$ (överstruken) och fås från $Ham(r,2)$ genom att lägga till en paritetscheckkoordinat.*

Man kan visa att den utökade Hammingkoden, $\overline{Ham(r,2)}$, har lägre överföringshastighet (ty den har hastighet $\frac{2^r-1-r}{2^r}$ medan $Ham(r,2)$ har $\frac{2^r-1-r}{2^r-1}$) än $Ham(r,2)$ men att den passar bättre för icke-fullständig avkodning (se Exempel 16 och Kapitel 2.3).

4.3 Konstruktion

För en godtycklig $(n, M, d)-$kod C på $\mathbb{F}_q$ vill man att både

$$R(C) = \frac{log_q M}{n} \text{ och } \delta(C) = \frac{d-1}{n}$$

ska vara stora. Det vill säga man vill ha så stort M som möjligt för givna n och d. Idealet är en kod som har storlek lika med $A_q(n,d)$ för givna q, n och d. Som vi sett tidigare får man ofta nöja sig med att komma nära $A_q(n,d)$ (Definition 26).

Följande två satser behövs för vår Reed-Muller kod i nästa kapitel men vi tar upp dem redan här och i samband med detta definierar vi även unionen av två binära ord.

Definition 29. *(union)*
Om $\boldsymbol{x} = x_1...x_n$ och $\boldsymbol{y} = y_1...y_n$ är två binära ord så definierar vi unionen mellan $\boldsymbol{x}$ och $\boldsymbol{y}$ som

$$\boldsymbol{x} \cup \boldsymbol{y} = (x_1 \dots x_n y_1 \dots y_n)$$

Sats 11 ($\boldsymbol{u}$,$\boldsymbol{u}+\boldsymbol{v}$ konstruktion)**.** *Låt C_i vara en $[n, k_i, d_i]-$lineär kod över $\mathbb{F}_q$, $i = 1, 2$. Då är koden C, definierad genom $C = \{(\boldsymbol{u},\boldsymbol{u}+\boldsymbol{v}) : \boldsymbol{u} \in C_1, \boldsymbol{v} \in C_2\}$, en $[2n, k_1 + k_2, min\{2d_1, d_2\}]-$lineär kod över $\mathbb{F}_q$.*

Bevis. Bevis finns bland annat i Xing [5] Kapitel 6. □

Sats 12. *Låt D vara en binär $[n, k, d]-$lineär kod. Då är koden C, som definieras genom $C = \{(\boldsymbol{u},\boldsymbol{u}) : \boldsymbol{u} \in D\} \cup \{(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{1}) : \boldsymbol{u} \in D\}$,*
en binär $[2n, k+1, min\{n, 2d\}]-$lineär kod.

Bevis. Resultatet nås genom att låta $C_1 = D$ och $C_2 = \{\boldsymbol{0}, \boldsymbol{1}\}$ i Sats 11. □

4.4 Tillämpning

Exempel 18 (Pixlar)**.** *Följande exempel visar ganska tydligt hur man kan använda koder på ett enkelt sätt. Exemplet finns i bland annat Biggs [2]. Vi tänker oss ett A4-papper med en svart-vit bild på. Pappret har vita (V) och svarta (S) pixlar. Låt säga att sannolikheten för att hitta en viss pixel på godtycklig plats är $0,8$ för vit och $0,2$ för svart. Som ett mått på informationen per pixel (egentligen är detta 'the* Shannon Entropy*' som representerar maximal kompressionsnivå) använder vi $0,8\ log_2(1/0,8) + 0,2$*

$log_2(1/0,2) \approx 0,722$.

Om man kodar varje pixel för sig är den uppenbara kodningen $V = 0$ *och* $S = 1$ *(eller tvärtom) lämpligast. Denna kod har medelordlängd* $L_1 = 0,2 \cdot 1 + 0,8 \cdot 1 = 1$ *vilket ska jämföras med* $0,722$. *Ett papper med* N *pixlar innehåller teoretiskt bara* $0,722N$ *bitar information enligt ovan.*

Genom att öka blocklängden till 2 *tar vi ett steg närmare detta tal. Det går till så här. Klumpa ihop pixlarna två och två och anpassa koden. Fyra möjligheter finns*

Pixlar	*Sannolikhet*	*Kod*
VV	*0,64*	*0*
VS	*0,16*	*10*
SV	*0,16*	*110*
SS	*0,04*	*111*

En kod kan göras exempelvis som i tredje kolumnen det vill säga $VV = 0$, $VS = 10$, $SV = 110$, $SS = 111$. *Medelordlängden blir nu* $L_2 = 0,64 \cdot 1 + 0,16 \cdot 2 + 0,16 \cdot 3 + 0,04 \cdot 3 = 1,56$. *Varje kodord representerar två pixlar så hela pappret kodas med hjälp av* $N/2$ *kodord. Vi har* $L_2 N/2 = 0,78N$. *En avsevärd förbättring.*

En fortsättning med blockkod av längd 3 *ger* 8 *kodord och medelordlängd* $= 2,184$. *Detta ger* $0,728N$ *bitar. Nu har vi kommit ganska nära* $0,722$ *men man kan fortsätta tills man är nöjd. Datakompressionenen som används i datorer bygger på det här sättet att tänka.*

5 Reed-Muller koder

En mycket användbar kod av felrättande typ är Reed-Muller koden, $RM(r, m)$ uppkallad efter matematikerna D.E. Muller och Irving S. Reed på 50-talet. Det är en binär, lineär kod som har många tillämpningar (specialfall av Reed-Muller koden inkluderar även 'the Hadamard code', 'the Walsh–Hadamard code' och 'the Reed–Solomon code').

Ett känt (i såna här sammanhang) exempel på tillämpning är $RM(1, 5)-$koden som användes av Rymdsonden Mariner 9 för att skicka svart-vita bilder av Mars till Jorden på 1970-talet. $RM(1, 5)-$koden har längd $2^5 = 32$ och minimumavstånd 16. Den kunde och kan således rätta 7 och detektera 15 fel. Detta var en stor fördel eftersom avståndet till Mars är långt och överföringen var osäker. Reed-Muller koden tillåter en speciell avkodning kallad 'Reed

decoding' som vi tittar närmare på i det sista avsnittet i den här uppsatsen.

I korthet gäller för koden att för varje positivt heltal r och m som uppfyller $0 \leq r \leq m$ finns en r:te ordningens Reed-Muller kod $RM(r,m)$ som är en binär lineär kod med parametrar

$$[2^m, \binom{m}{0} + \binom{m}{1} + ... + \binom{m}{r}, 2^{m-r}]$$

Här kommer för det mesta gälla att $r = 1$ men det finns några generaliseringar och kommentarer för $r > 1$.

5.1 Definition

För $r = 0$, som är trivial, definierar vi

Definition 30. $(RM(0,m))$
Den 0*:te ordnigens Reed-Muller kod,* $RM(0,m)$*, för* $m \geq 0$ *definieras som repetitionskoden* $\{0...0, 1...1\}$*, där längden är* 2^m*.*

Koden innehåller alltså två kodord, varav det första har Hammingvikten 0 och det andra Hammingvikten 2^m.

Definition 31. $(RM(1,m)$*, rekursiv*$)$
Den första ordningens Reed-Muller kod $RM(1,m)$ *är binära koder rekursivt (referens till sig själv) definierade för alla heltal* $m \geq 1$ *enligt följande*
(i) $RM(1,1) = \mathbb{F}_2^2 = \{00, 01, 10, 11\}$
(ii) *för* $m+1 \geq 2$ *så är* $RM(1, m+1) =$

$$\{(\boldsymbol{u},\boldsymbol{u}) : \text{där } \boldsymbol{u} \in RM(1,m)\} \quad \cup \quad \{(\boldsymbol{u},\boldsymbol{u}+\boldsymbol{1}) : \text{där } \boldsymbol{u} \in RM(1,m)\}$$

Exempel 19. $(RM(1,2)-$*kod*$)$
Med hjälp av (ii) *och till exempel* $\boldsymbol{u} = 00$ *får vi att två av kodorden är* $\{00, 00\} \cup \{00, 11\}$ *vilket ger* 0000 *och* 0011*. Fortsätt på samma sätt med de övriga tre kodorden från* $RM(1,1)$ *så får vi fram alla* 8 *kodord i* $RM(1,2)$

$$\begin{array}{cccc} 0000 & 0011 & 0101 & 0110 \\ 1001 & 1010 & 1100 & 1111 \end{array}$$

Med våra tidigare beteckningar är det en $[4,3,2]-$*lineär kod. En genereringsmatris för* $RM(1,2)$ *är*

$$G_{RM(1,2)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Exempel 20. $(RM(1,3)-$*kod*$)$
På samma sätt men med lite mer arbete kan man skriva upp alla kodord i $RM(1,3)$*. De är* 16 *stycken*

00000000	00001111	00110011	00111100
01011010	01010101	01100110	01101001
10010110	10011001	10100101	10101010
11000011	11001100	11110000	11111111

och bildar en $[8, 4, 4]-$*lineär kod.*

Det är inte svårt att gissa en genereringsmatris, som ju ska ha 4 rader, men vi återkommer till den. Först det allmänna resultatet av det vi sett ovan.

Sats 13. *För* $m \geq 1$ *så är* $RM(1, m)$ *en binär* $[2^m, m+1, 2^{m-1}]-$*lineär kod där varje kodord utom* $\mathbf{0}$ *och* $\mathbf{1}$ *har Hammingvikt* 2^{m-1}

Bevis. Det är klart att $RM(1, 1)$ är en binär $[2, 2, 1]-$lineär kod. Vi noterar att $RM(1, m)$ även kan fås från $RM(1, m-1)$ genom konstruktionen i Sats 12, förra kapitlet. Vi använder induktion och antar att $RM(1, m-1)$ är en binär $[2^{m-1}, m, 2^{m-2}]-$lineär kod. Från Sats 12 får vi att $RM(1, m)$ är en binär

$$[2 \cdot 2^{m-1}, m+1, min\{2 \cdot 2^{m-2}, 2^{m-1}\}] = [2^m, m+1, 2^{m-1}]-\text{lineär kod}$$

Visa nu att förutom för $\mathbf{0}$ och $\mathbf{1}$ så har varje kodord i $RM(1, m+1)$ Hammingvikten $2^{(m+1)-1} = 2^m$. Enligt Definition 31 är ett ord i $RM(1, m+1)$ antingen av typen $(\boldsymbol{u},\boldsymbol{u})$ eller $(\boldsymbol{u},\boldsymbol{u}+\mathbf{1})$, där $\boldsymbol{u}$ är ett ord i $RM(1, m)$.

Fall 1. $(\boldsymbol{u},\boldsymbol{u})$, där $\boldsymbol{u} \in RM(1, m)$:
Notera att $\boldsymbol{u}$ inte kan vara $\mathbf{0}$ eller $\mathbf{1}$ för då är $(\boldsymbol{u},\boldsymbol{u})$ noll-vektorn eller ettvektorn. Så, från induktionsantagandet har vi att $\boldsymbol{u}$ har Hammingvikten 2^{m-1}. Därför har $(\boldsymbol{u},\boldsymbol{u})$ Hammingvikten $2^{m-1} + 2^{m-1} = 2 \cdot 2^{m-1} = 2^m$

Fall 2. $(\boldsymbol{u},\boldsymbol{u}+\mathbf{1})$, där $\boldsymbol{u} \in RM(1, m)$:

a) Om $\boldsymbol{u}$ inte är $\mathbf{0}$ eller $\mathbf{1}$ så har den vikten $2^{m-1} = \frac{2^m}{2}$ på grund av $RM-$kodens konstruktion, dvs exakt hälften av dess koordinater är 1. Så hälften av koordinaterna i $\boldsymbol{u}+\mathbf{1}$ är också 1 (den andra hälften), dvs Hammingvikten av $\boldsymbol{u}+\mathbf{1}$ är också 2^{m-1}. Därför är Hammingvikten av $(\boldsymbol{u},\boldsymbol{u}+\mathbf{1})$ exakt 2^m

b) Om $\boldsymbol{u} = \mathbf{0}$ så är $\boldsymbol{u}+\mathbf{1} = \mathbf{1}$ så Hammingvikten av $(\mathbf{0}, \mathbf{1})$ är 2^m

c) Om $\boldsymbol{u} = \mathbf{1}$ så är $\boldsymbol{u}+\mathbf{1} = \mathbf{0}$ och Hammingvikten av $(\mathbf{1}, \mathbf{0})$ är åter 2^m

□

Med hjälp av satsen ovan (eller Exempel 19) ser vi att $RM(1, 2)$ har minimumavstånd $2^{2-1} = 2$. Det innebär att den detekterar 1 fel, Sats 2. På samma sätt har $RM(1, 3)$ minimumavstånd $2^{3-1} = 4$ och kan därför rätta 1 fel enligt Sats 3.

I likhet med ovan kan man också ta fram en genereringsmatris för $RM(1, m+1)$ med hjälp av genereringsmatrisen för $RM(1, m)$.

Sats 14. *För första ordningens Reed-Muller kod gäller att*
(i) En genereringsmatris för $RM(1,1)$ *är* $G_{RM(1,1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$
(ii) Om $G_{RM(1,m)}$ *är en genereringsmatris för* $RM(1, m)$ *så får vi en genereringsmatris för* $RM(1, m+1)$ *genom*

$$G_{RM(1,m+1)} = \begin{pmatrix} G_{RM(1,m)} & G_{RM(1,m)} \\ 0...0 & 1...1 \end{pmatrix}$$

Bevis. (i) är uppenbart, (ii) är en direkt följd av resultatet i Sats 12. □

Exempel 21. *Med hjälp av* $G_{RM(1,2)}$ *från Exempel 19 och Sats 14 får vi att en genereringsmatris (som inte är på standardform) för* $RM(1, 3)$ *är (detta sätt att skriva matrisen, inklusive namnen på radvektorer, är vanligt i samband med RM-koden)*

$$G_{RM(1,3)} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

5.2 Kodning

I det här och nästa avsnitt, som handlar om avkodning, kommer jag använda mig av $RM(1, 3)-$koden. Dess dimension enligt den allmänna formeln ges av

$$k = \begin{pmatrix} m \\ 0 \end{pmatrix} + \begin{pmatrix} m \\ 1 \end{pmatrix} + \begin{pmatrix} m \\ 2 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ r \end{pmatrix} = 4 \text{ då } r = 1,\ m = 3$$

Alltså samma som antalet rader i genereringsmatrisen. Om vi vill skicka ett meddelande gör vi det i block om längden 4. Låt (uncoded message)

$$M_{uc} = (a_0 a_1 a_2 a_3)$$

vara ett sådant block, där varje a_i är ett tecken.

Det kodade meddelandet blir $M_c = \sum_{i=0}^{3} a_i G_{RM(1,3)_i}$ där $G_{RM(1,3)_i}$ är rad i från genereringsmatrisen. Mer om kodning finns i avsnittet Kodning med lineär kod, Kapitel 3.4.

Exempel 22. *(Kodning)*
Kodning av meddelandet $M_{uc} = 1010$ *ger med hjälp av genereringsmatrisen för* $RM(1, 3)$ *(se Exempel 21) kodordet,*

$$M_c = 1 \cdot (11111111) + 0 \cdot (00001111) + 1 \cdot (00110011) + 0 \cdot (01010101) =$$
$$= 11111111 + 00110011 = 11001100$$

Vi noterar att det kodade meddelandet, M_c, är dubbelt så långt som det ursprungliga meddelandet, M_{uc}. Eftersom meddelandet vi vill koda bara har kodsymbolen 1 på position 1 och 3 används bara dessa rader från genereringsmatrisen. Just den informationen är vad man vill få fram för att kunna avkoda ett ord man mottagit.

5.3 Avkodning

Det finns flera metoder att tillgå för avkodning av Reed-Muller koden. Syndromavkodningen från Kapitel 3.5.2 är en av dem. Vi ska här titta på metoden som nämndes i början av kapitlet och som är mycket vanlig för Reed-Muller kod (eng. 'Reed decoding'). Den bygger på majoritetsavkodning och görs i flera steg. Majoritetsavkodning innebär helt enkelt att man använder det värde (0 eller 1 för binära koder) som är i majoritet efter att man genomfört sina uträkningar (se nedan).

För avkodningen fortsätter vi med den Reed-Muller kod vi nu är bekanta med, nämligen $RM(1,3)$. Först behöver vi definiera snittet mellan två binära lineära koder (inte helt olikt unionen i Defintion 29).

Definition 32. *(snitt)*
Om $\boldsymbol{x} = x_1 \dots x_n$ och $\boldsymbol{y} = y_1 \dots y_n$ är två binära ord så definierar vi snittet mellan x och y som

$$\boldsymbol{x} \cap \boldsymbol{y} = (x_1 \cdot y_1 \dots x_n \cdot y_n)$$

$\boldsymbol{x} \cap \boldsymbol{y}$ är alltså 1 på den i:te positionen om både $\boldsymbol{x}$ och $\boldsymbol{y}$ är 1 på den i:te positionen.

Vår avkodning använder sig av Booleska monomialer och polynom (bland annat Roman [6] Kapitel 6. Vi definerar dem.

Definition 33. *(Boolesk monomial, polynom och reducerad form)*
En Boolesk monomial i variablerna $x_1, \dots, x_m$ är ett uttryck på formen $x_{i_1} x_{i_2} \dots x_{i_s}$, där varje x_i är vald från mängden $x_1, \dots, x_m$. Dess reducerade form fås genom att använda de två allmänna reglerna $x_i x_j = x_j x_i$ och $x_i^2 = x_i$. Monomialens grad ges av antalet variabler i den reducerade formen.

Ett Booleskt polynom, $p(x_1, \dots, x_m)$, är en lineärkombination av Booleska monomialer (med koeffecienter i $\mathbb{Z}_2$). Det Booleska polynomet är i sin tur på reducerad form om alla monomialer är på reducerad form och eventuella monomialer som är lika har tagits bort. Det Booleska polynomets grad bestäms av den monomial som har högst grad i den reducerade formen.

Exempel 23. *Det är lättare att se vad som händer i en tabell så vi gör en sådan. De tre första kolumnerna innehåller möjliga kombinationer av* 0 *och* 1*. Här är* $m = 3$ *vilket ger* $2 \cdot 2 \cdot 2 = 8$ *rader. Det är lämpligt att (som här) ha dem i storleksordning. Notera likheten med genereringsmatrisen. Den sista kolumnen är polynomets värde, som ju beror på polynomkoeffecienterna, men just här visas* $p(x_1, x_2, x_3) = 1 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3$

x_1	x_2	x_3	p
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Transponatet av p *är ett kodord och kommer att vara mycket intressant för oss varför vi ger det ett namn,* $a_p = 00111100$*. För första ordningens* $RM-$*kod blir det alltid lika många nollor som ettor, utom för kodorden* 00000000 *och* 11111111*. Detta visades i Sats 13.*

En allmän definition av Reed-Muller koden (jämför med tidigare Definition 31) kan nu göras med hjälp av Booleska polynom.

Definition 34. $(RM(r, m))$
Låt $0 \leq r \leq m$*. Den* r*:te ordningens Reed-Muller kod är mängden av alla binära ord* a_p *av längd* $n = 2^m$ *associerade med de Booleska polynomen* $p(x_1, ..., x_m)$ *med grad högst* r*.*

Exempel 24. *För* $RM(1, 3)$ *har vi* $r = 1$*,* $m = 3$ *och* $n = 2^3 = 8$*. Monomialerna är* $\{1, x_1, x_2, x_3\}$*. Monomialen* 1 *har grad 0 och monomialerna* x_1, x_2 *och* x_3 *har grad 1. De vektorer som associeras med dessa är samma som i genereringsmatrisen, Exempel 21. Koden ges nu av mängden binära ord associerade med de Booleska polynomet* $p(x_1, x_2, x_3)$*, som har grad högst* 1 *(inga potenser). Det har därför formen*

$$p(x_1, x_2, x_3) = a_0 1 + a_1 x_1 + a_2 x_2 + a_3 x_3$$

där $a_i = 0$ *eller* 1*. Med hjälp av generingsmatrisens rader i Exempel 21 blir detta*

$$a_0(11111111) + a_1(00001111) + a_2(00110011) + a_3(01010101).$$

Räknar man ut dessa kombinationer får man fram samma 16 *kodord som i Exempel 20. Förutom de kodord vi döpte i Exempel 21 kan man till exempel få fram att* $a_0 = 0$, $a_1 = 1$, $a_2 = 1$ *och* $a_3 = 0$ *dvs det Booleska polynomet* $x_1 + x_2$ *svarar mot kodordet* $00001111 + 00110011 = 00111100$.

5.3.1 Euklidisk geometri

Det sista vi behöver innan vi kommer igång med själva avkodningen är några rader Euklidisk geometri.

Definition 35. *(*$k-$*flat)*
Låt S *vara en* $k-$*dimensionell delmängd av* $\mathbb{Z}_2^m$*, dvs dimension* m*. Bimängderna till* S *skrivs som* $b + S$*, där* $b \in \mathbb{Z}_2^m$*. Då* $dim(S) = k$ *kallar vi* $b + S$ *för* $k-$*flat.*

Eftersom S är en lineär kod (Definition 12) har den en paritetscheckmatris, säg H. Vektorerna i S kan beskrivas genom

$$x \in S \iff xH^T = 0$$

Det gäller också att

$$x \in b + S \iff x - b \in S$$

vilket inträffar precis då $(x - b)H^T = 0$. Vi sammanfattar och ser att

$$x \in b + S \text{ om och endast om } xH^T = bH^T$$

Med geometriska termer är vektorrummet $\mathbb{Z}_2^m$ den Euklidiska geometrin, $EG(m, 2)$, av dimension m på $\mathbb{Z}_2$.

Om $dim(S) = 0$ så är bimängderna $b + S$ för $0-$flater mängder med en punkt. Dessa kan skrivas $b+\{0\} = \{b\}$ och representerar punkter i $EG(m, 2)$. Linjerna i $EG(m, 2)$ kallas $1-$flater, består av två punkter, och är på formen

$$\{b, b + c\} \quad \text{där } c \in \mathbb{Z}_2^m - \{0\}$$

Exempel 25. *Vektorrummet* $\mathbb{Z}_2^2$*, dvs den Euklidiska geometrin* $EG(2, 2)$ *har* $2^2 = 4$ *stycken* $0-$*flater (punkter) och* $\binom{4}{2} = 6$ *stycken* $1-$*flater (linjer).*

$EG(3, 2)$ *har* $2^3 = 8$ *stycken* $0-$*flater,* $\binom{8}{2} = 28$ *stycken* $1-$*flater (och även* $\binom{8}{3}/\binom{4}{3} = 14$ *stycken* $2-$*flater).*

Raderna från de tre första kolumnerna i tabellen från Exempel 23 bildar de 8 *tredimensionella punkterna i* $EG(3, 2)$.

I Exempel 23 kan man betrakta ordet från den sista kolumnen, $a_p = 00111100$, som en *karakteristisk vektor* för mängden {010,011,100,101} i EG(3,2) eftersom dessa är punkterna i $EG(3,2)$ som hör till a_p på det platser där a_p är 1. Mängden F ={010,011,100,101} är en 2−flat och bildar ett hyperplan i EG(3,2). Den är helt bestämd av de tre första vektorerna eftersom den fjärde är en summa av dessa.

Sats 15. *Reed-Muller koden $RM(1,m)$, $m \leq 3$, spänns upp av de karakteristiska vektorerna för alla flater av dimension $k = m-1$*

Bevis. För $m = 1, k = 0$ är $RM(1,1) = \{00,01,10,11\}$. $\mathbb{Z}_2^1$ dvs $EG(1,2)$ ger $2^1 = 2$ punkter och $\binom{2}{2} = 1$ linje. De karakteristiska vektorerna är 01 och 10 för 0−flaterna. Dessa spänner upp $RM(1,1)$−koden. Se också Definition 9.

För $m = 2$ är $RM(1,2) = \{0000,0011,0101,0110,1001,1010,1100,1111\}$. $\mathbb{Z}_2^2$ dvs $EG(2,2)$ ger $(2^2) = 4$ punkter och $\binom{4}{2} = 6$ linjer. De karakteristiska vektorerna 0110, 0011 och 1100 för 1−flaterna spänner upp $RM(1,2)$−koden.

För $m = 3$ spänns $RM(1,3)$−koden upp av fyra av de 14 karakteristiska vektorerna 00111100, 01101001, 01010101 och 10101010 för 2−flaterna. □

Sats 16. *Om F och E är flater i $EG(m,2)$ så gäller att $a_F \cdot a_E \equiv |F \cap E|$ mod 2.*

Bevis. $|F|$ anger storleken av F, som i Definition 1. Vänsterledet är = 0 om skalärprodukten är 0 eller jämnt tal och 1 annars. Det gäller att $a_F \cdot a_E \equiv wt(a_F \cap a_E) = wt(a_{F\cap E}) = |F \cap E|$ □

Sats 17. *Låt $F = \boldsymbol{b} + S$ och $E = \boldsymbol{c} + T$ vara flater i $EG(m,2)$ och $\boldsymbol{b}$ och $\boldsymbol{c}$ tillhöra $\mathbb{Z}_2^m$. Då gäller antingen $F \cap E = \emptyset$ eller $F \cap E = \boldsymbol{x} + (S \cap T)$ för något icke-noll $\boldsymbol{x}$ i $F \cap E$.*

Bevis. Om $F \cap E = \emptyset$ så är det klart. Så vi kan anta att $F \cap E \neq \emptyset$. Låt x vara ett (eller det enda) elementet i $F \cap E$. Visa nu
(1) att alla element $x + r \in F \cap E$, där $r \in S \cap T$
(2) att varje element y i $F \cap E$ har formen $x + r$ för något r, där $r \in S \cap T$.

För att visa (1) så vet vi att $x = b + s$ eftersom $x \in b + S$, om nu $r \in S \cap T$, och speciellt S, så gäller också att $x + r = b + (s + r) \in b + S$ eftersom $s + r \in S$ som är ett vektorrum. På samma sätt gäller att $x + r \in c + T$, och därför tillhör $x + r$ snittet mellan $b + S$ och $c + T$ dvs $F \cap E$.

För att visa (2) antag att också y tillhör $F \cap E$. Då gäller att $x = b + s$ och $y = b + s_1$ så $x - y$ tillhör S. På samma sätt gäller också $x - y \in T$ och därmed $x - y \in T \cap S$. Kalla slutligen $x - y$ för $-r$, då är $y = x + r$, vilket skulle visas. □

Sats 18. *Alla flater i EG(3,2) utom* $0-$*flaterna har ett jämnt antal vektorer.*

Bevis. Resultatet följer av resonemanget i beviset för Sats 15. □

5.3.2 Algoritm

Enligt Definition 34 kommer ett kodord i $RM(r,m)$ från det Booleska polynomet

$$p(x_1,\ldots,x_m)=\sum_{s=0}^{r}\sum_{i_1,\ldots,i_s} a_{i_1,\ldots,i_s}x_{i_1}\ldots x_{i_s}=a_0+a_1x_1+\cdots+a_{12}x_1x_2+\ldots$$

med grad högst r. Vi kan skriva

$$a_p=\sum_{s=0}^{r}\sum_{i_1,\ldots,i_s} a_{i_1,\ldots,i_s}a_{x_{i_1}\ldots x_{i_s}} \tag{1}$$

där $a_{x_{i_1}\ldots x_{i_s}}$ är den karakteristiska vektor som motsvarar monomialen $x_{i_1}\cdots x_{i_s}$. Till exempel $a_{x_1x_2}$ =00000011 för $x_1\cdot x_2$, se tabellen på sid 38. Vi är nu intresserade av att beräkna koeffecienterna i $p(x_1,\ldots,x_m)$ Om vi kan beräkna varje koeffecient på flera sätt kan vi använda vår majoritetsavkodning för att besluta vilket värde vi ska ta när värdena skiljer sig åt. Det kommer visa sig att detta låter sig göras genom att multiplicera a_p med olika vektorer. Då $r>1$ får man arbeta sig neråt via $r, r-1, r-2$ osv.

För $RM(1,3)-$koden har vi $r=1, m=3$ så

$$a_p=a_0 1+\sum_{i=1}^{m}a_i a_{x_i}$$

Låt

$$\{k\}^c=\{j_1,\ldots,j_{m-1}\}$$

där komplementet tas med avseende på $\{1,\ldots,m\}$ som ju är $\{1,2,3\}$ för $RM(1,3)-$koden och om $\{k\}=\{3\}$ så är $\{k\}^c=\{1,2\}$.

Beräkna sen skalärprodukten

$$a_p\cdot a_{x_{j_1}\cdots x_{j_{m-1}}} \tag{2}$$

eftersom det är en skalärprodukt kan vi beräkna den för varje term i a_p separat. Från Sats 16 har vi att

$$a_{x_i}\cdot a_{x_{j_1}\ldots x_{j_{m-1}}}\equiv |F_{x_i}\cap F_{x_{j_1}\ldots x_{j_{m-1}}}|\ mod\ 2\equiv |F_{x_ix_{j_1}\ldots x_{j_{m-1}}}|\ mod\ 2$$

Där F_{x_1} är är mängden (flaten) som x_1 är karakteristisk vektor för. Med hjälp av Sats 18 blir detta 0 utom då $F_{x_i x_{j_1} \dots x_{j_{m-1}}}$ är en enda vektor (0−flat). Vilket inträffar om och endast om

$$\{i, j_1, \dots, j_{m-1}\} = \{1, \dots, m\}$$

alltså om

$$\{i\} = \{j_1 \dots j_{m-1}\}^c = \{k\} \tag{3}$$

För $RM(1,3)$ och fortfarande $\{k\} = \{3\}$ skulle (3) bli

$$\{3\} = \{1,2\}^c = \{3\}$$

Så genom att beräkna (2) får vi tag i den önskade koeffecienten

$$a_p \cdot a_{x_{j_1} \dots x_{j_2}} = a_i$$

Detta är bara en beräkning av koeffecienten. Vi generaliserar genom att ta skalärprodukten

$$a_p \cdot a_{b + F_{x_{j_1} \dots x_{j_2}}}$$

där $b + F_{x_{j_1} \dots x_{j_2}}$ är en translation av flaten $F_{x_{j_1} \dots x_{j_2}}$

I Euklidisk geometri är en translation en flytt av varje punkt i en given riktning och längd. Alltså addition av en konstant vektor till varje punkt. Translationerna av en vektor är isometriska.

I det här fallet får vi

$$a_{x_i} \cdot a_{b + F_{x_{j_1} \dots x_{j_2}}} \equiv |F_{x_i} \cap (b + F_{x_{j_1} \dots x_{j_2}})| \ mod\ 2$$

Produkten kommer vara noll utom då högerledet

$$|F_{x_i} \cap (b + F_{x_{j_1} \dots x_{j_2}})| = 1$$

Om $\boldsymbol{x} \in F_{x_i} \cap (b + F_{x_{j_1} \dots x_{j_2}})$ så kan vi använda Sats 17 som ger att

$$F_{x_i} \cap (b + F_{x_{j_1} \dots x_{j_2}}) = \boldsymbol{x} + (F_{x_i} \cap F_{x_{j_1} \dots x_{j_2}}) = \boldsymbol{x} + F_{x_i x_{j_1} \dots x_{j_2}}$$

och detta är 0−flat om och endast om $F_{x_i x_{j_1} \dots x_{j_{m-1}}}$ består av en ensam vektor, som tidigare. Alltså,

$$a_p \cdot a_{b + F_{x_{j_1} \dots x_{j_{m-1}}}} = a_i$$

där $\{j_1 \dots j_{m-1}\} = \{k\}^c$ och b är godtycklig vektor i $EG(3,2)$. Eftersom det finns 2^{m-1} stycken translationer av 1−flaten

$$F_{x_{j_1} \dots x_{j_2}} \tag{4}$$

får vi 2^{m-1}, alltså 4 för $RM(1,3)$, uttryck för varje koeffecient a_i. Om alla fyra ger samma värde har inga fel inträffat. Annars kan vi avgöra rätt värde genom att välja det som är vanligast förekommande. Förutsatt att högst $(2^{m-1}-1)/2$ fel inträffat.

Nästa tabell visar binära ord och motsvarande monomialer i $RM(1,3)-$koden. Det underlättar att ha den framför sig vid avkodning.

Ord	1	x_1	x_2	x_3	$x_1 \cdot x_2$	$x_2 \cdot x_3$	$x_1 \cdot x_3$
000	1	0	0	0	0	0	0
001	1	0	0	1	0	0	0
010	1	0	1	0	0	0	0
011	1	0	1	1	0	1	0
100	1	1	0	0	0	0	0
101	1	1	0	1	0	0	1
110	1	1	1	0	1	0	0
111	1	1	1	1	1	1	1

Sammanfattningsvis, avkodning av $RM(1,m)-$kod:

- Undersök koeffecienterna för monomialer med grad 1 och beräkna 2^{m-1} karakteristiska vektorer för varje sådan.
- Bestäm koeffecienterna genom att välja det vanligast förekommande värdet.
- Avgör om koeffecienten a_0 är 0 eller 1 (se Exempel 26).
- Undersök om det finns några fel i det mottagna kodordet.

Exempel 26. *Avkodningsexempel för $RM(1,3)$.*
Det kodade meddelandet (coded) $M_c = (c_0c_1...c_7)$, det mottagna meddelandet (received) $M_r = (r_0r_1...r_7)$ och det eventuella felet (error) $M_e = (e_0e_1...e_7)$ förhåller sig till varandra genom

$$M_r = M_c + M_e \text{ som också är lika med } a_p$$

Antag att vi vill avkoda $M_c = 00111100$ men precis ett fel har inträffat så det mottagna meddelandet faktiskt är $M_r = 10111100$. Vi ser i tabellen ovan att M_c motsvarar flaten $\{010, 011, 100, 101\}$. När meddelandet kodades (med hjälp av genereringsmatrisen från Exempel 21) gjordes det på följande sätt

$$M_c = (a_0a_1a_2a_3)G_{RM(1,3)} = a_0 1 + a_1x_1 + a_2x_2 + a_3x_3$$

Vi vill beräkna koeffecienterna a_0, a_1, a_2 och a_3. Som vi sett tidigare kommer vi få fyra ekvationer för varje koeffecient. Om det uppstått precis ett fel kommer tre av dem att ge samma resultat och den fjärde ett annat resultat. Om

inga fel uppstått kommer alla fyra beräkningar ge samma resultat.

Flaterna som är intressanta är

Flaten för $a_{x_1x_2}$ *är* $F_{x_1x_2} = \{110, 111\}$

Flaten för $a_{x_2x_3}$ *är* $F_{x_2x_3} = \{011, 111\}$

Flaten för $a_{x_1x_3}$ *är* $F_{x_1x_3} = \{101, 111\}$

Låt oss börja med a_3 *alltså* $k = 3$ *och* $\{k\}^c = \{1, 2\}$ *och se vilka translationer som finns. Det bör finnas fyra stycken...Translationerna av flaten* $F_{x_1x_2}$

med b=000 är $\{110, 111\}$ *som har karakteristisk vektor 00000011*

med b=011 är $\{100, 101\}$ *som har karakteristisk vektor 00001100*

med b=100 är $\{010, 011\}$ *som har karakteristisk vektor 00110000*

med b=110 är $\{000, 001\}$ *som har karakteristisk vektor 11000000*

Vi beräknar nu, för varje b, de fyra skalärprodukterna

$$a_3 = a_p \cdot a_{b+F_{x_{j_1}\cdots x_{j_2}}}$$

Det ger,

$$10111100 \cdot 00000011 = 0$$

$$10111100 \cdot 00001100 = 0$$

$$10111100 \cdot 00110000 = 0$$

$$10111100 \cdot 11000000 = 1$$

Vårt majoritetsval säger oss att koeffecienten a_3 *är* 0*. På samma sätt får man fram att koeffecienten* $a_2 = a_1 = 1$*. För att beräkna* a_0 *gör så här*

$$M_r - a_1a_2a_3 \cdot G_{RM(1,3)_{2,3,4}} = 10111100 - 110 \cdot G_{RM(1,3)_{2,3,4}}$$
$$= 10111100 - 00111100 = 10000000 = a_0 \cdot 1 + M_e$$

Om a_0 *är* 1 *så är felet* 01111111*, dvs 7 stycken fel. Om* a_0 *är* 0 *så är felet* 10000000*, dvs 1 fel. Varför vi av sannolikhetsskäl såklart drar slutsatsen att* $a_0 = 0$*.*

Eftersom koden klarar att rätta ett fel kommer det inte hända att vi står inför vårt majoritetsval med 2 st 0*:or och 2 st* 1*:or.*

Vi kan sätta samman de fyra koeffecienterna, $0, 1, 1$ *och* 0*, och se att det ursprungliga meddelandet var* $M_{uc} = 0110$*. Man kan med hjälp av* M_e *också se att det var det första tecknet som var fel i* $M_r = 10111100$*. Det skulle ha varit* $M_r - M_e = M_r + M_e = 00111100$*.*

Sakregister

6 Appendix

6.1 Kodord

$RM(0,1)$

00 11

$RM(1,1)$

00	01
10	11

$RM(1,2)$

0000	0011	0101	0110
1001	1010	1100	1111

$RM(1,3)$

00000000	00001111	00110011	00111100
01011010	01010101	01100110	01101001
10010110	10011001	10100101	10101010
11000011	11001100	11110000	11111111

$RM(1,4)$

0000000000000000	0000000011111111	0000111100001111	0000111111110000
0011001111001100	0011001100110011	0011110000111100	0011110011000011
0110011010011001	0101101010100101	0110100110010110	0101010110101010
0101010101010101	0101101001011010	0110100101101001	0110011001100110
1111000000001111	1111000011110000	1100110011001100	1001011010010110
1100001100111100	1100001111000011	1100110000110011	1010101010101010
1001100110011001	1010010110100101	1010101001010101	1001100101100110
1010010101011010	1001011001101001	1111111100000000	1111111111111111

6.2 Symbolförteckning

A	kodalfabet
a_0	koeffecient
c	element i C
C	mängd med kodord, lineär kod
$\overline{C}$	den utökade koden till C
$d(x, y)$	Hammingavstånd mellan x och y
d	minimumavståndet
$d(C)$	minimumavståndet i C
e_1	basvektor
$\mathbb{F}_q$	ändlig kropp
F	flat, mängd
G	genereringsmatris
H	paritetscheckmatris
I_k	identitetsmatris $k \times k$
k	dimension
m	dimension
$M = \|C\|$	antal kodord i C
M_{uc}	okodat meddelande
M_c	kodat meddelande
M_r	mottaget meddelande
M_e	meddelandets eventuella fel
n	kodordslängd
p	sannolikhet för fel
q	antal symboler i alfabetet A
$R(C)$	Informationsgrad i C
$< S >$	uppspänd av S
s_i	kodsymbol nr i från alfabetet A
r_i	rad nr i
v	binär vektor
v	binär vektor
w	binär vektor
$wt(x)$	Hammingvikten av x
x	kodord/vektor
y	kodord/vektor
$\lfloor \ \rfloor$	floor-function

7 Referenser

[1] Beachy J., Blair, W. Abstract Algebra, Second edition., Waveland Press 1990.

[2] Biggs, Norman L. An Introduction to Cryptography, Chapman-Hall/CRC, Taylor-Francis Group 2007.

[3] Bøgvad, Rikard Kompendium i Algebra 1, Stockholms Universitet.

[4] Ekedahl, Torsten Felrättande koder, Kompendium Stockholms Universitet.

[5] Ling, San and Xing, Chaoping Coding Theory, A First Course, Cambridge 2004.

[6] Roman, Steven Coding and Information Theory, Springer-Verlag 1992.

[7] Tengstrand, Anders Linjär algebra med vektorgeometri, upplaga 2:8, Studentlitteratur 2005.