TILLÄMPADE

DISKRETA STRUKTURER

Julius Brzezinski och Jan Stevens

MATEMATIK
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
GÖTEBORG 2001

FÖRORD

Termen Diskret matematik" täcker ett mycket brett spektrum av olika matematiska ämnen som på ett eller annat sätt är relaterade till datoranvändningar. Ofta har man att göra med mycket gamla och mycket varierande matematiska teorier som plötsligt visade sig vara av stor betydelse i datateknikens tjänst. Därför är det helt omöjligt att täcka all relevant matematik i en kort kurs. Vi har valt en inriktning som har mycket stark algebraisk prägel – kursen handlar i stor utsträckning om tillämpad algebra. Efter en teoretisk inledning som introducerar en rad viktiga algebraiska strukturer – grupper, ringar och kroppar, övergår vi till flera tillämpningar. Vi diskuterar algebraiska koder, krypteringssystem, Booleska algebror, ändliga automater, snabba Fouriertransformer och deras tillämpningar på snabba multiplikationsalgoritmer.

Trots att vi har placerat teoridelen först i kompendiet, kan olika kapitel läsas i en icke-linjär ordning som framgår av innehållsbeskrivningen på nästa sida. Detta gör att olika tillämpningar kan varvas med mera teoretiska avsnitt, vilket förhoppningsvis kommer att underlätta inlärningsprocessen.

Texten kom till under en relativt kort tid då det visade sig att den gamla kurslitteraturen inte var tillgänglig. Därför kan man vänta sig en rad brister i de första upplagorna. Vi är mycket tacksamma för alla kommentarer om både oklarheter och tryckfel. Skicka gärna dessa till Julius Brzezinski (jub@math.chalmers.se) eller Jan Stevens (**stevens@math.chalmers.se**).

INNEHÅLLSBESKRIVNING

Kapitel 1 – 10 innehåller grundläggande egenskaper hos mycket viktiga algebraiska strukturer – grupper, ringar och kroppar. De efterföljande kapitlen handlar om fyra tillämpningsområden.

Det första tillämpningsområdet är algebraiska koder. "En kort inledning till algebraiska koder” samt Tipsproblemet" kan läsas utan några som helst förkunskaper, men i den givna ordningen. Kapitlet Polynomkoder” kräver både det inledande kapitlet om algebraiska koder och alla kapitel 1 – 10.

Det andra tillämpningsområdet handlar om kryptering och kräver kapitel 1 – 5.

Den tredje tillämpningen är relaterad till digital teknik. Kapitlet om Booleska algebror kräver endast kapitel 1 – 3 (i mycket begränsad omfattning), däremot kapitlet om linjära automater bygger på alla kapitel 1 – 10 och kapitlet om Booleska algebror.

Den sista tillämpningen handlar om snabba Fouriertransformer och snabba multiplikationsalgoritmer. Dessa kapitel förutsätter innehållet i kapitel 1 – 9.

Som appendix har vi bifogat ett kapitel om logiska konnektiv och kvantorer.

INNEHÅLL

Kapitel 1

DELBARHET OCH PRIMTAL

Vi börjar med en kort repetition av några viktiga egenskaper hos **heltalen**:

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \ldots\}.$$

(1.1) Definition. Om a och b är två heltal så säger man att b **delar** a om $a = bq$, där q är ett heltal. Man säger också att a är **delbart** med b eller att a är en **multipel** av b. Man skriver då $b|a$. □

Exempel. $3|6$, $641|2^{32}+1$ (det är inte så lätt att bevisa - se dock övning 5.3 i Kapitel 5.) □

Rent allmänt gäller följande viktiga och välkända egenskap:

(1.2) Divisionsalgoritmen. *Om a och b är heltal och $b \neq 0$ så är*

$$a = bq + r, \quad \textit{där} \;\; 0 \leq r < |b|.$$

Både q och r är definierade entydigt av a och b.

Bevis. Betrakta alla heltal $a - bx$, där x är ett godtyckligt heltal. Bland dessa tall finns det positiva ty olikheten $a - bx > 0$ har med all säkerhet heltaliga lösningar ($x > a/b$ då $b > 0$ och $x < a/b$ då $b < 0$). Låt r vara det minsta icke-negativa heltalet bland talen $a - bx$ då x är ett heltal och låt $r = a - bq$. Vi påstår att $0 \leq r < |b|$. Annars är, $r \geq |b|$ så att $0 \leq r - |b| < r$ och $r - |b| = a - bq - |b| = a - b(q \pm 1)$ dvs $r - |b|$ är ett icke-negativt tal på formen $a - bx$ som är mindre än r. Detta strider mot definitionen av r. Alltså har vi

$$a = bq + r \quad \text{och} \quad 0 \leq r < |b|.$$

Bevis att q och r definieras entydigt av a och b lämnar vi som övning 1.1. □

(1.3) Definition. Om $a = bq + r$, där $0 \leq r < |b|$ (som i Divisionsalgoritmen ovan) så kallas q **kvoten**, och r **resten** vid division av a med b. □

Ofta utnyttjar man följande egenskaper hos delbarhetsrelationen:

(1.4) Proposition. *Låt a, b, c, d beteckna heltal. Då gäller:*

(a) *om $d|a$ och $d|b$ så $d|a \pm b$,*

(b) *om $a|b$ och $b|c$ så $a|c$,*

(c) *om i likheten $a + b = c$ är två av talen a, b, c delbara med d så är också det tredje talet delbart med d,*

(d) *om $a|b$ och $b|a$ så är $b = \pm a$.*

Alla dessa egenskaper är mycket enkla och vi lämnar ett bevis som övning (se övning 1.2).

Med **största gemensamma delaren** till a och b menar man ett positivt heltal d som delar a och b och är delbart med varje gemensam delare till a och b. Den största gemensamma delaren till a och b är definierad entydigt därför att om både d och d' är sådana delare så gäller $d|d'$ och $d'|d$, vilket innebär att $d' = \pm d$. Men både d och d' är positiva så att $d' = d$. Största gemensamma delaren till a och b betecknas med $SGD(a, b)$. Man brukar definiera $SGD(0, 0) = 0$.

Med **minsta gemensamma multipeln** till a och b menar man ett positivt heltal m som är delbart med a och b och som delar varje gemensam multipel av a och b. Även minsta gemensamma multipeln av a och b definieras entydigt av dessa tal (motivera detta påstående med liknande argument som för $SGD(a, b)$ ovan!). Minsta gemensamma multipeln av a och b betecknas med $MGM(a, b)$. Som för SGD definierar man $MGM(0, 0) = 0$.

Följande egenskap av största gemensamma delaren till två heltal kommer att användas flera gånger under kursens gång.

(1.5) Proposition. *Om a och b är heltal och $d = SGD(a, b)$ så existerar två heltal x_0 och y_0 sådana att*

$$d = ax_0 + by_0.$$

Bevis. Om $a = b = 0$ så är påståendet klart (som x och y kan man välja helt godtyckliga heltal). Anta att a eller b inte är 0. Det är klart att det finns positiva heltal som kan skrivas på formen $ax + by$ t ex om $a \neq 0$ så är $\pm a = a \cdot (\pm 1) + b \cdot 0$ och antingen a eller $-a$ är ett positivt heltal. Även $b = a \cdot 0 + b \cdot 1$ kan skrivas på formen $ax + by$. Låt d_0 vara det minsta positiva heltal som kan skrivas på den önskade formen dvs

$$(*) \qquad d_0 = ax_0 + by_0.$$

Vi påstår att $d_0 = d$. Först observerar vi att varje heltal $ax + by$ är delbart med d_0, ty

$$ax + by = qd_0 + r,$$

där resten r är mindre än delaren d_0. Men

$$r = a(x - qx_0) + b(y - qy_0)$$

så att r måste vara 0 ty annars får man ett tal som är mindre än d_0 och som kan skrivas på den önskade formen. Alltså dividerar d_0 både a och b ty bägge kan skrivas på formen $ax + by$. Ekvationen $(*)$ säger att om d' är en delare till a och b så är d' en delare till d_0. Alltså är d_0 den största gemensamma delaren till a och b. □

Den sista propositionen säger inte hur man kan hitta x och y. För det mesta spelar det inte någon större roll – existensen är helt tillräcklig. Men ibland vill man beräkna x_0 och y_0. Det gör man ofta (och ganska snabbt) med hjälp av **Euklides algoritm**. Euklides algoritm säger hur man kan beräkna $SGD(a, b)$. Man bildar en divisionskedja:

$$\begin{array}{rcll}
a & = & bq_1 + r_1, & 0 \le r_1 < |b|, \\
b & = & r_1q_2 + r_2, & 0 \le r_2 < r_1, \\
r_1 & = & r_2q_3 + r_3, & 0 \le r_3 < r_2, \\
\vdots & & \vdots & \vdots \\
r_{n-3} & = & r_{n-2}q_{n-1} + r_{n-1}, & 0 \le r_{n-1} < r_{n-2}, \\
r_{n-2} & = & r_{n-1}q_n + r_n, & 0 \le r_n < r_{n-1}, \\
r_{n-1} & = & r_nq_{n+1}. &
\end{array}$$

Varje kedja av den här typen måste vara ändlig därför att en avtagande kedja av resterna $r_1 > r_2 > r_3 > \ldots \ge 0$ måste vara ändlig. Vi påstår att den sista icke-försvinnande resten i denna kedja, dvs r_n, är den största gemensamma delaren till a och b. Att det verkligen är sant kontrollerar man mycket enkelt med hjälp av definitionen av $SGD(a, b)$. Den sista likheten i kedjan säger att r_n är delaren till r_{n-1}. Alltså visar den näst sista likheten att r_n är delaren till r_{n-2}. Nu vet vi att r_n delar r_{n-1} och r_{n-2}. Alltså visar likheten för r_{n-3} att även denna rest är delbar med r_n. Vi fortsätter vår vandring uppåt och steg efter steg visar vi att alla tal r_{n-1}, r_{n-2}, r_{n-3}, …, r_1, b, a är delbara med r_n. Alltså är r_n en gemensam delare till a och b.

Om nu d är en godtycklig gemensam delare till a och b så visar den första likheten att d delar r_1. Alltså ger den andra likheten att d delar r_2. Då vi vet att d delar r_1 och r_2 så får vi ur den tredje likheten att d också delar r_3. På det sättet får vi att d är en delare till alla tal i sekvensen $a, b, r_1, r_2, r_3, \ldots, r_{n-2}, r_{n-1}, r_n$. Detta visar att r_n är den största gemensamma delaren till a och b. Det är klart att man kan formalisera vårt resonemang genom att använda matematisk induktion.

Med hjälp av Euklides algoritm kan man inte bara beräkna $SGD(a, b)$ utan också två heltal x, y sådana att $SGD(a, b) = ax + by$. Vi illustrerar detta med ett exempel:

(1.6) Exempel. Låt $a = 2406$ och $b = 654$. Euklides algoritm ger

$$\begin{aligned}
2406 &= 654 \cdot 3 + 444 \\
654 &= 444 \cdot 1 + 210 \\
444 &= 210 \cdot 2 + 24 \\
210 &= 24 \cdot 8 + 18 \\
24 &= 18 \cdot 1 + 6 \\
18 &= 6 \cdot 3
\end{aligned}$$

så att $SGD(2406, 654) = 6$ (den sista nollskilda resten). Nu har vi

$$\begin{aligned}
6 &= \underline{24} - \underline{18} \cdot 1 = \underline{24} - (\underline{210} - \underline{24} \cdot 8) \cdot 1 = \\
&= \underline{24} \cdot 9 - \underline{210} = (\underline{444} - \underline{210} \cdot 2) \cdot 9 - \underline{210} = \\
&= \underline{444} \cdot 9 - \underline{210} \cdot 19 = \underline{444} \cdot 9 - (\underline{654} - 444 \cdot 1) \cdot 19 = \\
&= \underline{444} \cdot 28 - \underline{654} \cdot 19 = (\underline{2406} - \underline{654} \cdot 3) \cdot 28 - \underline{654} \cdot 19 = \\
&= \underline{2406} \cdot 28 + \underline{654} \cdot (-103)
\end{aligned}$$

□

Det finns en annan möjlighet att beräkna $SGD(a, b)$ då a och b är två heltal. Även om denna möjlighet inte är särskilt praktisk används den flitigt i skolan. Den bygger på faktoruppdelningar av heltal i produkt av primtal.

Man säger att ett positivt heltal p är ett **primtal** om p har exakt två olika delare: 1 och sig självt. Primtalen mindre än 100 är

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.$$

Följden av primtalen är oändlig. Detta påstående visades för mer än 2000 år sedan av Euklides. Innan vi visar Euklides sats tittar vi närmare på primtalens viktiga roll som byggstenar för alla heltal – varje heltal större än 1 är en produkt av primtal. Vi skall visa detta påstående om en liten stund. Först behöver vi en mycket viktig egenskap hos primtalen:

(1.7) Sats. *En primdelare till en produkt av två heltal är en delare till (minst) en av faktorerna dvs om $p|ab$ så $p|a$ eller $p|b$, då p är ett primtal och a, b är heltal.*

Bevis. Antag att $p \nmid a$. Då är $SGD(p, a) = 1$ därför att p är ett primtal. Enligt (1.5) existerar två heltal x, y sådana att $px + ay = 1$. Om man multiplicerar den likheten med b får man $b = pbx + aby$. Men enligt förutsättningen är $ab = pq$ för ett heltal q. Alltså är $b = p(bx + qy)$ dvs $p|b$. □

Nu kan vi visa satsen om faktoruppdelningar av heltal i produkter av primtal:

(1.8) Aritmetikens fundamentalsats. *Varje heltal större än 1 är en entydig produkt av primtal dvs om*

$$n = p_1 p_2 \cdots p_m = p'_1 p'_2 \cdots p'_n,$$

där p_i och p'_j är primtal så är $m = n$ och vid en lämplig numrering av faktorerna är $p_i = p'_i$.

Bevis. Först visar vi med induktion att varje heltal $N > 1$ är en produkt av primtal. Vi början med $N = 2$ då vårt påstående gäller. Låt $N > 2$ och antag att varje positivt heltal större än 1 och mindre än N är en produkt av primtal. Låt p beteckna den minsta delaren till N. Det är klart att p är ett primtal, ty motsatsen innebär att p har en delare $d \neq 1, p$ så att $1 < d < p$ och d är en delare till N (se (1.2) (b)) som är mindre än p. Vi har $N = pq$, där $1 \leq q < N$. Men om $q > 1$ så är q en produkt av primtal enligt induktionsantagandet, vilket visar att N också är en sådan produkt.

Entydigheten visar vi med induktion med avseende på summan $s = m + n$. Om $s = 2$ så har vi $m = n = 1$ och $p_1 = p'_1$. Antag att vårt påstående gäller då antalet faktorer är mindre än s och låt

$$p_1 p_2 \cdots p_m = p'_1 p'_2 \cdots p'_n,$$

där $m + n = s$. Primtalet p_m är en delare till produkten till höger så att enligt (1.7) är p_m en delare till en av faktorerna. Genom att eventuellt nummrera om dessa faktorer kan vi anta att $p_m | p'_n$. Men båda dessa tal är primtal så att $p_m = p'_n$. Alltså gäller

$$p_1 p_2 \cdots p_{m-1} = p'_1 p'_2 \cdots p'_{n-1},$$

och i denna likhet är antalet primfaktorer lika med $s - 2 < s$. Enligt induktionsantagandet är antalet faktorer till vänster lika med antalet faktorer till höger dvs $m - 1 = n - 1$. Alltså är $m = n$. Dessutom kan man numrera faktorerna så att $p_i = p'_i$ då $i = 1, \ldots, n - 1$. □

(1.9) Anmärkning. Ofta kallar man sats (1.7) för aritmetikens fundamentalsats. Även om formuleringen ovan handlar om positiva heltal så kan vi säga rent allmänt att varje heltal $N \neq \pm 1$ är en produkt

$$N = \varepsilon p_1 p_2 \cdots p_n,$$

där p_i är primtal och $\varepsilon = \pm 1$. Eligt aritmetikens fundamentalsats är sådan framställning entydig så när som på faktorernas ordningsföljd. Faktoruppdelningar av liknande typ är kända t ex för polynom. Vi diskuterar både faktoruppdelningar för heltalen och för polynom i ett senare kapitel. □

Nu kan vi bevisa att det finns oändligt många primtal.

(1.10) Euklides sats. *Det finns oändligt många primtal.*

Bevis. Antag att $p_1, p_2, \ldots, p_n$ är alla primtal. Bilda talet

$$N = p_1 p_2 \cdots p_n + 1.$$

Talet N är större än 1 så att det måste vara en produkt av primtal dvs något av primtalen $p_1, p_2, \ldots, p_n$ är en delare till N. Låt oss beteckna en sådan delare med p dvs $N = pq$, där p är ett av primtalen $p_1, p_2, \ldots, p_n$. Alltså är

$$1 = N - p_1 p_2 \cdots p_n = p(q - \frac{p_1 p_2 \cdots p_n}{p}).$$

Detta betyder att p dividerar 1, vilket är helt orimligt eftersom primtalet p är större än 1. Vårt antagande att det endast finns ändligt många primtal har lett oss till en motsägelse. Alltså måste antagandet vara falskt, dvs det finns oändligt många primtal. □

ÖVNINGAR

1.1. Visa att kvoten och resten vid division av två heltal är entydigt definierade dvs om $a = bq + r = bq' + r'$, där $a, b \neq 0, q, r, q', r'$ är heltal och $0 \leq r < |b|, 0 \leq r' < |b|$, så är $q = q'$ och $r = r'$.

1.2. Visa Proposition (1.4).

1.3. Faktoruppdela följande tal i produkt av primtal:

(a) 2704, (b) 392688, (c) 749088, (d) 8051.

1.4. Beräkna $SGD(a, b)$ samt två heltal x och y sådana att $SGD(a, b) = ax + by$ då

(a) $a = 577, b = 257$,

(b) $a = 1111, b = 1133$.

1.5. Låt a och b vara två heltal. Visa att $SGD(a, b)MGM(a, b) = ab$.

Kapitel 2

RELATIONER

Begreppet “relation” i matematiska sammanhang anknyter till betydelsen av samma ord i vardagliga situationer då en relation är ofta ett samband mellan två individer (dvs ett par).

(2.1) Definition. Med en **relation** R på en mängd X menas en godtycklig mängd bestående av par (x, y), där $x, y \in X$. Med andra ord är en relation på X en godtycklig delmängd R till den kartesiska produkten

$$X \times X = \{(x, y) : x, y \in X\}.$$

$\square$

Om $x, y \in X$ och $(x, y) \in R$, där R är en relation på X så skriver man ofta $x \sim y$. Men “ $\sim$ ” ersätts oftast med andra tecken som traditionellt betecknar kända relationer t ex med “ $\leq$ ” eller “$|$”.

(2.2) Exempel. (a) Låt $X = \{1, 2, 3, 4\}$ och låt $R = \{(1, 3), (2, 4), (2, 2), (4, 4)\}$. Man kan skriva $1 \sim 3$ eller $2 \sim 2$. Man har sammanlagt 16 par (x, y), men endast 4 par ingår i relationen R.

(b) Låt $X = \mathbb{R}$ vara mängden av de reella talen. Definiera $R = \{(x, x^2) : x \in \mathbb{R}\} \subset X \times X$. Relationen R är helt enkelt grafen av funktionen $f(x) = x^2$ dvs den består av alla punkter på parabeln $y = x^2$. Här har vi $x \sim y$ precis då $y = x^2$. $\square$

Ett så allmänt relationsbegrepp är inte särskilt användbart. Men i matematiska situationer har man överallt olika relationer som satisfierar olika ytterligare villkor. Vi diskuterar först ekvivalensrelationer och därefter, mycket kort, ordningsrelationer och funktionsgrafer.

(2.3) Definition. En relation " $\sim$ " på en mängd X kallas för en **ekvivalensrelation** om

(a) $x \sim x$ (reflexivitet),

(b) $x \sim y$ implicerar $y \sim x$ (symmetri),

(c) $x \sim y$ och $y \sim z$ implicerar $x \sim z$ (transitivitet),

då $x, y, z \in X$. □

(2.4) Exempel. (a) Låt $X = \mathbb{Z}$ och låt $x \sim y$ då och endast då $5 \mid x - y$ för $x, y \in \mathbb{Z}$. Då gäller $x \sim x$, ty $5|x - x = 0$, $x \sim y$ implicerar $y \sim x$, ty $5|x - y$ implicerar $5|y - x = -(x - y)$ samt $x \sim y$ och $y \sim z$ ger $x \sim z$, ty $5|x - y$ och $5|y - z$ ger $5|x - z = (x - y) + (y - z)$.

(b) Låt $X = \mathbb{N} = \{1, 2, \ldots\}$ och låt $x \sim y$ då och endast då x och y har exakt samma primtalsdelare. Man kontrollerar mycket lätt att " $\sim$ " är en ekvivalensrelation (gör det!).

(c) Låt X vara en mängd och låt X_i vara icke-tomma delmängder till X för i tillhörande en indexmängd I. Låt oss anta att dessa mängder utgör en **partition** av X, vilket betyder att $X = \cup X_i$ är unionen av alla X_i och X_i är parvis disjunkta dvs $X_i \cap X_j = \emptyset$ om $i \neq j$. Definiera nu $x \sim y$ om och endast om det finns i så att $x, y \in X_i$. Man får en ekvivalensrelation på X. Man kan tänka på X som mängden av alla elever i en skola medan X_i betecknar alla elever i samma klass (vi förutsätter att skolan är av "gammal modell" så att varje elev tillhör exakt en klass). Två elever x och y är relaterade (dvs $x \sim y$) precis då x och y går i samma klass. Vi visar strax att varje ekvivalensrelation på en godtycklig mängd X får man på detta sätt. □

(2.5) Definition. Låt $\sim$ vara en ekvivalensrelation på en mängd X. Med **ekvivalensklassen** av $x \in X$ menas mängden

$$[x] = \{y \in X : y \sim x\}.$$

□

(2.6) Proposition. *(a)* $x \in [x]$.

(b) $[x] = [y] \Leftrightarrow x \sim y$.

(c) Två olika ekvivalensklasser är disjunkta.

(d) X är unionen av alla ekvivalensklasser.

Bevis. (a) Klart från (2.3) (a).

(b) $[x] = [y] \Rightarrow x \in [x] = [y] \Rightarrow x \sim y$. Antag nu att $x \sim y$. Om $z \in [x]$ så ger $z \sim x$ och $x \sim y$ att $z \sim y$ så att $z \in [y]$. Alltså är $[x] \subseteq [y]$. Av symmetriskäl har man också $[y] \subseteq [x]$.

(c) Om $z \in [x] \cap [y]$ så är $z \sim x$ och $z \sim y$ så att $x \sim y$ ur symmetrin och transitiviteten ($z \sim x$ ger $x \sim z$ som med $z \sim y$ ger $x \sim y$). Enligt (b) är $[x] = [y]$. Detta betyder att om $[x] \neq [y]$ så saknar dessa klasser något gemensamt element z.

(d) Följer direkt ur (a). □

(2.7) Följdsats. *Ekvivalensklasserna av varje ekvivalensrelation på X bildar en partition av X.*

Bevis. Följer omedelbart från (c) och (d) i (2.6). □.

(2.8) Exempel. (a) För ekvivalensrelationen i (2.4) (a) har man

$$[x] = [r],$$

där r är resten vid division av x med 5 ty $5|x - r$ dvs $x \sim r$. Eftersom det finns 5 olika rester r så finns det exakt 5 olika ekvivalensklasser [0], [1], [2], [3], [4].

(b) I exempel (2.4) (b) är alla ekvivalensklasser av följande form: $[x] = [p_1 p_2 \cdots p_r]$, där $p_1, p_2, \ldots, p_r$ är alla olika primdelare till x om $x \neq 1$ och [1] (bestående av enbart 1). Kontrollera detta påstående!

(c) I exempel (2.4) (c) är just partitionsmängderna X_i ekvivalensklasserna, ty om x tillhör X_i så är $[x] = X_i$. □

Mängden av alla ekvivalensklasser för en ekvivalensrelation " $\sim$ " på X betecknas med $X/\sim$. Denna mängd kallar man ofta för X **modulo** $\sim$.

En annan mycket vanlig typ av relationer är ordningsrelationer.

(2.9) Definition. En relation " $\leq$ " på en mängd X kallas en **partiell ordningsrelation** (eller en **partiell ordning**) om

(a) $x \leq x$ (reflexivitet),

(b) $x \leq y$ och $y \leq z$ implicerar att $x \leq z$ (transitivitet),

(c) $x \leq y$ och $y \leq x$ implicerar att $x = y$ (antisymmetri).

Man skriver $x < y$ om $x \leq y$ och $x \neq y$. Om dessutom en relation " $\leq$ " satisfierar

(d) för godtyckliga $x, y \in X$ gäller det att $x < y$ eller $y < x$ eller $x = y$

så säger man att relationen är en **ordningsrelation** (eller en **ordning**) på X. □

(2.10) Exempel. (a) Låt $X = \mathbb{R}$ och låt $x \leq y$ betecknar den vanliga ordningsrelationen på de reella talen. Vi vet mycket väl att den relationen är en ordningsrelation i enlighet med definitionen ovan.

(b) Låt $X = \mathbb{N} = \{1, 2, 3, \ldots\}$ vara mängden av de naturliga talen. Relationen $x|y$ är en partiell ordningsrelation på $\mathbb{N}$ ty $x|x$, om $x|y$ och $y|z$ så $x|z$ samt $x|y$ och $y|x$ ger $x = y$. Men "|" är inte en ordningsrelation, ty (d) i definitionen ovan gäller inte då man t ex väljer $x = 2$ och $y = 3$. □

(2.11) Vi avslutar med en observation att varje funktion $f : X \to X$ definierar en relation – nämligen mängden av alla par $(x, f(x)) \in X \times X$. Låt oss påminna att med en funktion från en mängd X till en mängd Y menar man vanligen en regel som mot varje $x \in X$ ordnar exakt ett element $y \in Y$. Då skriver man $y = f(x)$ och $f : X \to Y$. I vårt fall har vi $X = Y$ och vi får en relation på X då $x \sim y$ om och endast om $y = f(x)$. Parmängden som svarar mot f består alltså av alla par $(x, f(x))$.

$$\Gamma_f = \{(x, f(x)) : x \in X\}$$

kallas ofta **grafen** av funktionen f.

ÖVNINGAR

2.1. Vilka av de följande relationerna på den givna mängden X är ekvivalensrelationer:

(a) $X = \mathbb{Z}$, $x \sim y$ då och endast då $n | x - y$, där n är ett fixt positivt heltal.

(b) $X = \mathbb{N}$, $x \sim y$ då och endast då xy är en kvadrat av ett naturligt tal.

(c) $X = \mathbb{R}^2$, $(a, b) \sim (c, d)$ då och endast då $b = d$.

(d) $X = \mathbb{R}^2$, $(a, b) \sim (c, d)$ då och endast då $a = c$ eller $b = d$.

(e) $X = \mathbb{R}$, $a \sim b$ då och endast då $a - b$ är ett heltal.

(f) $X = \mathbb{R}$, $a \sim b$ då och endast då $ab > 0$.

2.2. Är det sant att reflexivitet i definitionen av en ekvivalensrelation följer ur symmetrin och transitivitet enligt följande resonemang: Låt $x \in X$. Att $x \sim y$ ger $y \sim x$ eftersom “ $\sim$ ” är symmetrisk. Alltså ger transitiviteten $x \sim x$.

2.3. Bestäm ekvivalensklasserna i alla fall då relationerna i den första övningen är ekvivalensrelationer. Försök tolka ekvivalensklasserna geometriskt då sådana tolkningar är möjliga.

2.4. Vilka av följande relationer på de givna mängderna X är partiella ordningsrelationer? Vilka av dem är ordningsrelationer?

(a) $X = \mathbb{R}$, $a \sim b$ då och endast då $a^2 \leq b^2$.

(b) $X = \mathbb{N}$, $a \sim b$ då och endast då $a^2 | b^2$.

(c) $X =$ alla reella funktioner $f : \mathbb{R} \to \mathbb{R}$ och $f \sim g$ då och endast då $f(x) \leq g(x)$ för varje $x \in \mathbb{R}$.

Kapitel 3

MÄNGDER MED OPERATIONER

De fyra räknesätten: addition, subtraktion, multiplikation och division är, vad man ofta kallar (aritmetiska) operationer i mängden av alla tal. Addition och multiplikation av vanliga funktioner kända från analyskurser är också operationer. Även matrisaddition eller matrismultiplikation är operationer i mängden av matriser av lämplig storlek.

I algebran är man ofta intresserad av olika egenskaper hos operationer. Två mängder som tillåter operationer med samma egenskaper kan ofta studeras samtidigt – man behöver inte bevisa samma satser flera gånger om man vet att dessa satser gäller för varje mängd med operationer som satisfierar vissa villkor. I detta avsnitt definierar vi begreppet operation och några mycket allmänna egenskaper hos operationer, t ex associativitet och kommutativitet.

Begreppet operation är ett specialfall av begreppet funktion. Därför repeterar vi först att med en funktion f från en mängd X till en mängd Y menar man vanligen en regel som till varje $x \in X$ ordnar exakt ett element $y \in Y$. Då skriver man $y = f(x)$ och $f : X \to Y$. Låt oss också repetera att $X \times Y$ betecknar (den kartesiska) produkten av mängderna X och Y dvs

$$X \times Y = \{(x, y) : x \in X \text{ och } y \in Y\}.$$

Nu är vi beredda att definiera begreppet operation:

(3.1) Definition. Med en **(binär) operation** på mängden M menar man en avbildning från $M \times M$ till M. Bilden av paret (a, b) betecknas ofta med $a * b$, och mängden M med operationen "$*$" med $(M, *)$. □

Definitionen säger att en operation på M ordnar mot två godtyckliga element $a, b \in M$ ett element $a * b \in M$. Här följer några exempel på operationer:

(3.2) Exempel. (a) Låt M vara en av mängderna $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ och låt $a * b = a + b$ vara den vanliga summan av a och b.

(b) Med samma M som i (a), låt $a * b = ab$ vara den vanliga produkten av a och b.

(c) Låt $M = M_2(\mathbb{R})$ vara mängden av (2×2)-matriser med reella element och $A * B = AB$ den vanliga matrisprodukten för $A, B \in M_2(\mathbb{R})$.

(d) Låt M vara mängden av alla reella funktioner och $f * g = f + g$ den vanliga summan av två funktioner $f, g \in M$ dvs $(f + g)(x) = f(x) + g(x)$ då $x \in \mathbb{R}$. □

Enbart det faktum att man har en operation på en mängd är oftast inte tillräckligt för att studera mängden. Därför vill man veta lite mera om olika egenskaper hos operationer.

(3.3) Definition. Man säger att operationen $*$ på M är **associativ** om $(a*b)*c = a*(b*c)$ då $a, b, c \in M$. Operationen är **kommutativ** om $a * b = b * a$ då $a, b \in M$. □

Exempel. (a) Alla operationer i Exempel (3.2) är associativa och enbart (3.2)(c) är inte kommutativ.

(b) Subtraktionen är varken kommutativ eller associativ på $\mathbb{Z}$ dvs om $a*b = a-b$ så gäller inte att $a*b = b*a$ eller $(a*b)*c = a*(b*c)$ ty vanligen $a-b \neq b-a$ och $(a-b)-c \neq a-(b-c)$. Bästa sättet att visa dessa påståenden är att ge exempel: t ex $2-3 \neq 3-2$ och $(3-2)-1 \neq 3-(2-1)$. □

(3.4) Definition. Man säger att $e \in M$ är ett **neutralt element** för operationen $*$ om $e*a = a*e = a$ då $a \in M$. Man säger att $a' \in M$ är en **invers** till $a \in M$ om $a*a' = a'*a = e$. □

Exempel. (a) 0 är ett neutralt element för additionen på $M = \mathbb{Z}$ (eller $\mathbb{Q}, \mathbb{R}, \mathbb{C}$) ty $0 + a = a + 0 = a$ då $a \in M$. Inversen till $a \in M$ är $-a$ ty $a + (-a) = (-a) + a = 0$. Inversen kallas här motsatta talet.

(b) Talet 1 är ett neutralt element för multiplikationen på M ur (a) ty $1 \cdot a = a \cdot 1 = a$ då $a \in M$. Inversen till $a \in M$ finns enbart då $a' = 1/a \in M$. Om $M = \mathbb{R}$ så har alla tal invers utom 0. Om $M = \mathbb{Z}$ så har enbart $a = \pm 1$ inverser (motivera varför!).

(c) Nollmatrisen

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

är ett neutralt element för matrisadditionen på $M = M_2(\mathbb{R})$. Inversen till $A \in M$ är då $-A$ ty $A + (-A) = (-A) + A = \mathbf{0}$. I stället för invers säger man då den motsatta matrisen. Enhetsmatrisen

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

är ett neutralt element för matrismultiplikationen ty $EA = AE = A$ då $A \in M$. Inversen till $A \in M$ är $A' = A^{-1}$ om $\det A \neq 0$. Om $\det A = 0$ så saknar A invers (om $AA' = E$ så ger $\det(AA') = \det A \det A' = \det E = 1$ motsägelsen $0 = 1$ om $\det A = 0$). □

(3.5) Proposition. *Ett neutralt element $e \in M$ är entydigt bestämt. Om operationen på M är associativ och $a \in M$ har invers så är den entydig.*

Bevis. Om även e' är ett neutralt element så har vi

$$e' = e * e' = e.$$

Låt a'_1 vara också en invers till a. Då gäller

$$a'_1 = a'_1 * e = a'_1 * (a * a') = (a'_1 * a) * a' = e * a' = a'.$$

□

(3.6) Anmärkning. Om $M = \{a_1, a_2, \ldots, a_n\}$ är en ändlig mängd så definierar man ofta operationer på M med hjälp av **"multiplikationstabeller"**:

$$\begin{array}{c|ccccc} * & a_1 & \ldots & a_j & \ldots & a_n \\ \hline a_1 & & & & & \\ \vdots & & & & & \\ a_i & & & a_i * a_j & & \\ \vdots & & & & & \\ a_n & & & & & \end{array}$$

Varje sådan tabell ger en operation på M. Med hjälp av tabellen kan man lätt avgöra om operationen på M är kommutativ (hur?) eller om det finns ett neutralt element (hur?). Men det är mycket besvärligare att avgöra om operationen är associativ. □

Mycket ofta betraktar man funktioner mellan mängder med operationer. Låt oss påminna om att en funktion $f : M \to M'$ kallas bijektiv om den avbildar olika element i M på olika element i M' så att varje element i M' är bilden av ett element i M. Särskilt viktiga är funktioner som respekterar operationer i följande mening:

(3.7) Definition. Låt $(M, *)$ och $(M', *')$ vara två mängder med operationer. Man säger att en funktion $f : M \to M'$ är en **homomorfism** om $f(a * b) = f(a) *' f(b)$ för alla $a, b \in M$. Om dessutom f är bijektiv så säger man att den är en **isomorfism**. Då skriver man $M \cong M'$.
□

(3.8) Exempel. (a) Låt $M = M' = \mathbb{Z}$ vara mängder med vanlig addition av heltalen och $f(n) = 2n$. Vi har

$$f(n_1 + n_2) = 2(n_1 + n_2) = 2n_1 + 2n_2 = f(n_1) + f(n_2)$$

så att f är en homomorfism. Men f är inte en isomorfism därför att f inte är bijektiv — bilder av f är endast jämna heltal.

(b) Betrakta $M = \mathbb{R}$ med addition som operation och de positiva reella talen $M' = \mathbb{R}_+$ med multiplikation som operation. Funktionen $f : M \to M'$, där $f(x) = 2^x$, är en homomorfism därför att

$$f(x_1 + x_2) = 2^{x_1+x_2} = 2^{x_1}2^{x_2} = f(x_1)f(x_2).$$

Funktionen f är som bekant bijektiv så att f är en isomorfism.

(c) Låt $M = V$ och $M' = W$ vara två vektorrum (över de reella talen) och låt $f : V \to W$ vara en linjär avbildning. Då är f en homomorfism ty

$$f(v_1 + v_2) = f(v_1) + f(v_2)$$

enligt definition av linjära avbildningar. f behöver inte vara en isomorfism. □

ÖVNINGAR

3.1. Vilka av följande operationer på $\mathbb{Z}$ är associativa, kommutativa, vilka har ett neutralt element? Varje gång då det finns ett neutralt element bestäm alla element som har invers.

(a) $m * n = mn + 1$
(b) $m * n = mn + m + n$
(c) $m * n = m^2 + n^2$
(d) $m * n = 2$
(e) $m * n = 2^{mn}$
(f) $m * n = SGD(m, n)$
(g) $m * n = \max(m, n)$
(h) $m * n = MGM(m, n)$

3.2. Hur många operationer finns det på en mängd med n element? Hur många av dessa är kommutativa?

3.3. Ge exempel på en mängd med en operation som är

(a) associativ, men ej kommutativ;
(b) kommutativ, men ej associativ.

3.4. Låt M vara en mängd med en operation $*$ och med ett neutralt element e. Visa att om

$$a * (b * c) = (a * c) * b \quad \text{för} \quad a, b, c \in M$$

så är operationen $*$ kommutativ och associativ.

3.5. (svårt?) Låt M vara en mängd med en operation $*$ sådan att

$$a * a = a \quad \text{och} \quad (a * b) * c = (b * c) * a$$

för $a, b, c \in M$. Visa att operationen är kommutativ och associativ.

3.6. Betrakta $\mathbb{Z}$ med operationen $m * n = mn + 1$. Definiera en operation på $\mathbb{R}$ så att funktionen $f\colon \mathbb{Z} \to \mathbb{R}$, $f(m) = m - 1$, blir en homomorfism.

3.7. Låt $f\colon M \to M'$ vara en isomorfism mellan två mängder med operationer $(M, *)$ och $(M', *')$ och $g\colon M' \to M$ dess invers. Visa att g är en homomorfism (och därmed själv en isomorfism).

Kapitel 4

GRUPPER: DEFINITIONER OCH EXEMPEL

En grupp är en mängd med en operation som uppfyller några mycket enkla villkor. Dessa enkla villkor leder till en mycket rik och intressant teori som har tillämpningar i hela matematiken och andra naturvetenskapliga ämnen (fysik, kemi). Grupper trädde in i matematiken redan under 1700-talet även om en formell definition av gruppbegreppet formulerades betydligt senare. Olika konkreta grupper studerades redan av L. Euler (restgrupper) och J. Lagrange som först introducerade begreppet permutationsgrupp. E. Galois visade hur man kan använda permutationsgrupper för att lösa viktiga och, under hans tid, mycket svåra problem i teorin för algebraiska ekvationer. Men den moderna definitionen av begreppet grupp gavs 1870 av L. Kronecker.

(4.1) Definition. Låt G vara en mängd och låt $*$ vara en operation på G, dvs

(0) $a * b \in G$ då $a, b \in G$ (slutenhet).

Man säger att $(G, *)$ är en **grupp** om

(1) $(a * b) * c = a * (b * c)$ då $a, b, c \in G$ (associativitet),

(2) det finns $e \in G$ så att $e * a = a * e = a$ då $a \in G$ (neutralt element),

(3) till varje $a \in G$ finns $a' \in G$ så att $a * a' = a' * a = e$ (invers). □

I varje grupp finns det endast ett neutralt element e och varje gruppelement a har endast en invers a'. Detta följer direkt från proposition (3.5).

(4.2) Exempel. (a) $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$, $(\mathbb{C}, +)$ är grupper. Om man utelämnar 0 ur $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ får man grupper med avseende på multiplikation. Det hjälper inte att utelämna 0 ur $\mathbb{Z}$

för att få en grupp med avseende på multiplikation därför att t ex heltalet 2 saknar heltalig invers (inversen i $\mathbb{Z}$ enbart existerar för ± 1).

(b) Alla $(n \times n)$-matriser med reella element och med determinant $\neq 0$ bildar en grupp med avseende på matrismultiplikation. Denna grupp har en standardbeteckning: $GL_n(\mathbb{R})$. Vi har

$$A, B \in GL_n(\mathbb{R}) \Rightarrow \det A \neq 0 \neq \det B \Rightarrow \det(AB) = \det A \cdot \det B \neq 0 \Rightarrow AB \in GL_n(\mathbb{R}),$$

vilket visar slutenheten. Associativiteten $(AB)C = A(BC)$ då $A, B, C \in GL_n(\mathbb{R})$ är en välkänd egenskap hos matrismultiplikation. Om E betecknar $(n \times n)$-enhetsmatrisen så är $EA = AE = A$ då $A \in GL_n(\mathbb{R})$ dvs E är det neutrala elementet. Slutligen $AA^{-1} = A^{-1}A = E$ om $A \in GL_n(\mathbb{R})$ dvs A^{-1} är inversen till A (observera att $\det A \neq 0$ så att inversen A^{-1} existerar).

(c) Låt $G = \{1, -1\}$ med vanlig multiplikation. G är en grupp med följande multiplikationstabell:

$$\begin{array}{r|rr} \cdot & 1 & -1 \\ \hline 1 & 1 & -1 \\ -1 & -1 & 1 \end{array}$$

$\square$

(4.3) Anmärkning. En multiplikationstabell för en ändlig grupp (som ovan) kallar man ofta för **grupptabell** eller **Cayleys tabell**. Det är inte lätt att avgöra om en operation på en ändlig mängd G definierar en grupp genom att studera grupptabellen:

$$\begin{array}{c|ccccc} * & a_1 & \ldots & a_j & \ldots & a_n \\ \hline a_1 & a_1 & \ldots & a_j & \ldots & a_n \\ \vdots & & & & & \\ a_i & a_i & \ldots & a_i * a_j & \ldots & \\ \vdots & & & & & \\ a_n & a_n & & & & \end{array}$$

Genom en inspektion av multiplikationstabellen inser man lätt om mängden är sluten med avseende på operationen – slutenheten innebär att varje element i tabellen tillhör mängden G. Det är också lätt att upptäcka om det finns ett neutralt element: om $a_1 = e$ är det neutrala elementet så är de första två raderna och kolonnerna identiska. Man kan se enkelt om varje element har invers – varje rad och varje kolonn måste innehålla e. I själva verket är det så att varje rad och varje kolonn är en omkastning av den första raden (eller kolonnen). Detta följer ur en mycket enkel observation i nästa proposition. Men att kontrollera att operationen är associativ är inte lika lätt. $\square$

(4.4) Proposition. *Låt G vara en grupp och $a, b, c \in G$. Då gäller strykningslagarna:*

(a) $a * c = b * c \Rightarrow a = b$,

(b) $c * a = c * b \Rightarrow a = b$.

Bevis. Vi visar (a). Multiplicera från höger med inversen c' till c. Tack vare associativiteten får vi att $a * c * c' = b * c * c'$ ger $a * e = b * e$ dvs $a = b$. □

(4.5) Anmärkning. En rad i tabellen ovan består av produkterna $a_i{*}a_1, \dots a_i{*}a_j, \dots, a_i{*}a_n$. Alla dessa produkter ger olika element i G därför att likheten $a_i * a_j = a_i * a_k$ ger att $a_j = a_k$ enligt strykningsegenskapen ovan. □

(4.6) Definition. Man säger att gruppen $(G, *)$ är **abelsk** (= kommutativ) om $a * b = b * a$ då $a, b \in G$. □

Exempel. Alla grupper i (4.2) (a) är abelska. Gruppen i (4.2) (b) är icke-abelsk då $n \geq 2$ ty vanligen $AB \neq BA$ för två $(n \times n)$-matriser. □

(4.7) Definition. Antalet element i en ändlig grupp kallas **gruppens ordning** och betecknas $o(G)$ (eller $|G|$). Om G inte är ändlig säger man att G har oändlig ordning och skriver $o(G) = \infty$. □

(4.8) Anmärkning. (a) När man definierar en grupp så beskriver man mängden G av dess element och gruppoperationen $*$. Formellt borde man säga att $(G, *)$ är en grupp. Trots detta säger man oftast att G är en grupp.

(b) Vi vet redan att symbolen "$*$" som betecknar en operation kan tolkas på olika sätt. När det gäller beteckningar finns det två vanliga typer som dels beror på traditionen dels på bekvämligheten. Det är bekvämare att skriva ab i stället för $a * b$. Då talar man om **multiplikativ notation**. Inversen betecknas då med a^{-1}. Ibland är denna notation inte helt naturlig, speciellt när gruppoperationen är addition. Då använder man **additiv notation**, dvs man tolkar "$*$" som "$+$". Villkoren (0) - (3) i (4.1) har då följande form:

(0) $a, b \in G \Rightarrow a + b \in G$

(1) $(a + b) + c = a + (b + c)$ då $a, b, c \in G$.

(2) Det finns $e \in G$ så att $e + a = a + e = a$ (man skriver ofta $e = 0$).

(3) Till varje $a \in G$ finns $a' \in G$ så att $a + a' = a' + a = e$ (man skriver ofta $a' = -a$). □

I exempel (4.2) har vi ett antal grupper med avseende på addition:

$$\mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}.$$

Man säger då att $\mathbb{Z}$ är en delgrupp till $\mathbb{Q}$ (eller $\mathbb{R}$ eller $\mathbb{C}$), $\mathbb{Q}$ är en delgrupp till $\mathbb{R}$ (eller $\mathbb{C}$) osv. Formellt har vi:

(4.9) Definition. Låt $H \subseteq G$. Man säger att H är en **delgrupp** till G om elementen i H bildar en grupp med avseende på operationen i G. □

(4.10) Proposition. *Låt $H \subseteq G$. H är en delgrupp till G då och endast då*

(a) $a, b \in H \Rightarrow a * b \in H$,

(b) $e \in H$,

(c) $a \in H \Rightarrow a^{-1} \in H$.

Bevis. Se övn. 4.7. □

(4.11) Cykliska grupper Låt G vara en grupp och $g \in G$. Elementet g definierar en delgrupp till G – den minsta delgrupp till G som innehåller g. Den måste innehålla alla potenser av G dvs $g, gg, ggg, \ldots$, deras inverser $g^{-1}, g^{-1}g^{-1}, g^{-1}g^{-1}g^{-1}, \ldots$ och e. Vi betecknar med g^n produkten $gg \ldots g$ av n stycken faktorer g, med g^{-n} potensen $(g^{-1})^n$, och med g^0 elementet e. Man visar lätt likheten $g^m g^n = g^{m+n}$ för godtyckliga hela m och n. Potenserna g^n, $n \in \mathbb{Z}$, bildar en delgrupp till G som innehåller g. Den betecknas med $< g >$ och kallas **den cykliska gruppen genererad av** g. Antalet element i $< g >$ kallas **ordningen av** g och betecknas $o(g)$. Ibland händer det att $G =< g >$. Då säger man att G är en **cyklisk grupp** och g är dess generator. Då är $G = \{g^n : n \in \mathbb{Z}\}$. Observera att med den additiva notationen måste man ersätta g^n med $ng \, (= g + \ldots + g$ då $n > 0)$. Ordet "potens" ersätter man då med "multipel".

Exempel. (a) Låt $G = \mathbb{C}^*$ vara gruppen av de komplexa talen med avseende på multiplikation. Om $g = i$ så får vi

$$i = 1, \; i^1 = i, \; i^2 = -1, \; i^3 = -1, \; i^4 = 1, \; i^5 = i, \; i^6 = -1, \ldots$$

osv så att vi endast får 4 olika tal. Å andra sidan är $i^{-1} = i^3$ så att varje negativ potens är lika med en positiv $(i^{-1})^n = i^{3n}$. Alltså får man $< i >= \{1, i, -1, -i\}$. Termen cyklisk förklaras delvis av detta exempel – likheten $i^4 = 1$ medför att vi får en cyklisk upprepning av potenserna $i^5 = i, i^6 = i^2, i^7 = i^3, i^8 = 1$ osv.

(b) Låt $G = \mathbb{Z}$ med addition och $g = 1$. Då är $< 1 >$ mängden av alla multipler $n \cdot 1$ (t ex $2 \cdot 1 = 1 + 1, 3 \cdot 1 = 1 + 1 + 1$, $-2 \cdot 1 = -(1 + 1)$ osv). Alltså är $< 1 >= \mathbb{Z}$ så att $\mathbb{Z}$ är en oändlig cyklisk grupp $\square$.

Anmärkning. Termen "cyklisk" kan te sig lite egendomlig om man konstaterar att $\mathbb{Z}$ är en cyklisk grupp. Terminologin är historisk motiverad – från början studerade man enbart ändliga grupper för vilka begreppet "cyklisk" är helt klart (se också nästa proposition). $\square$

Man kan ge en helt allmän beskrivning av cykliska grupper:

(4.12) Proposition. *Låt G vara en grupp och $g \in G$.*

(a) *Om $o(g) = n$ så är $< g >= \{e, g, g^2, \ldots, g^{n-1}\}$ och $g^n = e$ (dvs n är den minsta positiva exponent sådan att $g^n = e$).*

(b) *Om $o(g) = \infty$ så är alla potenser g^n, $n \in \mathbb{Z}$, olika.*

Bevis. Antag att $g^m = e, m > 0$. Då finns det högst m olika potenser av g, nämligen $g^0 = e, g, g^2, \ldots, g^{m-1}$, ty om $N = mq + r$ med $0 \leq r < m$, så är $g^N = g^{mq+r} = (g^m)^q g^r = g^r$. Detta betyder att varje potens av g är lika med en av potenserna $e, g, g^2, \ldots, g^{m-1}$.

(a) $o(g) = n$ betyder att det finns n olika potenser av g. Vi påstår att just $g^0 = e$, g, g^2, $\ldots$, g^{n-1} är olika ty $g^i = g^j, 0 \leq i < j < n$ ger att $g^{j-i} = e$, där $j - i = m < n$. Men likheten $g^m = e$ med $0 < m < n$ är omöjlig (om $g^m = e$ så finns det endast m olika potenser av g). Alltså är $< g >= \{e, g, g^2, \ldots, g^{n-1}\}$. g^n är lika med någon av dessa potenser. Men $g^n = g^i$ där $0 < i < n$ ger $g^{n-i} = e$, dvs $g^m = e$ med $m = n - i < n$. En sådan likhet är utesluten så att $g^n = g^0 = e$.

(b) Om $o(g) = \infty$ så måste alla potenser $g^n, n \in \mathbb{Z}$, vara olika ty $g^i = g^j$ för $i < j$ ger $g^m = e$ där $m = j - i > 0$, vilket är omöjligt (enligt första stycket i beviset). $\square$

Vi avslutar detta kapitel med en mycket allmän konstruktion som möjliggör att definiera nya grupper med hjälp av sådana som man redan känner.

(4.13) Exempel. Låt $G_1, G_2, \ldots, G_n$ vara godtyckliga gruper. Vi definierar en ny grupp $G_1 \times G_2 \times \ldots \times G_n$ vars element är $(g_1, g_2, \ldots, g_n)$, där $g_i \in G_i$ för $i = 1, 2, \ldots, n$. Operationen är definierad på följande sätt:

$$(g_1, g_2, \ldots, g_n)(g_1', g_2', \ldots, g_n') = (g_1 g_1', g_2 g_2', \ldots, g_n g_n')$$

Det är klart att den operationen är associativ (man multiplicerar ju i varje grupp G_i separat). Det neutrala elementet är $e = (e_2, e_2, \ldots, e_n)$ där e_i är det neutrala elementet i G_i. Inversen till $(g_1, g_2, \ldots, g_n)$ är $(g_1^{-1}, g_2^{-1}, \ldots, g_n^{-1})$. Gruppen $G_1 \times G_2 \times \ldots \times G_n$ kallas **direkta produkten** av G_i. Om $G_1 = G_2 = \ldots = G_n = G$ skriver man G^n.

Om tex $G = \mathbb{R}$ är gruppen av de reella talen med addition så är $\mathbb{R}^2 = \{(r_1, r_2) : r_1, r_2 \in \mathbb{R}\}$ med koordinatvis addition. $\mathbb{R}^2$ kan tolkas som gruppen av alla vektorer i planet. På samma sätt är $\mathbb{R}^3$ gruppen av alla vektorer i rymden. Om t ex $G = \{1, -1\}$ med multiplikation så består $G \times G$ av $(1, 1), (1, -1), (-1, 1), (-1, 1)$ med koordinatvis multiplikation. $\square$

ÖVNINGAR

4.1. Vilka av följande talmängder är grupper med avseende på multiplikation av tal ?

(a) $\mathbb{Q}^*$ = alla rationella tal $\neq 0$, (b) $\mathbb{Z} \setminus \{0\}$ = alla heltal $\neq 0$,

(c) $\mathbb{C}^*$ = alla komplexa tal $\neq 0$, (d) $U = \{z \in \mathbb{C} : |z| = 1\}$,

(e) $\mathbb{R}_{>0}$ = positiva reella tal, (f) $G = \{2^m 3^n : m, n \in \mathbb{Z}\}$.

4.2. Visa att följande talmängder är grupper med avseende på multiplikation av tal:

(a) $C_n = \{z \in \mathbb{C} : z^n = 1, n$ ett fixt positivt heltal$\}$ (alla n:te enhetsrötter),

(b) $C_\infty = \{z \in \mathbb{C} : z^n = 1$ för något $n \geq 1\}$.

4.3. Bestäm ordningarna av matriserna:

(a) $A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, (b) $B = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, (c) $C = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

i gruppen av alla (2×2)-matriser med determinant $\neq 0$ med avseende på multiplikation (dvs i $GL_2(\mathbb{R})$).

4.4. Låt G vara en grupp och $a, b \in G$. Visa att

(a) $(a^{-1})^{-1} = a$, (b) $(ab)^{-1} = b^{-1}a^{-1}$.

4.5. Visa att G är en abelsk grupp då och endast då $(ab)^{-1} = a^{-1}b^{-1}$ för $a, b \in G$.

4.6. Visa att G är en abelsk grupp då och endast då $(ab)^2 = a^2b^2$ för $a, b \in G$.

4.7. Visa Proposition (4.10).

4.8. Låt H vara en icke-tom ändlig delmängd till en grupp G sådan at $h, h' \in H$ implicerar $h(h')^{-1} \in H$. Visa att H är en delgrupp till G.

4.9. Visa att om ordningen av en grupp G är jämn så finns det ett element $g \in G$ av ordningen 2.

4.10. Låt $G =< a >= \{e, a, \ldots, a^{n-1}\}$, där $a^n = e$.

(a) Visa att $G =< a^k >$ då och endast då $SGD(k, n) = 1$.

(b) Visa att om $H \subseteq G$ och $o(H) = m$ så är $H =< a^d >$ där $d = \frac{n}{m}$.

Ledning. Visa att d är det minsta positiva heltalet sådant att $a^d \in H$ om $H \neq< e >$.

4.11. Visa att varje delgrupp till en cyklisk grupp är cyklisk.

Ledning. Utnyttja övn. 4.10.

4.12. Låt G vara en grupp och A en icke-tom delmängd till G. Visa att den minsta delgrupp till G som innehåller A är

$$< A >= \{a_1 a_2 \ldots a_n : a_i \in A \quad \text{eller} \quad a_i^{-1} \in A \quad \text{och} \quad n \geq 1\}$$

Anmärkning. Om $G =< A >$ så säger man att A är ett generatorsystem för G. Om $A = \{a\}$ så är $< A >=< a >$ den cykliska gruppen genererad av a.

Kapitel 5

RESTGRUPPER

Grupper av rester vid division med naturliga tal är troligen de första exemplen på grupper som har använts i matematiska sammanhang. De har mycket intressanta tillämpningar både i talteorin och t ex i samband med konstruktioner av både koder och krypteringssystem som kommer att diskuteras i fortsättningen av kursen.

Låt n vara ett positivt heltal. Vi skall beteckna med $[a]_n$ resten av ett heltal a vid division med n. T ex är $[11]_5 = 1, [8]_3 = 2$ osv. Vi har:

$$[a]_n = [b]_n \quad \text{då och endast då} \quad n|a-b$$

(se övning 5.4). Likheten $[a]_n = [b]_n$ skriver man ofta som

$$a \equiv b \pmod{n}.$$

Man säger då att a **och** b **är kongruenta modulo** n. Den beteckningen är mycket vanlig och introducerades av C. F. Gauss. Uttrycket $a \equiv b \pmod{n}$ kallas **kongruens**.

Mängden av alla rester vid division med n kommer att betecknas med $\mathbb{Z}_n$. T ex är $\mathbb{Z}_2 = \{0,1\}$, $\mathbb{Z}_5 = \{0,1,2,3,4\}$ och allmänt $\mathbb{Z}_n = \{0,1,\ldots,n-1\}$. Resterna vid division med n kan adderas och multipliceras på följande sätt:

$$r_1 \underset{n}{\oplus} r_2 = [r_1 + r_2]_n, \qquad r_1 \underset{n}{\odot} r_2 = [r_1 r_2]_n \tag{5.1}$$

Dessa operationer kallas **addition och multiplikation modulo** n. T ex är $2 \underset{5}{\oplus} 1 = [2+1]_5 = 3$, $3 \underset{5}{\oplus} 3 = [3+3]_5 = 1$, $3 \underset{5}{\odot} 3 = [9]_5 = 4$ osv. Ofta utelämnar man "n" i symbolerna "$\underset{n}{\oplus}$" och "$\underset{n}{\odot}$" som förenklas till "$\oplus$" och "$\odot$" eller till "$+$" och "$\cdot$". För addition och multiplikation modulo 2 och 3 har vi

$\oplus$	0	1
0	0	1
1	1	0

$\odot$	0	1
0	0	0
1	0	1

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\odot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Det är klart att $\underset{n}{\oplus}$ och $\underset{n}{\odot}$ är kommutativa operationer. Det är också klart att bägge har neutrala element — för addition 0, för multiplikation 1. Men det är inte lika självklart att dessa operationer är associativa. För addition och multiplikation modulo 10 vet vi det sedan länge. När man adderar tre tal t ex

$$\begin{array}{r} 123 \\ 25 \\ 38 \\ \hline \ldots 6 \end{array}$$

så räknar man ut den sista siffran genom att addera 3+5+8 vilket ger sista siffran 6. Med vår nya addition betyder det att $3 \underset{10}{\oplus} 5 \underset{10}{\oplus} 8 = 6$. Det är just addition modulo 10 och det faktum att vi inte bryr oss om hur parenteserna placeras beror på att vi litar på associativiteten. För att bevisa den helt allmänt behöver vi en viktig egenskap hos $\oplus$ och $\odot$:

(5.2) Lemma. *Låt a, b vara godtyckliga heltal. Då gäller:*

$$\begin{aligned} [a+b]_n &= [a]_n \oplus [b]_n, \\ [ab]_n &= [a]_n \odot [b]_n. \end{aligned}$$

Bevis. Låt

$$a = nq_a + r_a, \;\; 0 \le r_a < n, \quad b = nq_b + r_b, \;\; 0 \le r_b < n$$

och

$$a + b = nq_{a+b} + r_{a+b}, \;\; 0 \le r_{a+b} < n, \quad ab = nq_{ab} + r_{ab}, \;\; 0 \le r_{ab} < n.$$

Vi har:

$$[a]_n \oplus [b]_n = r_a \oplus r_b \;\; = \;\; [r_a + r_b]_n = r_{a+b} = [a+b]_n,$$

ty $r_a + r_b = (a - nq_a) + (b - nq_b) = n(q_{a+b} - q_a - q_b) + r_{a+b}$, dvs r_{a+b} är resten vid division av $r_a + r_b$ med n, och

$$[a]_n \odot [b]_n = r_a \odot r_b \quad = \quad [r_a r_b]_n = r_{ab} = [ab]_n,$$

ty $r_a r_b = (a - nq_a)(b - nq_b) = n(q_{ab} - q_a b - q_b a + nq_a q_b) + r_{ab}$, dvs r_{ab} är resten vid division av $r_a r_b$ med n. □

(5.3) Följdsats. *$\oplus$ och $\odot$ är associativa operationer på $\mathbb{Z}_n$.*

Bevis.

$$\begin{aligned}(r_1 \oplus r_2) \oplus r_3 &= [r_1 + r_2]_n \oplus [r_3]_n = [(r_1 + r_2) + r_3]_n \\ r_1 \oplus (r_2 \oplus r_3) &= [r_1]_n \oplus [r_2 + r_3]_n = [r_1 + (r_2 + r_3)]_n\end{aligned}$$

så att $(r_1 \oplus r_2) \oplus r_3 = r_1 \oplus (r_2 \oplus r_3)$ ty $(r_1 + r_2) + r_3 = r_1 + (r_2 + r_3)$.

Exakt samma argument för $\odot$ som för $\oplus$ ger associativiteten av multiplikation modulo n. □

Nu kan vi konstatera:

(5.4) Proposition. *$(\mathbb{Z}_n, \oplus)$ är en grupp. Den är cyklisk.*

Bevis. Slutenheten följer direkt ur definitionen av $\oplus$ i (5.1). Associativiteten har vi just bevisat. 0 är det neutrala elementet. Inversen till r kallas den motsatta resten och är $n - r$ då $r \neq 0$, ty $r \oplus (n - r) = [n]_n = 0$. Vi har $r = 1 + \cdots + 1$ (r ettor) så att $\mathbb{Z}_n =< 1 >$. □

$(\mathbb{Z}_n, \odot)$ är aldrig en grupp ty resten 0 saknar invers ($r \odot 0 = 0$). Man kan försöka rädda situationen genom att eliminera 0. Men $\mathbb{Z}_n \setminus \{0\}$ behöver inte heller vara en grupp. T ex är $2 \odot 3 = [6]_6 = 0$ i $\mathbb{Z}_6$ så att $\mathbb{Z}_6 \setminus \{0\}$ inte är sluten med avseende på $\odot$. Skälet till att man får 0 är att 2 och 3 har gemensamma delare med 6. För att få en grupp räcker det med att eliminera den situationen. Låt $\mathbb{Z}_n^*$ beteckna alla rester som saknar gemensamma delare $\neq 1$ med n, dvs $r \in \mathbb{Z}_n^*$ då och endast då $SGD(r, n) = 1$. T ex

$$\mathbb{Z}_2^* = \{1\},\ \ \mathbb{Z}_3^* = \{1, 2\},\ \ \mathbb{Z}_4^* = \{1, 3\},\ \ \mathbb{Z}_5^* = \{1, 2, 3, 4\},\ \ \mathbb{Z}_6^* = \{1, 5\}.$$

Nu har vi

(5.5) Proposition. *$(\mathbb{Z}_n^*, \odot)$ är en grupp.*

Bevis. För att bevisa slutenheten betrakta två rester sådana att $SGD(r_1, n) = 1$ och $SGD(r_2, n) = 1$. Då är även $SGD(r_1 r_2, n) = 1$. Motsatsen betyder att det finns ett primtal p sådant att $p|n$ och $p|r_1 r_2$. Då är $p|r_1$ eller $p|r_2$, vilket strider mot vårt antagande att r_1 och r_2 saknar gemensamma delare $\neq 1$ med n. Associativiteten av $\odot$ visade vi i (5.3). Det neutrala elementet är 1. Låt $r \in \mathbb{Z}_n^*$. Som vi vet kan man med t ex Euklides algoritm bestämma två heltal x och y sådana att

$$rx + ny = 1$$

(ty $SGD(r, n) = 1$). Detta betyder att $1 = [rx + ny]_n = [rx]_n = [r]_n \odot [x]_n = r \odot [x]_n$ så att $[x]_n$ är inversen till $r \in \mathbb{Z}_n^*$. □

(5.6) Anmärkning. Det framgår från propositionen att för varje $a \in \mathbb{Z}_n^*$ har ekvationen $ax = 1$ exakt en lösning $x \in \mathbb{Z}_n$. I termer av kongruenser kan man säga att kongruensen $ax \equiv 1 \pmod{n}$ har en lösning då $\mathrm{SGD}(a, n) = 1$. Observera att beviset av (5.5) visar att kongruensen kan lösas med hjälp av Euklides algoritm. □

Exempel. Låt $n = 12$. Då är $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$ och multiplikationstabellen är

$\odot$	1	5	7	11
1	1	5	7	1
5	5	1	11	7
7	7	11	1	5
11	11	7	5	1

□

Ett särskilt viktigt fall får man då $n = p$ är ett primtal. Då är $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$. Här räcker det alltså att utelämna 0 ur $\mathbb{Z}_p$ för att få en grupp med avseende på multiplikation.

(5.7) Definition. Ordningen av $\mathbb{Z}_n^*$ betecknas med $\varphi(n)$. Funktionen $\varphi(n)$ kallas Eulers funktion. Alltså är:

$$\varphi(n) = \text{antalet heltal } k \text{ sådana att } 0 \leq k < n \quad \text{och } SGD(k, n) = 1.$$

□

Exempel. $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(6) = 2$ osv. Om p är ett primtal så är $\varphi(p) = p - 1$ (varför?). □

Här följer några viktiga egenskaper hos Eulers funktion:

(5.8) Proposition. *Eulers funktion har följande egenskaper:*

(a) $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$ *då p är ett primtal och $\alpha \geq 1$,*

(b) $\varphi(ab) = \varphi(a)\varphi(b)$ *då $SGD(a, b) = 1$,*

(c) $\varphi(n) = n(1 - \frac{1}{p_1}) \dots (1 - \frac{1}{p_k})$, *där p_i är alla olika primdelare till n.*

Bevis. (a) är en enkel övning (se övning 5.8). Ett bevis av (b) ger vi senare. (c) följer direkt ur (a) och (b): Låt $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$. Då är

$$\begin{aligned}
\varphi(n) = \varphi(p_1^{\alpha_1} \dots p_k^{\alpha_k}) &= \varphi(p_1^{\alpha_1}) \dots \varphi(p_k^{\alpha_k}) \quad \text{(enligt (b))} \\
&= (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \dots (p_k^{\alpha_k} - p_k^{\alpha_k - 1}) \quad \text{(enligt (a))} \\
&= p_1^{\alpha_1} \dots p_k^{\alpha_k}(1 - \frac{1}{p_1}) \dots (1 - \frac{1}{p_k}) = n(1 - \frac{1}{p_1}) \dots (1 - \frac{1}{p_k}).
\end{aligned}$$

□

Med hjälp av (5.8) kan man räkna ut $\varphi(n)$. T ex $\varphi(1000) = \varphi(2^3 \cdot 5^3) = \varphi(2^3)\varphi(5^3) = 4 \cdot 100 = 400$.

Vi avslutar detta kapitel med en intressant och mycket gammal sats om restaritmetiken som brukar kallas "Kinesiska restsatsen". I det enklaste fallet säger satsen att man alltid kan finna ett heltal som ger givna rester modulo två givna relativt prima heltal.

(5.9) Kinesiska restsatsen. *Låt $n_1, n_2, \dots, n_k$ vara relativt prima positiva heltal och låt $r_1 \in \mathbb{Z}_{n_1}$, $r_2 \in \mathbb{Z}_{n_2}$, ..., $r_k \in \mathbb{Z}_{n_k}$. Då existerar ett heltal x entydigt bestämt modulo $n_1 n_2 \dots n_k$ sådant att*

$$[x]_{n_1} = r_1, \ [x]_{n_2} = r_2, \ \dots, \ [x]_{n_k} = r_k.$$

Bevis. Vi skall visa hur man kan beräkna ett tal x som har den önskade egenskapen och därefter visa att det är entydigt modulo $N = n_1 n_2 \cdots n_k$. Beräkna först x_i så att

$$\frac{N}{n_i} x_i \equiv 1 \pmod{n_i}, \qquad \text{dvs} \qquad \frac{N}{n_i} x_i = 1 \quad \text{i} \quad \mathbb{Z}_{n_i}.$$

Eftersom SGD $(\frac{N}{n_i}, n_i) = 1$ enligt förutsättningen kan man beräkna x_i med hjälp av Euklides algoritm (se (5.6)). Välj nu

$$x = r_1 \frac{N}{n_1} x_1 + r_2 \frac{N}{n_2} x_2 + \dots + r_k \frac{N}{n_k} x_k.$$

Då gäller:

$$[x]_{n_i} = [\sum_{j=1}^{k} r_j \frac{N}{n_j} x_j]_{n_i} = \bigoplus_{j=1}^{k} [r_j]_{n_i} \odot [\frac{N}{n_j} x_j]_{n_i} = [r_i]_{n_i} \odot [\frac{N}{n_i} x_i]_{n_i} = [r_i]_{n_i}$$

ty

$$[\frac{N}{n_j} x_j]_{n_i} = \begin{cases} 1 & \text{om} \quad j = i \\ 0 & \text{om} \quad j \neq i \end{cases}$$

Alltså är $x \equiv r_i \pmod{n_i}$ för $i = 1, 2, \ldots, k$.

Antag nu att x och x' ät två heltal sådana att $[x]_{n_i} = [x']_{n_i} = r_i$ för alla i. Då gäller det att $n_i | x - x'$ för alla i och detta innebär att $n_1 n_2 \cdots n_k | x - x'$ därför att alla n_i är relativt prima. Alltså lämnar x och x' samma rest modulo $N = n_1 n_2 \cdots n_k$. $\square$

(5.10) Anmärkning. Kinesiska restsatsen formuleras ofta med hjälp av kongruenser. Då säger den att för relativt prima positiva heltal $n_1, n_2, \ldots, n_k$ och godtyckliga heltal $r_1, r_2, \ldots, r_k$ existerar ett heltal x så att

$$x \equiv r_1 \pmod{n_1}, \; x \equiv r_2 \pmod{n_2}, \; \ldots \; , x \equiv r_k \pmod{n_k}.$$

Man behöver inte förutsätta att r_i är resten vid division med n_i därför att för varje heltal a gäller ju att $a \equiv [a]_{n_i} \pmod{n_i}$. $\square$

Exempel. Låt oss bestämma ett heltal x som vid division med 3 ger resten 2, med 4 resten 3 och med 5 resten 4 dvs

$$x \equiv 2 \pmod 3, \; x \equiv 3 \pmod 4, \; x \equiv 4 \pmod 5.$$

Låt $N = 3 \cdot 4 \cdot 5 = 60$. Först måste vi bestämma x_1, x_2, x_3 sådana att

$$\frac{60}{3} x_1 = 20 x_1 \equiv 1 \pmod 3, \; \frac{60}{4} x_2 = 15 x_2 \equiv 1 \pmod 4, \; \frac{60}{5} x_3 = 12 x_3 \equiv 1 \pmod 5.$$

Detta betyder att vi måste lösa ekvationerna:

$$2x_1 = 1 \quad \text{i} \quad \mathbb{Z}_3, \quad 3x_2 = 1 \quad \text{i} \quad \mathbb{Z}_4, \; 2x_3 = 1 \quad \text{i} \quad \mathbb{Z}_5.$$

Vi hittar lätt (utan Euklides algoritm) att $x_1 = 2, x_2 = 3, x_3 = 3$. Enligt beviset av (5.9) är

$$x = 2 \cdot \frac{60}{3} \cdot 2 + 3 \cdot \frac{60}{4} \cdot 3 + 4 \cdot \frac{60}{5} \cdot 3 = 359$$

en lösning. Den minsta icke-negativa lösningen är $[359]_{60} = 59$ (lösningen är entydigt bestämd modulo 60 enligt (5.9)). Lägg märke till att $x = 60q + 59$ med ett godtyckligt $q \in \mathbb{Z}$ är en lösning (ty $[x]_{60} = 59$) och att sådana x ger alla lösningar (se övning 5.5). Observera också att en uppmärksam student kunde skrivit en lösning direkt utan att använda Kinesiska restsatsen (hur?). □

Vi skall avsluta detta kapitel med en annan formulering och ett annat bevis av Kinesiska restsatsen eftersom det finns flera tillämpningar som baseras just på den formen av satsen.

(5.11) Sats. *Låt $n_1, n_2, \ldots, n_k$ vara parvis relativt prima positiva heltal (dvs $SGD(n_i, n_j) = 1$ då $i \neq j$). Då är*

$$\mathbb{Z}_{n_1 n_2 \cdots n_k} \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$$

och

$$\mathbb{Z}^*_{n_1 n_2 \cdots n_k} \cong \mathbb{Z}^*_{n_1} \times \mathbb{Z}^*_{n_2} \times \ldots \times \mathbb{Z}^*_{n_k}.$$

Bevis. Låt $N = n_1 n_2 \cdots n_k$. Betrakta funktionen:

$$\theta : \mathbb{Z}_N \longrightarrow \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$$

sådan att $\theta([a]_N) = ([a]_{n_1}, [a]_{n_2}, \ldots, [a]_{n_k})$. Definitionen av denna funktion beror inte på heltalet a som definierar resten: Om $[a]_N = [b]_N$ så är $[a]_{n_1} = [b]_{n_1}$, $[a]_{n_2} = [b]_{n_2}$, …, $[a]_{n_k} = [b]_{n_k}$ ty $N | a - b$ implicerar att $n_1 | a - b$, $n_2 | a - b$, …, $n_k | a - b$. Vi har

$$\theta([a+b]_N) = ([a+b]_{n_1}, [a+b]_{n_2}, \ldots, [a+b]_{n_k}) =$$

$$([a]_{n_1}, [a]_{n_2}, \ldots, [a]_{n_k}) + ([b]_{n_1}, [b]_{n_2}, \ldots, [b]_{n_k}) = \theta([a]_N) + \theta([b]_N)$$

så att θ är en grupphomomorfism. Vi vill visa att θ är en isomorfism. Man kontrollerar lätt att olika rester $[a]_N$ och $[b]_N$ har olika bilder: $[a]_{n_1} = [b]_{n_1}$, $[a]_{n_2} = [b]_{n_2}$, …, $[a]_{n_k} = [b]_{n_k}$ betyder att $n_1 | a-b$, $n_2 | a-b$, …, $n_k | a-b$, vilket ger $N = n_1 n_2 \cdots n_k | a - b$, därför att $n_1, n_2, \ldots, n_k$ är parvis relativt prima. Detta innebär att $[a]_N = [b]_N$. Funktionen θ är alltså en-entydig. Men antalet element i $\mathbb{Z}_N$ är N och antalet element i $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$ är lika stort, vilket innebär att varje element i produkten är bilden av ett element i $\mathbb{Z}_N$. Detta visar att θ är en isomorfism.

Det återstår att visa den andra isomorfismen. Först observerar vi att om a är relativt primt med N så är också a relativt primt med varje faktor n_i av N. Detta visar att θ avbildar $\mathbb{Z}^*_N$

i produkten $\mathbb{Z}^*_{n_1} \times \mathbb{Z}^*_{n_2} \times \ldots \times \mathbb{Z}^*_{n_k}$. Å andra sidan om $([a]_{n_1}, [a]_{n_2}, \ldots, [a]_{n_k}) \in \mathbb{Z}^*_{n_1} \times \mathbb{Z}^*_{n_2} \times \ldots \times \mathbb{Z}^*_{n_k}$, så är a relativt primt med alla n_i och således med $N = n_1 n_2 \cdots n_k$. Detta visar att funktionen θ avbildar en-entydigt $\mathbb{Z}^*_N$ på hela $\mathbb{Z}^*_{n_1} \times \mathbb{Z}^*_{n_2} \times \ldots \times \mathbb{Z}^*_{n_k}$. För att kunna påstå att funktionen θ definierar en isomorfism mellan dessa grupper kontrollerar vi att

$$\theta([ab]_N) = ([ab]_{n_1}, [ab]_{n_2}, \ldots, [ab]_{n_k}) =$$

$$([a]_{n_1}, [a]_{n_2}, \ldots, [a]_{n_k})([b]_{n_1}, [b]_{n_2}, \ldots, [b]_{n_k}) = \theta([a]_N)\theta([b]_N).$$

□

(5.12) Anmärkning. Det är mycket lätt att härleda Kinesiska restsatsen från gruppisomorfismen $\mathbb{Z}_{n_1 n_2 \ldots n_k} \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$. Om $(r_1, r_2, \ldots, r_k) \in \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$ så säger satsen att det finns exakt en rest $[x]_N \in \mathbb{Z}_{n_1 n_2 \ldots n_k}$ sådan att

$$[x]_{n_1} = r_1, \;\; [x]_{n_2} = r_2, \ldots, \;\; [x]_{n_k} = r_k.$$

□

(5.13) Exempel. Gruppen $\mathbb{Z}_{100}$ kan enligt sats (5.11) skrivas som produkt av mindre grupper: $100 = 2^2 5^2$ så att $\mathbb{Z}_{100} \cong \mathbb{Z}_4 \times \mathbb{Z}_{25}$.

□

Nu kan vi bevisa multiplikativiteten av Eulers funktion (se (5.8)(b)):

(5.14) Följdsats. *För Eulers funktion φ gäller att $\varphi(ab) = \varphi(a)\varphi(b)$ då a och b är relativt prima naturliga tal.*

Bevis. Enligt (5.11) är $\mathbb{Z}^*_{ab} \cong \mathbb{Z}^*_a \times \mathbb{Z}^*_b$. Antalet element i vänsterledet är $\varphi(ab)$, medan det i högerledet är $\varphi(a)\varphi(b)$. □

ÖVNINGAR

5.1. Bestäm sista siffran av talet:

a) 2^{1998}, b) $13^{20} + 22^{30}$, c) 7^{7^7}.

5.2. Bestäm resten vid division av

a) 3^{100} med 7, b) 2^{1000} med 3,5,11,13.

5.3. Talen $F_n = 2^{2^n} + 1$, $n = 0, 1, 2, \ldots$, kallas Fermattalen. $F_0 = 3, F_1 = 5, F_2 = 17, F_3 = 257, F_4 = 65537$ är alla primtal. Pierre Fermat (1601-1665) påstod att alla tal F_n är primtal, men 100 år senare visade Leonard Euler (1707-1783) att $641|F_5$. Visa det genom att utnyttja likheterna $5 \cdot 2^7 + 1 = 641$ och $5^4 + 2^4 = 641$. Räkna i $\mathbb{Z}_{641}$.

5.4. Låt $a, b, n \in \mathbb{Z}$ och $n > 0$. Visa att $[a]_n = [b]_n$ då och endast då $n|a - b$.

5.5. Låt a och n vara relativt prima heltal. Låt x_0 vara en lösning till kongruensen $ax \equiv b \pmod{n}$ för ett heltal b. Visa att alla andra lösningar till denna kongruens kan skrivas på formen $x_0 + kn$, där $k = 0, \pm 1, \pm 2, \ldots$.

5.6. Visa att grupperna $\mathbb{Z}_5^*$, $\mathbb{Z}_7^*$ och $\mathbb{Z}_9^*$ är cykliska men $\mathbb{Z}_8^*$ inte är cyklisk.

5.7. Skriv ut grupptabellerna för

a) $\mathbb{Z}_2 \times \mathbb{Z}_3$, b) $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Är dessa grupper cykliska?

5.8. Visa att $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$ då p är ett primtal och $\alpha \geq 1$.
Ledning. Skriv ut alla heltal k sådana att $0 \leq k < p^\alpha$ och $p|k$.

5.9. Skriv följande grupper som produkt av mindre restgrupper

(a) $\mathbb{Z}_{36}$, (b) $\mathbb{Z}_{75}$, (c) $\mathbb{Z}_{15} \times \mathbb{Z}_{28}$, (d) $\mathbb{Z}_{75600}$.

5.10. Lös följande ekvationer:

(a) $17x = 1$ i $\mathbb{Z}_{23}$, (b) $6x = 17$ i $\mathbb{Z}_{41}$, (c) $x^2 = 5$ i $\mathbb{Z}_{29}$.

5.11. Låt $\theta : \mathbb{Z}_{360} \to \mathbb{Z}_8 \times \mathbb{Z}_9 \times \mathbb{Z}_5$ vara definierad som i beviset av (5.11). Bestäm $\theta^{-1}(1, 0, 0)$, $\theta^{-1}(0, 1, 0)$, $\theta^{-1}(0, 0, 1)$. Beräkna därefter $\theta^{-1}(1, 2, 3)$.

Kapitel 6

TRANSFORMATIONSGRUPPER

Även detta kapitel handlar om exempel på grupper. Vi bekantar oss med grupper relaterade till olika geometriska rum och geometriska figurer i dessa rum. En stor del av gruppteorin utvecklades med utgångspunkt från dessa exempel och den slutliga definitionen av gruppbegreppet formulerades först när man upptäckte att grupper är lika vanliga i geometrin som i algebran (se vidare anmärkning (6.8)). Eftersom funktioner mellan olika rum ofta kallas transformationer (eller avbildningar) kallar man grupper bestående av sådana funktioner för transformationsgrupper.

Först måste vi repetera och något komplettera våra kunskaper om funktioner:

(6.1) Definition. Låt $f : X \to Y$ vara en funktion. Man säger att f är **injektiv** (eller entydig) om f avbildar olika element i X på olika element i Y dvs om $x_1 \neq x_2$ ger $f(x_1) \neq f(x_2)$. f kallas **surjektiv** (eller på hela Y) om till varje $y \in Y$ finns $x \in X$ så att $f(x) = y$. En funktion som samtidigt är injektiv och surjektiv kallas **bijektiv**.

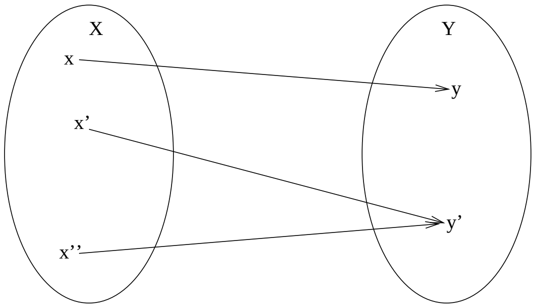
Man säger att två funktioner $f_1 : X \to Y$ och $f_2 : X \to Y$ är lika om $f_1(x) = f_2(x)$ för varje $x \in X$.

Med **sammansättningen** av två funktioner $f : X \to Y$ och $g : Y \to Z$ menar man funktionen $g \circ f : X \to Z$ (“g ring f”) sådan att:

$$(g \circ f)(x) = g(f(x)).$$

Man säger att en funktion $g : Y \to X$ är en **invers** till $f : X \to Y$ om $g \circ f = i_X$ och $f \circ g = i_Y$, där i_X är den **identiska funktionen** på X och i_Y den identiska funktionen på Y dvs $i_X(x) = x$ då $x \in X$ och $i_Y(y) = y$ då $y \in Y$. □

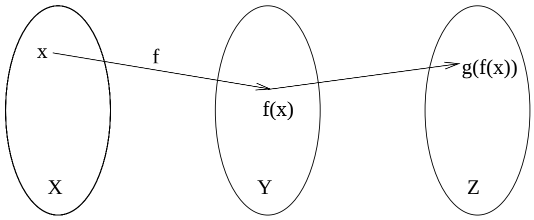
Om man tänker på en funktion f från X till Y som pilar från alla element i X till vissa element i Y (se fig. 6.1) så kan man lätt åskådliggöra alla dessa begrepp. f är injektiv om pilar som startar från olika punkter i X kommer fram till olika punkter i Y, f är surjektiv om



Figur 6.1

till varje punkt i Y kommer en pil, och f är bijektiv om de bägge egenskaperna gäller. Om f är bijektiv så kan man vända på alla pilar från X till Y och då får man inversen g till f (vi visar detta påstående helt formellt i nästa proposition).

Sammansättningen av $g(f(x))$ innebär geometriskt att man först följer pilen från punkten $x \in X$ till punkten $f(x) \in Y$ och därefter pilen från punkten $f(x) \in Y$ till punkten $g(f(x)) \in Z$.



(6.2) Proposition. *$f : X \to Y$ har en invers $g : Y \to X$ då och endast då f är bijektiv. Inversen g är entydigt bestämd (den betecknas f^{-1}).*

Bevis. "$\Rightarrow$" Låt g vara en invers till f dvs $g(f(x)) = x$ då $x \in X$ och $f(g(y)) = y$ då $y \in Y$. Om $x_1 \neq x_2$ så har vi $f(x_1) \neq f(x_2)$ ty likheten $f(x_1) = f(x_2)$ ger $g(f(x_1)) = g(f(x_2))$ dvs $x_1 = x_2$. Alltså är f injektiv. Låt $y \in Y$. Då är $y = f(g(y))$ dvs f avbildar $g(y)$ på y. Detta visar att f är surjektiv. Följaktligen är f bijektiv.

"$\Leftarrow$" Låt f vara bijektiv. Då är varje element $y \in Y$ bilden av exakt ett element $x \in X$. Definiera:

$$g(y) = x \Leftrightarrow f(x) = y.$$

Då har vi: $(g \circ f)(x) = g(f(x)) = g(y) = x$ för $x \in X$ och $(f \circ g)(y) = f(g(y)) = f(x) = y$ för $y \in Y$. Detta visar att g är en invers till f. Slutligen om även g' är en invers till f så har vi

$$f(g(y)) = f(g'(y)) = y$$

då $y \in Y$. Men f är injektiv så att $g(y) = g'(y)$ för varje $y \in Y$ vilket visar att $g = g'$. $\square$

Vi antecknar också följande egenskaper hos funktioner vars bevis lämnar vi som övning.

(6.3) Proposition. *Låt $f : X \to Y$ och $g : Y \to Z$ vara funktioner.*

(a) *Om f och g är injektiva så är $g \circ f$ injektiv.*

(b) *Om f och g är surjektiva så är $g \circ f$ surjektiv.*

(c) *Om f och g är bijektiva så är $g \circ f$ bijektiv.*

Låt nu X vara en mängd och låt $G(X)$ vara mängden av alla bijektiva funktioner (med andra ord: bijektiva transformationer) $f : X \to X$.

(6.4) Proposition. *$(G(X), \circ)$ är en grupp med avseende på sammansättningen av funktioner.*

Bevis. Om $f : X \to X$ och $g : X \to X$ är bijektiva funktioner så är även $g \circ f : X \to X$ en bijektiv funktion enligt (6.3) (c). Alltså är $G(X)$ sluten med avseende på operationen. För att visa associativiteten låt $h : X \to X$ vara en bijektiv funktion. Då är:

$$[(f \circ g) \circ h](x) = (f \circ g)(h(x)) = f(g(h(x)))$$

och

$$[f \circ (g \circ h)](x) = f((g \circ h)(x)) = f(g(h(x)))$$

för $x \in X$. Alltså är $(f \circ g) \circ h = f \circ (g \circ h)$. Det neutrala elementet är den identiska funktionen $i_X(x) = x$ för $x \in X$. Inversen till f är den inversa funktionen f^{-1} som existerar (och är bijektiv) enligt (6.2). □

(6.5) Permutationsgrupper. Låt $X = \{1, 2, \ldots, n\}$. $G(X)$ består av alla bijektiva funktioner $f : X \to X$ dvs $f(1) = p_1, f(2) = p_2, \ldots, f(n) = p_n$, där $p_1, p_2, \ldots, p_n$ är en ordningsföljd av talen $1, 2, \ldots, n$. Sådana funktioner kallas som bekant **permutationer**. Vi kommer att skriva:

$$f = \left(\begin{array}{cccc} 1 & 2 & \ldots & n \\ p_1 & p_2 & \ldots & p_n \end{array} \right).$$

Gruppen $G(X)$ betecknas ofta med S_n och kallas **symmetriska gruppen** av graden n. Låt oss påminna om att $o(S_n) = n!$ (antalet olika permutationer av n element). T ex då $n = 3$ får man gruppen S_3 bestående av $3! = 6$ permutationer

$$I = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 2 & 3 \end{array} \right), f_1 = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 3 & 2 \end{array} \right), f_2 = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 3 & 2 & 1 \end{array} \right), f_3 = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 2 & 1 & 3 \end{array} \right),$$

$$f_4 = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 2 & 3 & 1 \end{array} \right), f_5 = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 3 & 1 & 2 \end{array} \right).$$

Gruppen S_2 har 2 element:

$$I = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, \ f = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}.$$

Permutationerna kan representeras mera kompakt. Låt $p_1, p_2, \ldots, p_k \in \{1, 2, \ldots, n\}$ och låt $(p_1, p_2, \ldots, p_k)$ beteckna funktionen:

$$f(p_1) = p_2, f(p_2) = p_3, \ldots, f(p_k) = p_1.$$

och $f(i) = i$ då $i \neq p_1, p_2, \ldots, p_k$.

Exempel. $(1, 2, 3) \in S_3$ är beteckningen för $\binom{123}{231}$, $(2, 4) \in S_4$ betyder $\binom{1234}{1432}$, $(3, 2, 4) \in S_4$ är $\binom{1234}{1423}$, $(1) \in S_3$ är $\binom{123}{123}$. □

Man säger att permutationen $(p_1, p_2, \ldots, p_k)$ är en **cykel** av längden k. Låt

$$f = (p_1, p_2, \ldots, p_k) \quad \text{och} \quad g = (p'_1, p'_2, \ldots, p'_l),$$

där alla tal $p_1, p_2, \ldots, p_k, p'_1, p'_2, \ldots, p'_l$ är olika. Då säger man att f och g är disjunkta cykler. För sådana cykler har vi $f \circ g = g \circ f$ (kontrollera att $(f \circ g)(x) = (g \circ f)(x)$ för varje $x \in \{1, 2, \ldots, n\}$). Varje permutation är en sammansättning av disjunkta cykler. T ex

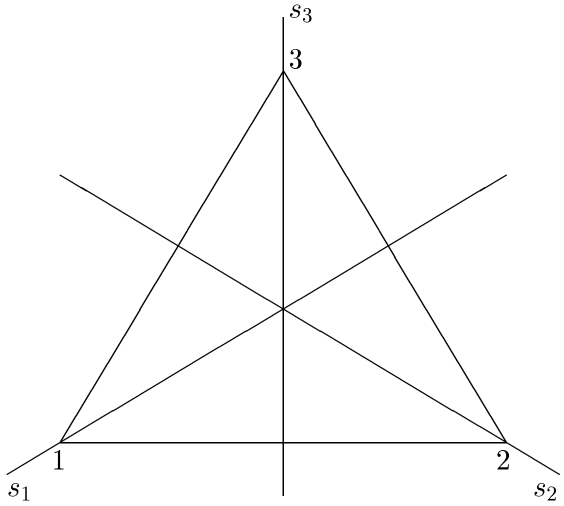
$$\begin{pmatrix} 123456789 \\ 217439568 \end{pmatrix} = (1, 2) \circ (3, 7, 5) \circ (6, 9, 8).$$

Hur får man en sådan framställning? Nedan följer ett enkelt recept:

(6.6) Hur skriver man en permutation som produkt av cykler? Man väljer ett tal p_1 som inte avbildas på sig självt. Därefter tar man bilden p_2 av p_1, bilden p_3 av p_2 osv, tills man får p_1 igen. Då har man en cykel. Nu tar vi ett tal som inte ingår i första cykeln och vi upprepar proceduren. Det gör vi så länge det finns tal som inte ingår i en tidigare bildad cykel och som inte avbildas på sig självt.

Ofta är man intresserad av olika delgrupper till $G(X)$. Man betraktar då bijektiva funktioner $f : X \to X$ med en viss egenskap och visar att funktioner med den egenskapen bildar en delgrupp till $G(X)$. Låt oss betrakta några exempel:

(6.7) Exempel. (a) Låt X vara en liksidig triangel i planet och låt G bestå av alla transformationer av planet som bevarar avståndet och avbildar triangeln på sig själv. G är en grupp med avseende på sammansättningen av avbildningarna (en delgrupp till $G(X)$) och kallas ofta **triangelgruppen** eller, mera exakt, **symmetrigruppen av en liksidig triangel**. Det är inte svårt att beskriva alla element i G. Man kan vrida triangeln $0°, 120°$ och $240°$ kring dess mittpunkt och spegla den i de tre symmetriaxlarna S_1, S_2, S_3. Man får alltså 6 transformationer som ges i form av permutationer av triangelns 3 hörn:



$$I = \begin{pmatrix} 123 \\ 123 \end{pmatrix} = (1);\ v_1 = \begin{pmatrix} 123 \\ 231 \end{pmatrix} = (1,2);\ v_2 = \begin{pmatrix} 123 \\ 312 \end{pmatrix} = (1,3,2)$$

$$s_1 = \begin{pmatrix} 123 \\ 132 \end{pmatrix} = (2,3);\ s_2 = \begin{pmatrix} 123 \\ 321 \end{pmatrix} = (1,3);\ s_3 = \begin{pmatrix} 123 \\ 213 \end{pmatrix} = (1,2).$$

På det sättet får vi alla möjliga avbildningar ty varje avbildning är en permutation av hörnen 1, 2, 3. Men det finns exakt 6 permutationer av talen 1, 2, 3 (de bildar den symmetriska gruppen av graden 3). Lägg märke till att gruppen inte är kommutativ. T ex $v_1 = s_1 \circ s_2 \neq s_2 \circ s_1 = v_2$. Gruppen G har följande grupptabell:

	I	v_1	v_2	s_1	s_2	s_3
I	I	v_1	v_2	s_1	s_2	s_3
v_1	v_1	v_2	I	s_3	s_1	s_2
v_2	v_2	I	v_1	s_2	s_3	s_1
s_1	s_1	s_2	s_3	I	v_1	v_2
s_2	s_2	s_3	s_1	v_2	I	v_1
s_3	s_3	s_1	s_2	v_1	v_2	I

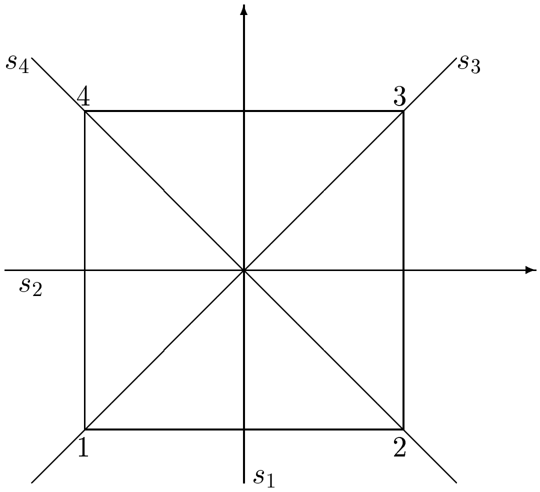
(b) Låt X vara en kvadrat i planet och låt G bestå av alla transformationer av planet som bevarar avståndet och kvadraten. Denna grupp kallas ofta **kvadratgruppen**. G består i detta fall av följande 8 transformationer: 4 vridningar $0°, 90°, 180°, 270°$ kring kvadratens mittpunkt och 4 speglingar i linjerna s_1, s_2, s_3 och s_4. Man kan beskriva dessa avbildningar med hjälp av följande permutationer av kvadratens hörn 1,2,3,4:

$$I = (1),\ v_1 = (1,2,3,4),\ v_2 = (1,3)(2,4),\ v_3 = (1,4,3,2)$$

(de fyra vridningarna) och

$$s_1 = (1,2)(3,4),\ s_2 = (1,4)(2,3),\ s_3 = (2,4),\ s_4 = (1,3)$$

(de fyra speglingarna).



Dessa 8 permutationer bildar en grupp därför att sammansättningen av två transformationer i G ger en transformation i G. Allt detta är relativt enkelt att se direkt men det följer också ur grupptabellen:

	I	v_1	v_2	v_3	s_1	s_2	s_3	s_4
I	I	v_1	v_2	v_3	s_1	s_2	s_3	s_4
v_1	v_1	v_2	v_3	I	s_4	s_3	s_1	s_2
v_2	v_2	v_3	I	v_1	s_2	s_1	s_4	s_3
v_3	v_3	I	v_1	v_2	s_3	s_4	s_2	s_1
s_1	s_1	s_3	s_2	s_4	I	v_2	v_1	v_3
s_2	s_2	s_4	s_1	s_3	v_2	I	v_3	v_1
s_3	s_3	s_2	s_1	s_3	v_3	v_1	I	v_2
s_4	s_4	s_1	s_3	s_2	v_1	v_3	v_2	I

(c) Helt allmänt kan man betrakta en godtycklig figur X i planet eller i rymden. Mängden G av alla transformationer som bevarar avståndet och figuren X är en grupp med avseende på sammansättningen av transformationerna. Denna grupp kallas ofta **symmetrigruppen av** X. Grupper av den typen har en stor betydelse i olika praktiska sammanhang. Bland annat utnyttjas sådana grupper i kristallografin där man klassificerar kristallografiska strukturer beroende på deras transformationsgrupper (dvs alla transformationer i rymden som bevarar avstånden och strukturen – man förutsätter då att kristallen fyller ut hela rymden). □

(6.8) Anmärkning. Från kursen i linjär algebra känner vi ortogonala avbildningar i Euklidiska rum. Om $\mathbb{R}^n$ betraktas med det vanliga avståndsbegreppet, dvs

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

för två vektorer $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$), så säger man att en linjär avbildning $f : \mathbb{R}^n \to \mathbb{R}^n$ är ortogonal (eller isometrisk) om f bevarar avståndet dvs $d(f(\mathbf{x}), f(\mathbf{y})) = d(\mathbf{x}, \mathbf{y})$. Då är $f(\mathbf{x}) = A\mathbf{x}$, där A är en ortogonal matris dvs $A^{-1} = A^t$, där A^t är den transponerade matrisen till A. Man kontrollerar mycket lätt (se övn. 6.6) att alla ortogonala transformationer bildar en grupp. Den Euklidiska geometrin i $\mathbb{R}^n$ är en studie av alla egenskaper hos $\mathbb{R}^n$ som bevaras vid ortogonala transformationer (exempel på sådana egenskaper är avstånden, vinklarna, volymerna osv). Man kan betrakta andra grupper av linjära avbildningar t ex alla icke-singulära avbildningar dvs alla f som ovan där A är en godtycklig matris med nollskild determinant dvs $A \in GL_n(\mathbb{R})$. En studie av alla egenskaper som bevaras vid dessa transformationer är uppgiften för den **affina geometrin** i $\mathbb{R}^n$. År 1872 formulerade den store tyske matematikern Felix Klein en allmän strategi för studier av olika rum. Hans "Erlangenprogrammet" definierar begreppet geometri i ett rum (t ex i $\mathbb{R}^n$) som alla de egenskaper i rummet som bevaras under verkan av en grupp. Kleins idéer hade stor betydelse för utvecklingen inom både matematiken och fysiken. Så småningom ledde dessa ideer till relativitetsteorin som beskriver olika egenskaper hos vektorer i $\mathbb{R}^4$ som bevars under verkan av Lorentzgruppen och dess delgrupper (se övning 6.6(b)). Det är mycket intressant att Felix Klein fick många av sina idéer under en vistelse i Paris hos C. Jordan då denne studerade Galois arbeten. Tack vare Jordan blev Galois idéer kända för den bredda matematiska allmänheten. Även den store norske matematikern Sophus Lie vistades hos Jordan samtidigt med Klein. S. Lie tillämpade gruppteorin på problem i matematisk analys bl a associerade han grupper med differentialekvationer. Teorin för Liegrupper, som samtidigt är grupper och analytiska mångfalder, har mycket stor betydelse både inom matematiken och inom fysiken. T ex har grupperna $O(n), SO(n), U(n), SU(n)$ den karaktären (se vidare övningar 6.6 och 6.7). □

ÖVNINGAR

6.1. Låt $f : X \to Y$ och $g : Y \to X$. Visa att om $g \circ f = i_X$ så är f injektiv och g surjektiv.

6.2. Låt $f : X \to X$ där X är en ändlig mängd. Visa att om f är injektiv eller surjektiv så är den bijektiv.

6.3. Låt G vara mängden av funktionerna

$$f_1(x) = x,\ f_2(x) = -x,\ f_3(x) = \frac{1}{x},\ f_4(x) = -\frac{1}{x},\ x \in \mathbb{R}^*.$$

Visa att G är en grupp m.a.p. sammansättning. Skriv ut grupptabellen.

6.4. Skriv ut grupptabeller för följande grupper:

(a) symmetrigruppen av en rektangel som inte är en kvadrat,

(b) symmetrigruppen av bokstaven **H**.

Anmärkning: Gruppen i (a) kallas ofta **Kleins fyra(gruppen)** och betecknas med V_4.

6.5. Försök beskriva geometriskt alla 24 element i symmetrigruppen av en regelbunden tetraeder.

6.6. Visa att följande $(n \times n)$-reella matriser (= linjära avbildningar av $\mathbb{R}^n$) bildar en grupp med avseende på matrismultiplikation (= sammansättning):

(a) alla ortogonala matriser (dvs alla $(n \times n)$-matriser A sådana att $A^tA = E$),

(b) alla (4×4)-matriser A sådana att $A^tMA = M$, där

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Anmärkning: Villkoret $A^tMA = M$, där M är en godtycklig symmetrisk matris betyder att A bevarar den kvadratiska form som har matrisen M (visas enkelt i kursen Linjär algebra). I (a) handlar det om formen $X_1^2 + X_2^2 + X_3^2$ och villkoret $A^tA = E$ betyder att om man tar en vektor $\mathbf{x}^t = (x_1, x_2, x_3)$ så är $(A\mathbf{x})^tA\mathbf{x} = \mathbf{x}^t\mathbf{x}$ dvs vektorns längd bevaras då den transformeras med hjälp av A (den kvadratiska formen har samma värde för både $\mathbf{x}$ och $A\mathbf{x}$). I (b) är M matrisen för $X_1^2+X_2^2+X_3^2-T^2$ och villkoret $A^tMA = M$ säger att denna kvadratiska form har samma värde för både $\mathbf{x}$ och $A\mathbf{x}$ dvs $(A\mathbf{x})^tMA\mathbf{x} = \mathbf{x}^tM\mathbf{x}$. Gruppen i (b) kallas **Lorentzgruppen** och spelar en mycket viktig roll i relativitetsteorin). Gruppen i (a) kallas **den ortogonala gruppen** och betecknas ofta med $O(n)$. Den delgrupp till $O(n)$ som består av alla matriser med determinanten lika med 1 kallas **den speciella ortogonala gruppen** och betecknas med $SO(n)$. Lorentzgruppen betecknas ofta $O(3, 1)$.

6.7. (a) Visa att alla **unimodulära** $(n \times n)$-matriser A dvs alla $(n \times n)$-matriser med komplexa element och sådana att $A^{-1} = \overline{A}^t$ ($\overline{A}$ betecknar matrisen som man får genom att konjugera alla element i A) bildar en grupp $U(n)$.

(b) Visa att alla **speciella unimodulära** $(n \times n)$-matriser dvs alla matriser A i (a) sådana att $\det A = 1$ bildar en delgrupp till $U(n)$. Denna delgrupp betecknas med $SU(n)$.

(c) Visa att varje matris i $SU(2)$ kan skrivas på formen

$$\begin{bmatrix} z_1 & z_2 \\ -\overline{z}_2 & \overline{z}_1 \end{bmatrix}$$

där z_1 och z_2 är komplexa tal sådana att $|z_1|^2 + |z_2|^2 = 1$.

6.8. Skriv ut de givna permutionerna som produkt av disjunkta cykler:

(a) $\binom{123456789}{214359678}$, (b) $\binom{1234567}{3542176}$.

6.9. (a) Låt $a = (p_1, p_2, \ldots p_k)$ vara en cykel i S_n. Visa att ordningen av a i denna grupp är lika med dess längd dvs $o(a) = k$.

(b) Visa att om en permutation är en produkt av disjunkta cykler så är dess ordning lika med MGM av längderna av dessa cykler.

(Exempel: Låt $f = (1,2,3)(4,5,7,6)(8,9) \in S_9$. Då är $o(f) = 3 \cdot 4 = 12$)

(c) Ge exempel på en abelsk grupp G och $a, b \in G$ sådana att $o(ab) \neq MGM(o(a), o(b))$.

6.10. Bevisa Proposition (6.3).

Kapitel 7

SIDOKLASSER OCH LAGRANGES SATS

Lagranges sats, som visades i gruppteorins begynnelse, säger att ordningen av en delgrupp till en ändlig grupp är en delare till gruppens ordning. I grunden för ett mycket enkelt bevis av satsen ligger en uppdelning av gruppens element i parvis disjunkta delmängder — sidoklasser till delgruppen. Sidoklasserna spelar en mycket viktig roll i hela gruppteorin.

(7.1) Definition. Mängden Hg av alla produkter hg, där g är ett fixt element av G och h är ett godtyckligt element av H kallar man för en **högersidoklass** till H i G. Alltså är

$$Hg = \{hg : h \in H\} \quad (\text{additivt} : H + g = \{h + g : h \in H\}).$$

Man säger att g är en **representant** för Hg. □

(7.2) Exempel. Låt $G = \mathbb{Z}$ (heltalen med addition) och låt $H =< 5 >$ dvs $H = \{0, \pm 5, \pm 10, \ldots\} = \{5k, k = 0, \pm 1, \pm 2, \ldots\}$. Här är

$$H + 1 = \{5k + 1,\ k = 0, \pm 1, \pm 2, \ldots\}$$

mängden av alla heltal som lämnar resten 1 vid division med 5. På liknande sätt är $H + 2 = \{5k + 2;\ k = 0, \pm 1, \pm 2, \ldots\}$ mängden av alla heltal som lämnar resten 2 vid division med 5. Sidoklasserna $H + 0 = H$, $H + 1$, $H + 2$, $H + 3$ och $H + 4$ är olika och består av alla heltal som är delbara med 5 ($H + 0 = H$), lämnar vid division med 5 resten 1 ($H + 1$), 2 ($H + 2$), 3 ($H + 3$) och 4 ($H + 4$). Dessa 5 mängder täcker hela mängden $\mathbb{Z}$ eftersom varje heltal lämnar (exakt) en av dessa 5 rester vid division med 5. Finns det några andra sidoklasser? $H + 5 = \{5k + 5, k = 0, \pm 1, \pm 2, \ldots\} = \{5(k + 1), k = 0, \pm 1, \pm 2, \ldots\} = H$. Vidare är $H + 6 = \{5k + 6, k = 0, \pm 1, \pm 2, \ldots\} = \{5(k + 1) + 1, k = 0, \pm 1, \pm 2, \ldots\} = H + 1$ osv. Det finns faktiskt inte några andra sidoklasser. Detta är inte en tillfällighet utan en konsekvens av några enkla egenskaper hos sidoklasserna. Nu skall vi diskutera dessa egenskaper och därefter återkomma till exempel. □

(7.3) Proposition. (a) $g \in Hg$

dvs. varje element $g \in G$ tillhör en högersidoklass till H.

(b) $g \in Hg_1 \cap Hg_2 \Rightarrow Hg_1 = Hg_2$

dvs två högersidoklasser som har ett gemensamt element är identiska, eller med andra ord, två olika högersidoklasser saknar gemensamma element.

(c) $g' \in Hg \Leftrightarrow Hg' = Hg$

dvs varje element i en högersidoklass kan väljas som dess representant.

(d) $g' \in Hg \Leftrightarrow g'g^{-1} \in H$ (*additivt:* $g' \in H + g \Leftrightarrow g' - g \in H$).

Bevis. (a) $g = eg \in Hg$ ty $e \in H$.

(b) Enligt förutsättningen är $g = h_1g_1 = h_2g_2$ där $h_1, h_2 \in H$. Vi har $x \in Hg_1 \Rightarrow x = hg_1, h \in H \Rightarrow x = h(h_1^{-1}h_2g_2) = (hh_1^{-1}h_2)g_2 \Rightarrow x \in Hg_2$ ty $hh_1^{-1}h_2 \in H$. Detta visar att $Hg_1 \subseteq Hg_2$. På samma sätt får vi $Hg_2 \subseteq Hg_1$. Alltså är $Hg_1 = Hg_2$.

(c) $g' \in Hg \Rightarrow g' \in Hg' \cap Hg$ (ty $g' \in Hg'$) $\Rightarrow Hg' = Hg$ enligt (b).

(d) $g' \in Hg \Leftrightarrow g' = hg$ för något $h \in H \Leftrightarrow g'g^{-1} = h \in H$. □

(7.4) Anmärkning. Egenskaperna (a) och (b) säger att högersidoklasserna Hg ger en partition av G dvs en uppdelning av alla element i G i parvis disjunkta delmängder (se (2.4) (c)). Detta betyder att högersidoklasserna definierar en ekvivalensrelation på G (se definitionen av ekvivalensrelationer (2.3)). Två gruppelement $x, y \in G$ är relaterade till varandra om de tillhör samma högersidoklass, vilket betyder att $x \sim y$ då och endast då det finns $z \in G$ så att $x, y \in Hz$. Enligt (b) ovan betyder det att $Hx = Hy$ dvs $xy^{-1} \in H$ enligt (d) ($Hx = Hy$ ger $x \in Hy$ så att $xy^{-1} \in H$ enligt (d)). Vi skall titta på några ytterligare exempel på partitioner av grupper med hjälp av högersidoklasser. I praktiska sammanhang när man vill beskriva alla element hörande till en högersidoklass Hg utnyttjar man egenskapen (d). □

(7.5) Exempel. (a) Vi fortsätter exempel (7.2). Vi har $n' \in < 5 > +n$ då och endast då $n' - n \in < 5 >$ enligt (7.3) (d), dvs $5|n' - n$. Man kan också uttrycka det som att

$$n' \in < 5 > +n \Leftrightarrow [n']_5 = [n]_5.$$

Detta betyder att sidoklassen $< 5 > +n$ består av alla tal som lämnar resten $[n]_5$ vid division med 5. Men $[n]_5 = 0, 1, 2, 3, 4$ så att sidoklasserna är $< 5 > +0 =< 5 >, < 5 > +1, < 5 > +2, < 5 > +3, < 5 > +4$.

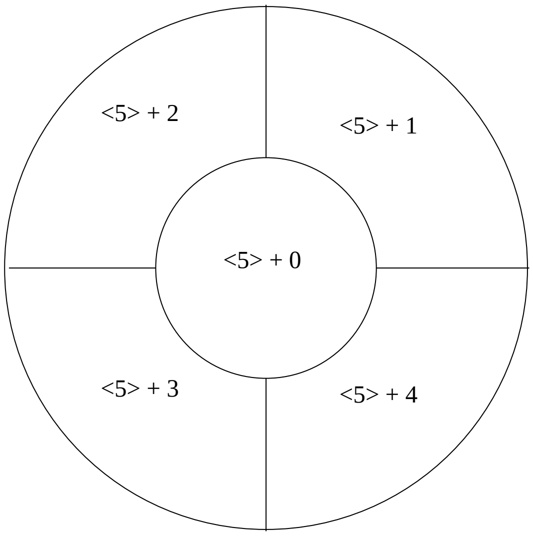


fig. 1

(b) Låt $G = \mathbb{R}^*$ vara gruppen av de reella talen $\neq 0$ och låt $H = \mathbb{R}^*_{>0}$ bestå av positiva reella tal. Då $r \in Hr \Leftrightarrow r'r^{-1} \in H = \mathbb{R}^*_{>0}$ enligt (7.3) (d) dvs $\frac{r'}{r} > 0$. Alltså tillhör r' sidoklassen Hr då och endast då r' har samma tecken som r. Men r kan ha två tecken – plus eller minus. Alltså får vi två sidoklasser – den ena är $H = \mathbb{R}^*_{>0}$ med $+1$ som en representant, den andra $H \cdot (-1) = -\mathbb{R}^*_{>0}$ med -1 som en representant.

(c) Låt $G = \mathbb{R}^2$ vara gruppen av alla vektorer i planet med avseende på addition av vektorer. Låt H vara den delgrupp till G som består av alla vektorer på x-axeln (fig. 2). Om $\mathbf{v}$ är en vektor så består sidoklassen $H + \mathbf{v}$ av alla vektorer som man får genom att addera $\mathbf{v}$ till alla vektorer på x-axeln. Då får man alla vektorer som slutar på den linje som är parallell med x-axeln och som går genom ändpunkten av $\mathbf{v}$. Olika sådana linjer svarar mot olika sidoklasser. Allmänt är $\mathbf{v}' \in H + \mathbf{v} \Leftrightarrow \mathbf{v}' - \mathbf{v} \in H$ dvs $\mathbf{v}' - \mathbf{v}$ är parallell med x-axeln, eller med andra ord, ändpunkten av $\mathbf{v}'$ ligger på den linje som går genom ändpunkten av $\mathbf{v}$ och är parallell med x-axeln. $\square$

(7.6) Anmärkning. Man kan naturligtvis definiera **vänstersidoklasser**

$$gH = \{gh : h \in H\}.$$

Om gruppen är abelsk har vi $gH = Hg$. Då använder vi oftast termen "sidoklass" i stället för "vänstersidoklass" eller "högersidoklass". Alla egenskaper hos högersidoklasser i (7.3) visas analogt för vänstersidoklasser. När gruppen inte är abelsk kan det finnas en distinktion mellan vänster– och högersidoklasser. $\square$

Betrakta nu ett exempel.

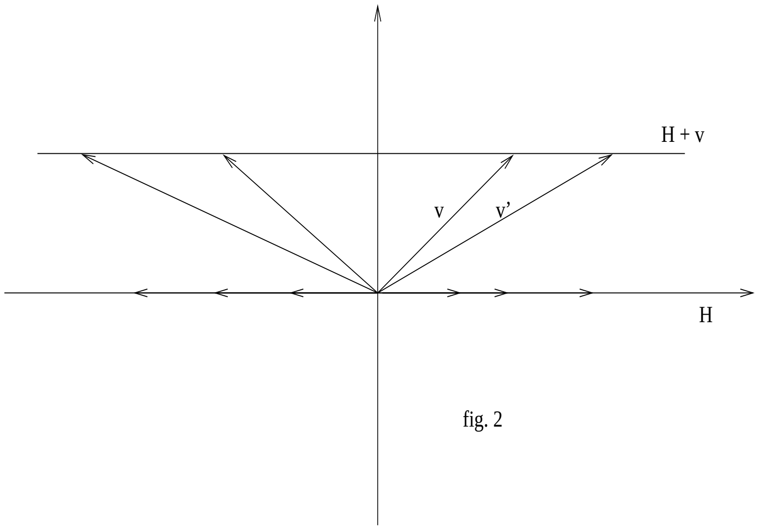


fig. 2

(7.7) Exempel. Låt G vara symmetrigruppen av en liksidig triangel (se exempel (6.7) (a)). Låt $H = \{I, s_1\}$, där $s_1 = (2, 3)$. Här har vi följande vänster- och höger- sidoklasser:

$$\begin{aligned} IH &= s_1H = \{I, s_1\}, \qquad HI = Hs_1 = \{I, s_1\}, \\ v_1H &= s_3H = \{v_1, s_3\}, \qquad Hv_1 = Hs_2 = \{v_1, s_2\}, \\ v_2H &= s_2H = \{v_2, s_2\}, \qquad Hv_2 = Hs_3 = \{v_2, s_3\}. \end{aligned}$$

Vi ser att t ex $s_2H \neq Hs_2$. □

Antalet sidoklasser till H i G är nära relaterat till ordningarna av H och G. Vi har redan sett att antalet element i varje sidoklass är lika med antalet element i H. Detta är ingen tillfällighet:

(7.8) Proposition. *Låt H vara en ändlig grupp. Då är $|Hg| = |H|$ för $g \in G$.*

Bevis. Låt $H = \{h_1, h_2, \ldots, h_m\}$. Då är $Hg = \{h_1g, h_2g, \ldots, h_mg\}$. Alla produkter h_ig är olika ty $h_ig = h_jg$ ger $h_i = h_j$ (multiplicera med g^{-1} från höger!). □

(7.9) Lagranges sats*. *Ordningen av en delgrupp till en ändlig grupp är en delare till gruppens ordning.*

*Joseph Louis Lagrange 1736 - 1813.

Bevis. Låt G vara en ändlig grupp och H en delgrupp till G. Vi vill visa att $o(H)|o(G)$. Vi delar G i högersidoklasserna till H. Sidoklasserna täcker hela gruppen enligt (7.3) (a). Olika sidoklasser saknar gemensamma element enligt (7.3) (b). Antalet element i varje sidoklass är lika med antalet element i H enligt (7.8). Låt i vara antalet högersidoklasser. Då är $o(G) = o(H) \cdot i$ dvs $o(H)$ är en delare till $o(G)$ och kvoten $o(G)/o(H)$ är lika med antalet högersidoklasser. $\square$

(7.10) Följdsats. *Låt G vara en ändlig grupp och H dess delgrupp. Då är antalet högersidoklasser till H i G lika med antalet vänstersidoklasser till H i G. Bägge är lika med $o(G) : o(H)$.*

Bevis. Beviset av Lagranges sats visar att antalet högersidoklasser är lika med $o(G) : o(H)$. När man bevisar Lagranges sats med hjälp av vänstersidoklasser i stället för högersidoklasser (som ovan) får man att $o(G) : o(H)$ är lika med antalet vänstersidoklasser. $\square$

(7.11) Definition. Antalet högersidoklasser (eller vänstersidoklasser) till H i G kallar man för **index** av H i G. Indexet betecknas ofta med $[G : H]$. $\square$

(7.12) Följdsats. *Ordningen av ett element i en ändlig grupp är en delare till gruppens ordning.*

Bevis. Om $g \in G$ så är ordningen $o(g)$ av g lika med ordningen av den delgrupp som g genererar (dvs den delgrupp som består av alla potenser av g). Enligt Lagranges sats är alltså $o(g)$ en delare till $o(G)$. $\square$

(7.13) Följdsats. *Om $o(G) = N$ och $g \in G$ så är $g^N = e$.*

Bevis. Om $o(g) = n$ så är $n|N$ enligt (7.12). Låt $N = n \cdot i$. Då är $g^N = (g^n)^i = e$ ty $g^n = e$ se ((4.12)). $\square$

(7.14) Exempel. Med hjälp av Lagranges sats skall vi beskriva alla delgrupper till kvadratgruppen. $G = \{I, v_1, v_2, v_3, s_1, s_2, s_3, s_4\}$ och grupptabellen finns på sid. 40. Vi har $o(v_1) = o(v_3) = 4$, $o(v_2) = 2$, $o(s_1) = o(s_2) = o(s_3) = o(s_4) = 2$. Om H är en delgrupp till G, så är $o(H) = 1$, 2, 4 eller 8. Det är klart att $o(H) = 1$ ger $H = \{I\}$ och $o(H) = 8$ ger $H = G$ — de två triviala delgrupperna. Om $o(H) = 2$, så måste $H = \{I, g\}$, där g har ordningen 2. Vi vet att det finns 5 sådana g: $g = v_2$ eller s_1 eller s_2 eller s_3 eller s_4. Alltså har vi fem delgrupper av ordningen 2: $\{I, v_2\}$, $\{I, s_1\}$, $\{I, s_2\}$, $\{I, s_3\}$, $\{I, s_4\}$.

Nu antar vi att $o(H) = 4$. Det finns säkert en delgrupp — $H_1 = \{I, v_1, v_2, v_3\}$. Den består av alla vridningar av kvadraten. Låt H vara en delgrupp som innehåller minst en symmetri. H kan inte innehålla v_1 eller v_3 eftersom deras potenser ger alla vridningar (4 stycken). Detta innebär att H måste innehålla två symmetrier. Om H innehåller s_1, så måste den andra vara

s_2, ty $s_1s_2 = v_2$, däremot är $s_1s_3 = v_1$ och $s_1s_4 = v_3$ inte tillåtna. Om H innehåller s_3, så måste den andra vara s_4, ty $s_3s_4 = v_2$, däremot $s_3s_1 = v_3$ och $s_3s_2 = v_1$. Vi får två möjliga delgrupper av ordningen 4: $H_2 = \{I, v_2, s_1, s_2\}$ och $H_3 = \{I, v_2, s_3, s_4\}$. Det finns alltså högst 3 delgrupper av ordningen 4. Vi vet att H_1 är en delgrupp och vi kontrollerar enkelt att H_2 och H_3 också är delgrupper. Det är intressant att tolka dessa grupper geometriskt. H_1 består av alla vridningar av kvadraten. H_2 är symmetrigruppen av en rektangel som inte är en kvadrat (fig. 3 (a)), däremot H_3 är symmetrigruppen av en romb som inte är en kvadrat (fig. 3 (b)). □

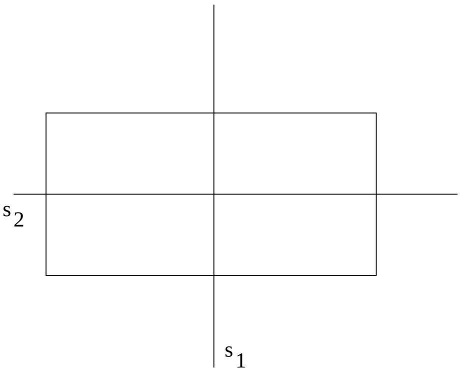


fig. 3 (a)

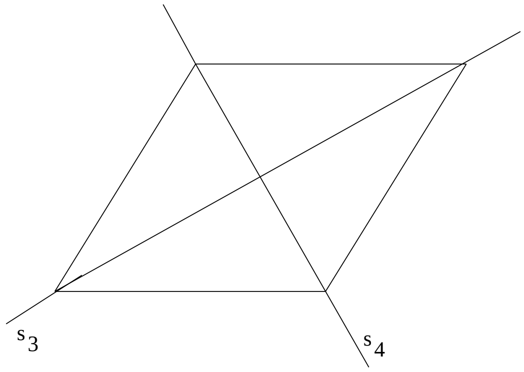


fig. 3 (b)

ÖVNINGAR

7.1. Beskriv alla (höger-)sidoklasser till H i G då

(a) $G = \mathbb{Z}$ (med addition) och $H =< 3 >$,

(b) $G = \mathbb{C}^*$ (de komplexa talen med multiplikation) och $H = \{z \in \mathbb{C}^* : |z| = 1\}$,

(c) $G = \mathbb{C}^*, H = \mathbb{R}^*$ (de reella talen $\neq 0$ med multiplikation),

(d) $G = \mathbb{C}^*, H = \mathbb{R}^*_+$ (de reella positiva talen),

(e) $G = GL_2(\mathbb{R})$ ((2 × 2)-reella matriser med determinant $\neq 0$), $H = SL_2(\mathbb{R}) = \{A \in G : \det A = 1\}$,

(f) $G = \mathbb{Z}_{18}$, $H =< 3 >$.

7.2. Låt $g \in G$ och $o(g) = n$. Visa att om $g^N = e$ så är $n|N$.

7.3. Låt $G = \mathbb{Z}_2^3$. Skriv ut alla sidoklasser till $H = \{000, 111\}$ i G.

7.4. Beskriv alla delgrupper till följande grupper:

(a) symmetrigruppen av en rektangel som inte är en kvadrat,

(b) symmetrigruppen av en liksidig triangel,

(c) $\mathbb{Z}_6$, (d) $\mathbb{Z}_{100}$, (e) $\mathbb{Z}_2 \times \mathbb{Z}_2$, (f) $\mathbb{Z}_2 \times \mathbb{Z}_4$.

Ledning: I (c) och (d) utnyttja övning 4.11 som säger att varje delgrupp till en cyklisk grupp är cyklisk.

7.5. Ge exempel på en delgrupp H till $\mathbb{Q}^*$ (de rationella talen $\neq 0$ med multiplikation) sådan att $H \neq \mathbb{Q}^*$ och index av H i $\mathbb{Q}^*$ är ändligt.

7.6. Visa att en oändlig grupp har oändligt många delgrupper.

7.7. Visa att en grupp G har exakt två delgrupper ($< e >$ och G) om och endast om $o(G)$ är ett primtal.

7.8. Med **exponenten** av en grupp G menas det minsta positiva heltalet m sådant att $g^m = e$ för varje element $g \in G$. Om m inte existerar säger man att gruppens exponent är oändlig.

(a) Visa att varje ändlig grupp har en ändlig exponent.

(b) Ge exempel på en oändlig grupp med en ändlig exponent.

(c) Beräkna exponenten för: $\mathbb{Z}_2, \mathbb{Z}_2 \times \mathbb{Z}_2, \mathbb{Z}_m \times \mathbb{Z}_n$.

(d) Visa att exponenten av en ändlig abelsk grupp är lika med maximalordningen av gruppens element

Ledning: I (d) utnyttja formeln $o(ab) = o(a)o(b)$ då a, b är två element i gruppen vars ordningar är relativt prima (se övning 6.9).

7.9. Skriv ut alla element i gruppen A_4 av alla jämna permutationer av 1,2,3,4. Visa att denna grupp saknar en delgrupp av ordning 6 ($o(A_4) = 12$).

7.10. Visa att en grupp G vars ordning är ett primtal är cyklisk.

Kapitel 8

RINGAR OCH KROPPAR

Grupper är mängder med en operation. Men så viktiga mängder som $\mathbb{Z}$ eller $\mathbb{Z}_n$ har två naturliga operationer – addition och multiplikation. Den situationen är så pass vanlig att det finns en allmän teori av liknande matematiska objekt. De kallas ringar.

(8.1) Definition. Låt R vara en mängd med två binära operationer – addition "+" och multiplikation "·". $(R, +, \cdot)$ kallas **ring** om

(a) $(R, +)$ är en abelsk grupp,

(b) $a(bc) = (ab)c$ då $a, b, c \in R$ dvs multiplikation är associativ,

(c) $a(b + c) = ab + ac$ och $(b + c)a = ba + ca$ då $a, b, c \in R$ dvs multiplikation är distributiv m.a.p. addition. □

(8.2) Anmärkning. Observera att vi oftast skriver ab i stället för $a{\cdot}b$. Det neutrala elementet i gruppen $(R, +)$ brukar betecknas med 0. Vanligen säger man att R är en ring utan att använda beteckningen $(R, +, \cdot)$. □

(8.3) Exempel. (a) $(\mathbb{Z}, +, \cdot), (\mathbb{Q}, +, \cdot), (\mathbb{R}, +, \cdot), (\mathbb{C}, +, \cdot)$ är ringar.

(b) $(\mathbb{Z}_n, \oplus, \odot)$ är en ring. Den enda egenskap som vi inte visade i Kapitel 5 är distributiviteten av $\odot$ m.a.p. $\oplus$. Den visas lätt med hjälp av (5.1) och (5.2):

$$\begin{aligned} a \odot (b \oplus c) &= a \odot [b + c]_n = [a(b + c)]_n = [ab + ac]_n = \\ &= [ab]_n \oplus [ac]_n = [a]_n \odot [b]_n \oplus [a]_n \odot [c]_n \\ &= a \odot b \oplus a \odot c \end{aligned}$$

för $a, b, c \in \mathbb{Z}_n$.

(c) Låt $R = M_n(\mathbb{R})$ mängden av alla $(n \times n)$-reella matriser med matrisaddition och matrismultiplikation. $M_n(\mathbb{R})$ är en ring vilket sammanfattar de viktigaste räknelagarna för matrisaritmetik. Dessa räknelagar visas i alla kurser i linjär algebra (oftast utan att använda termen ring).

(d) Låt $R = C(0,1)$ vara mängden av alla kontinuerliga funktioner på intervallet $(0,1)$ med addition $f+g$ och multiplikation fg av funktioner dvs

$$(f+g)(x) = f(x) + g(x) \quad \text{och} \quad (fg)(x) = f(x)g(x)$$

då $x \in (0,1)$. R är en ring. Man kan naturligtvis ersätta intervallet $(0,1)$ med ett godtyckligt intervall. □

I en ring $(R,+,\cdot)$ har man en blandning av två operationer. Men medan man kräver relativt mycket från den ena – $(R,+)$ skall vara en abelsk grupp, ställer man inte så stora krav på den andra – $(R,\cdot)$ behöver enbart vara en halvgrupp (dvs en mängd med en associativ multiplikation). Ofta betraktar man ringar i vilka $(R,\cdot)$ uppfyller hårdare restriktioner. Här följer några sådana villkor:

(8.4) Definition. Låt $(R,+,\cdot)$ vara en ring.

(a) R är **kommutativ** om $ab = ba$ då $a, b \in R$.

(b) R har en **etta** om det finns ett neutralt element $1 \in R$ m.a.p. multiplikation dvs $1a = a1 = a$ då $a \in R$.

(c) R **saknar nolldelare** om $ab = 0$ ger $a = 0$ eller $b = 0$ då $a, b \in R$ (om $ab = 0$ där $a \neq 0$ och $b \neq 0$ så kallas a och b **nolldelare**).

(d) R är en **kropp** om $(R \setminus \{0\}, \cdot)$ är en abelsk grupp. □

(8.5) Exempel. (a) Alla ringar i exempel (8.3) är kommutativa med undantag av $M_n(\mathbb{R})$ då $n \geq 2$.

(b) Alla ringar i exempel (8.3) har en etta. Ett exempel på en ring utan etta är ringen av de jämna heltalen med vanlig addition och multiplikation.

(c) Alla ringar i exempel (8.3) (a) saknar nolldelare. Men det finns nolldelare i t.ex. $\mathbb{Z}_6$ ty $2 \odot 3 = 0$ (se vidare övning 8.9). Ringen $M_2(\mathbb{R})$ ur (8.3) (c) har nolldelare ty t.ex.

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

(d) $(\mathbb{Q},+,\cdot), (\mathbb{R},+,\cdot), (\mathbb{C},+,\cdot)$ är exempel på kroppar. $(\mathbb{Z},+,\cdot)$ är inte en kropp ty $(\mathbb{Z} \setminus \{0\}, \cdot)$ är inte en grupp. □

Ringarna ur (8.3) (b) är särskilt viktiga:

(8.6) Sats. $(\mathbb{Z}_n, \oplus, \odot)$ *är en kropp då och endast då* n *är ett primtal.*

Bevis. Om $n = p$ är ett primtal så är $\mathbb{Z}_p \setminus \{0\} = \mathbb{Z}_p^*$ en grupp m.a.p. $\odot$ enligt (5.5) dvs $\mathbb{Z}_p$ är en kropp. Om n inte är ett primtal dvs $n = kl$ med $1 < k, l < n$ så är $k \odot l = 0$ i $\mathbb{Z}_n$ dvs $\mathbb{Z}_n \setminus \{0\}$ är inte sluten m.a.p. multiplikation. Detta betyder att $\mathbb{Z}_n \setminus \{0\}$ inte är en grupp och följaktligen $(\mathbb{Z}_n, \oplus, \odot)$ inte är en kropp. $\square$

(8.7) Definition. Man säger att en ring R är ett **integritetsområde** om R är kommutativ, saknar nolldelare och har en etta $1 \neq 0$. $\square$

(8.8) Exempel. (a) Varje kropp K är ett integritetsområde ty $ab = 0$ och $a \neq 0$ ger att $a^{-1}(ab) = b = 0$, där $a, b \in K$, varför K saknar nolldelare.

(b) $\mathbb{Z}_n$ är ett integritetsområde då och endast då n är ett primtal. Detta följer ur (8.6). Om n är ett primtal så är $\mathbb{Z}_n$ en kropp och vi kan hänvisa till (a). Om $n = kl$, $1 < k, l < n$ så har $\mathbb{Z}_n$ nolldelare ty $k \odot l = 0$ trots att $k \neq 0 \neq l$. $\square$

Nu skall vi utvidga vår lista med exempel på ringar med två viktiga ringkonstruktioner:

(8.9) Polynomringar. Låt R vara en kommutativ ring med etta. Med ett polynom med koefficienter i R menar man ett uttryck

$$a_0 + a_1X + \ldots + a_nX^n,$$

där $a_i \in R$. Mängden av alla polynom med koefficienter i R är en ring med avseende på addition:

$$(a_0 + a_1X + a_2X^2 + \ldots) + (b_0 + b_1X + b_2X^2 + \ldots) =$$

$$= (a_0 + b_0) + (a_1 + b_1)X + (a_2 + b_2)X^2 \ldots$$

och multiplikation:

$$(a_0 + a_1X + a_2X^2 + \ldots)(b_0 + b_1X + b_2X^2 + \ldots) =$$

$$= a_0b_0 + (a_0b_1 + a_1b_0)X + (a_0b_2 + a_1b_1 + a_2b_0)X^2 + \ldots$$

Polynomringen av alla polynom med koefficienter i R betecknas med $R[X]$. Det faktum att $R[X]$ är en ring med avseende på addition och multiplikation av polynom kräver naturligtvis en kontroll av alla villkor i definitionen (8.1) men vår erfarenhet av vanliga polynom med t.ex. reella koefficienter (dvs ringen $\mathbb{R}[X]$) borde vara tillräcklig för att kunna acceptera att alla formella villkor i ringdefinitionen verkligen gäller.

Det finns dock en aspekt av definitionen av $R[X]$ som en läsare krävande en större matematisk stringens kan ifrågasätta. Ett polynom definieras som "ett uttryck". Och en sådan formulering kan vara otillfredställande (t.ex. för den som inte ser uttrycket). Vill man undvika den, kan man definiera ett polynom som en oändlig följd:

$$(a_0, a_1, a_2, \ldots, a_n, \ldots)$$

där $a_i \in R$ och $a_i \neq 0$ endast för ett ändligt antal i. Man definierar addition och multiplikation av följderna så att:

$$(a_0, a_1, \ldots, a_n, \ldots) + (b_0, b_1, \ldots, b_n, \ldots) = (a_0 + b_0, a_1 + b_1, \ldots, a_n + b_n, \ldots)$$

och

$$(a_0, a_1, \ldots, a_n, \ldots)(b_0, b_1, \ldots, b_n, \ldots) = (a_0b_0, a_0b_1 + a_1b_0, \ldots, a_0b_n + a_1b_{n-1} + \ldots + a_nb_0, \ldots)$$

Nu kan vi definiera $X = (0, 1, 0, \ldots)$. Då är

$$\begin{aligned} X^2 &= (0,0,1,0,0,\ldots), \\ X^3 &= (0,0,0,1,0,\ldots), \\ X^4 &= (0,0,0,0,1,\ldots), \\ &\vdots \end{aligned}$$

och vi har:

$$(a_0, a_1, a_2, \ldots, a_n, \ldots) =$$
$$(a_0, 0, \ldots) + (a_1, 0, \ldots)X + (a_2, 0, \ldots)X^2 + \ldots + (a_n, 0, \ldots)X^n + \ldots$$

Om vi nu kommer överens om att i stället för $(a, 0, \ldots)$ skriva a (dvs vi identifierar a med "konstantpolynom" $(a, 0, \ldots)$) så har vi vårt tidigare uttryck:

$$(a_0, a_1, a_2, \ldots, a_n, \ldots) = a_0 + a_1X + a_2X^2 + \ldots + a_nX^n + \ldots$$

fast med oändligt många koefficienter a_i (men enbart ett ändligt antal av dessa är $\neq 0$). Det utan tvivel viktigaste exemplet för olika typer av tillämpningar är ringen $\mathbb{Z}_2[X]$ av alla polynom med koefficienter i $\mathbb{Z}_2$. Vi diskuterar polynomringarna närmare i Kap. 9.

(8.10) Produkt av ringar. Låt $R_1, R_2, \ldots, R_k$ vara ringar. Mängden

$$R_1 \times R_2 \times \ldots \times R_k$$

är en ring med avseende på koordinatvis addition och multiplikation dvs

$$\begin{aligned}(r_1, r_2, \ldots, r_k) + (r_1', r_2', \ldots, r_k') &= (r_1 + r_1', r_2 + r_2', \ldots, r_k + r_k'),\\ (r_1, r_2, \ldots, r_k)(r_1', r_2', \ldots, r_k') &= (r_1 r_1', r_2 r_2', \ldots, r_k r_k').\end{aligned}$$

Ringen $R_1 \times R_2 \times \ldots \times R_k$ kallas **produkten** av ringarna $R_1, R_2, \ldots, R_k$. Om $R_1 = R_2 = \ldots = R_k = R$ skriver man oftast R^k.

Exempel. $\mathbb{R}^2$ är ringen av alla reella talpar med koordinatvis addition och multiplikation. $\mathbb{Z}^2$ är ringen av alla heltaliga talpar med samma operationer. □

(8.11) Definition. Man säger att S är en **delring** till R om $S \subseteq R$ och elementen i S bildar en ring med avseende på operationerna i R. □

Exempel. $(\mathbb{Z}, +, \cdot) \subset (\mathbb{Q}, +, \cdot) \subset (\mathbb{R}, +, \cdot) \subset (\mathbb{C}, +, \cdot)$. □

(8.12) Definition. Ett element $r \in R$ kallar man för en **enhet** om r har en multiplikativ invers dvs det finns $r' \in R$ så att $rr' = r'r = 1$. Mängden av alla enheter i R betecknas med R^*. □

(8.13) Sats. *Alla enheter i en kommutativ ring R med etta bildar en (abelsk) grupp med avseende på multiplikation.*

Bevis. Om $r_1, r_2 \in R^*$ så $r_1 r_2 \in R^*$ ty $r_1 r_1' = 1$ och $r_2 r_2' = 1$ ger att $(r_1 r_2)(r_1' r_2') = 1$. Multiplikation är associativ, det neutrala elementet är 1 och definitionsmässigt finns en invers till varje $r \in R$. □

(8.14) Exempel. (a) $\mathbb{Z}$ har enbart två enheter ± 1.

(b) Om K är en kropp så är alla element $a \in K$, $a \neq 0$ enheter ty $(K \setminus \{0\}, \cdot)$ är en grupp. □

(8.15) Sats. *Gruppen av alla enheter i $\mathbb{Z}_n$ är $\mathbb{Z}_n^* = \{k \in \mathbb{Z}_n : SGD(k, n) = 1\}$.*

Bevis. Vi vet redan från (5.4) att varje $k \in \mathbb{Z}_n$ sådant att $SGD(k, n) = 1$ har invers. Antag att $k \in \mathbb{Z}_n$ har invers $k' \in \mathbb{Z}_n$ dvs $k \odot k' = 1$. Alltså är $kk' - 1 = nq$ för ett heltal q. Den sista likheten visar att k och n saknar gemensamma delare $\neq 1$ dvs $SGD(k, n) = 1$. □

ÖVNINGAR

8.1. Vilka av följande talmängder är ringar med avseende på addition och multiplikation av tal? Vilka är kroppar?

(a) $3\mathbb{Z}$,
(b) $\mathbb{Z}[i] = \{a + bi,\ a, b \in \mathbb{Z}\}$,
(c) $\mathbb{Z}[\sqrt{d}] = \{a + b\sqrt{d},\ a, b, d \in \mathbb{Z}\}$,
(d) alla tal $a + b\sqrt{2},\ a, b \in \mathbb{Q}$,
(e) alla tal $a + b\sqrt[3]{2},\ a, b \in \mathbb{Q}$,
(f) alla tal $\frac{a}{b},\ a, b \in \mathbb{Z}, b$ udda.

8.2. Vilka av följande mängder av matriser är ringar med avseende på matrisaddition och matrismultiplikation? Vilka är kroppar?

(a) $\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}, a, b \in \mathbb{R}$,
(b) $\begin{bmatrix} a & b \\ 0 & c \end{bmatrix}, a, b, c \in \mathbb{Z}$,
(c) $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, a, b, c, d \in \mathbb{Z}_2$,
(d) $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}, a, b \in \mathbb{R}$,
(e) $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}, a, b \in \mathbb{Z}_2$,
(f) $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}, a, b \in \mathbb{Z}_3$.

8.3. Låt R vara en ring och X en mängd. Visa att alla funktioner $f : X \to R$ bildar en ring $\mathcal{F}(X, R)$ under operationerna:

$$(f + g)(x) = f(x) + g(x) \quad \text{och} \quad (fg)(x) = f(x)g(x) \quad \text{för} \quad x \in X.$$

Har $\mathcal{F}(X, R)$ en etta? Har $\mathcal{F}(X, R)$ nolldelare?

8.4. Låt $\mathcal{F}(\mathbb{R}, \mathbb{R})$ vara ringen ur 8.3 ($X = \mathbb{R}, R = \mathbb{R}$). Vilka av följande delmängder till $\mathcal{F}(\mathbb{R}, \mathbb{R})$ är delringar?

(a) $\{f \in \mathcal{F}(\mathbb{R}, \mathbb{R}) : f(x) = f(-x)\}$ (jämna funktioner)

(b) $\{f \in \mathcal{F}(\mathbb{R}, \mathbb{R}) : f(-x) = -f(x)\}$ (udda funktioner)

(c) $\{f \in \mathcal{F}(\mathbb{R}, \mathbb{R}) : f$ kontinuerlig$\}$

(d) $\{f \in \mathcal{F}(\mathbb{R}, \mathbb{R}) : f$ deriverbar$\}$

(e) $\{f \in \mathcal{F}(\mathbb{R}, \mathbb{R}) : f(x_0) = 0, x_0$ ett fixt reellt tal$\}$.

8.5. Låt $R \subseteq S$ vara ringar med en gemensam etta och låt $a \in S$. Visa att varje delring till S som innehåller R och a innehåller alla polynomuttryck $a_0 + a_1a + \ldots + a_na^n$ där $a_i \in R$, $n \geq 1$. Den ringen betecknas med $R[a]$. Visa att

(a) $\mathbb{Z}[i] = \{a + bi, a, b \in \mathbb{Z}\}$,
(b) $\mathbb{Z}[\sqrt{2}] = \{a + b\sqrt{2}, a, b \in \mathbb{Z}\}$,
(c) $\mathbb{Q}[i] = \{a + bi, a, b \in \mathbb{Q}\}$,
(d) $\mathbb{Z}[\sqrt[3]{2}] = \{a + b\sqrt[3]{2} + c\sqrt[3]{4}, a, b \in \mathbb{Z}\}$,
(e) $\mathbb{Z}[5] = \mathbb{Z}$,
(f) $\mathbb{Z}[\frac{1}{2}] = \{\frac{a}{2^m}, a, m \in \mathbb{Z}, m \geq 0\}$.

8.6. Låt $K \subseteq L$ vara kroppar och låt $\alpha \in L \setminus K, \alpha^2 \in K$. Visa att $K[\alpha] = \{a + b\alpha, a, b \in K\}$ är en kropp.

8.7. Visa att alla matriser

$$\begin{bmatrix} z_1 & z_2 \\ -\bar{z}_2 & \bar{z}_1 \end{bmatrix}$$

där $z_1, z_2 \in \mathbb{C}$ bildar en ring med avseende på matrisaddition och matrismultiplikation (en delring till ringen $M_2(\mathbb{C})$ av alla 2×2 komplexa matriser). Visa att ringen är icke-kommutativ och att varje element $\neq 0$ har invers.

Anmärkning: En ring med den egenskapen kallas **skevkropp** eller **divisionsring**. Ringen i övningen kallas **kvaternioner** eller mera exakt **Hamiltons kvaternioner**. Hamilton kom på idéen om kvaternioner år 1843 under en promenad längs Royal Canal i Dublin. Till minne av den händelsen finns idag en tavla vid Hamiltons promenadväg på Brougham Bridge där man återfinner huvudregler för kvaternionaritmetiken: $i^2 = j^2 = k^2 = ijk = -1$. Med vår definition är

$$1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad j = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \quad k = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}, \quad .$$

8.8. Visa att i en godtycklig ring R:

(a) $0a = a0 = 0$

(b) $a(-b) = (-a)b = -ab$

(c) $(-a)(-b) = ab$

(d) $-(-a) = a$

8.9. Visa att $\mathbb{Z}_n$ har nolldelare då och endast då n är sammansatt.

8.10. Visa att ett ändligt integritetsområde är en kropp.

Ledning. Låt $R = \{0, a_1, a_2, \ldots, a_n\}$, där $a_1 = 1$ och låt $a \in R$, $a \neq 0$. Betrakta produkterna $aa_1, aa_2, \ldots, aa_n$ och visa att 1 är bland dem.

8.11. Bestäm alla enheter i följande ringar:

(a) $\mathbb{R}[X]$,

(b) $\mathbb{Z}[i]$

(c) $\mathbb{Z}[\sqrt{-d}], d \in \mathbb{Z}, d > 0$.

8.12. (a) Låt R_1 och R_2 vara två kommutativa ringar med etta. Visa att $(R_1 \times R_2)^* = R_1^* \times R_2^*$.

(b) Låt a och b vara två relativt prima positiva heltal. Utnyttja (a) och isomorfismen $\mathbb{Z}_{ab} \cong \mathbb{Z}_a \times \mathbb{Z}_b$ (se (5.9)) för att bevisa att Eulers funktion är multiplikativ dvs $\phi(ab) = \phi(a)\phi(b)$ då $SGD(a, b) = 1$.

8.13. En funktion $E : \mathbb{Z}_n \to \mathbb{Z}_n$ kallas **modulär krypteringsfunktion** om $E(x) = r \odot x \oplus k$ (vi skriver vidare $rx+k$), där $SGD(r, n) = 1$. Om $r = 1$ kallar man E för **Caesarkryptot** (med t.ex. $n = 26$). Visa att E är en bijektion och bestäm inversen D till E.

Anmärkning: Med hjälp av en modulär krypteringsfunktion krypteras klartexten $r_1 r_2 \ldots r_n$ till $E(r_1)E(r_2)\ldots E(r_n)$. Låt $E_i : \mathbb{Z}_n \to \mathbb{Z}_n$ vara modulära krypteringsfunktioner för $i = 1, 2, \ldots, p$. Med ett **periodiskt substitutionskrypto** menar man krypteringsfunktionen $E : \mathbb{Z}_n^N \to \mathbb{Z}_n^N$ sådan att en klartext av längden N:

$$r_1 r_2 \ldots r_p r_{p+1} r_{p+2} \ldots r_{2p} \ldots$$

krypteras till

$$E_1(r_1)E_2(r_2)\dots E_p(r_p)E_1(r_{p+1})E_2(r_{p+2})\dots E_p(r_{2p})\dots$$

Ett specialfall av detta krypto är **Vigenérekryptot**. Då är $E_i(x) = x + k_i$ varvid $k_1k_2\dots k_p \in \mathbb{Z}_n^p$ svarar mot ett "ord" (t.ex. $n = 26$, $(k_1, k_2, k_3, k_4, k_5, k_6) = (0,\ 11,\ 6,\ 4,\ 1,\ 17,\ 0) =$ "ALGEBRA"). **Vernamskryptot** är uppbyggt på liknande sätt men $i = 1,\ 2,\ \dots,\ N$ dvs $(k_1, k_2, \dots, k_n)$ har samma längd som klartexten dvs $r_1r_2\dots r_n$ krypteras till $E_1(r_1)E_2(r_2)\dots E_N(r_n)$. Sekvensen $(k_1, k_2, \dots, k_N)$ är lagrad både hos sändaren och mottagaren och är helt slumpmässigt vald.

Kapitel 9

POLYNOMRINGAR

Låt R vara en kommutativ ring med etta. Som vi redan vet från (8.9) är ett polynom med koefficienter i R ett uttryck

$$a_0 + a_1X + \ldots + a_nX^n,$$

där $a_i \in R$. Mängden av alla polynom med koefficienter i R är en ring med avseende på addition:

$$(a_0 + a_1X + a_2X^2 + \ldots) + (b_0 + b_1X + b_2X^2 + \ldots) =$$

$$= (a_0 + b_0) + (a_1 + b_1)X + (a_2 + b_2)X^2 + \ldots$$

och multiplikation:

$$(a_0 + a_1X + a_2X^2 + \ldots)(b_0 + b_1X + b_2X^2 + \ldots) =$$

$$= a_0b_0 + (a_0b_1 + a_1b_0)X + (a_0b_2 + a_1b_1 + a_2b_0)X^2 + \ldots$$

Polynomringen av alla polynom med koefficienter i R betecknas med $R[X]$. Det faktum att $R[X]$ är en ring med avseende på addition och multiplikation av polynom kräver naturligtvis en kontroll av alla villkor, men vår erfarenhet av vanliga polynom med t.ex. reella koefficienter (dvs ringen $\mathbb{R}[X]$) borde vara tillräcklig för att kunna acceptera att alla formella villkor i ringdefinitionen verkligen gäller.

(9.1) Definition. Låt $f(X) = a_0 + a_1X + \ldots + a_nX^n \in R[X]$ där R är en kommutativ ring med etta. Om $a_n \neq 0$ så säger man att **graden** av $f(X)$ är n. Vi antar att graden av nollpolynomet (dvs $a_0 = a_1 = \ldots = a_n = 0$) är -1. a_n kallas **högsta koefficienten** av $f(X)$. Polynom av graden 0 kallas **konstanta polynom**. $\square$

(9.2) Divisionsalgoritmen. *Låt $f(X), g(X) \in R[X]$, där $g(X)$ är ett polynom vars högsta koefficient är en enhet i R. Då finns det två entydigt bestämda polynom $q(X), r(X) \in R[X]$ sådana att*

$$f(X) = g(X)q(X) + r(X)$$

där $\operatorname{grad} r(X) < \operatorname{grad} g(X)$.

Bevis. Vi bevisar satsen med hjälp av induktion efter graden av $f(X)$. Om graden av $f(X)$ är -1 (dvs $f(X)$ är nollpolynomet) så är $f(X) = g(X) \cdot 0$ dvs $q(X) = 0$ och $r(X) = 0$. Nu antar vi att satsen gäller för alla polynom $f(X)$ vars grad är $< n$, där $n \geq 0$. Låt $f(X) = a_nX^n + \ldots + a_0$, $g(X) = b_mX^m + \ldots + b_0$ där $a_n \neq 0$, och b_m är en enhet. Om $n < m$ så har vi $f(X) = g(X) \cdot 0 + f(X)$ dvs $q(X) = 0$ och $r(X) = f(X)$. Antag att $n \geq m$. Låt

$$f_1(X) = f(X) - \frac{a_n}{b_m} g(X) X^{n-m}$$

Då är grad $f_1(X) <$ grad $f(X)$ så att

$$f_1(X) = g(X)q_1(X) + r(X), \text{ grad } r(X) < \text{grad } g(X)$$

enligt induktionsantagandet. Men då är

$$f(X) = f_1(X) + \frac{a_n}{b_m} g(X) X^{n-m} = g(X)(q_1(X) + \frac{a_n}{b_m} X^{n-m}) + r(X)$$

dvs

$$f(X) = g(X)q(X) + r(X), \quad \operatorname{grad} r(X) < \operatorname{grad} g(X)$$

där $q(X) = q_1(X) + \frac{a_n}{b_m} X^{n-m}$.

Det återstår att visa entydigheten av q och r. Antag att

$$f(X) = g(X)q(X) + r(X) = g(X)q_1(X) + r_1(X)$$

där även $\operatorname{grad} r_1(X) < \operatorname{grad} g(X)$. Då är

$$(*) \qquad g(X)(q(X) - q_1(X)) = r_1(X) - r(X)$$

Men $\operatorname{grad}(r_1(X) - r(X)) < \operatorname{grad} g(X)$, medan likheten $(*)$ säger att om $q(X) - q_1(X) \neq 0$ så är

$$\text{grad}\,(r_1(X) - r(X)) = \text{grad}\,g(X)(q(X) - q_1(X)) \geq \text{grad}\ g(X)$$

(observera att här utnyttjar vi förutsättningen om högsta koefficienten av $g(X)$). Alltså är $q(X) - q_1(X) = 0$ dvs $q(X) = q_1(X)$ och likheten $(*)$ ger att $r_1(X) = r(X)$. □

Exempel. Låt $f(X) = 2X^3 + 3X^2 + X + 1$, $g(X) = X^2 + 2$ i $\mathbb{Z}_4[X]$. Vi har

$$\begin{array}{r|l}
 & 2X \quad +3 \qquad\qquad = \ q(X) \\ \hline
\underline{X^2+2} & 2X^3 + 3X^2 + X + 1 \\
 & -2X^3 \\ \cline{2-2}
 & \qquad 3X^2 + X + 1 \\
 & \quad -3X^2 \qquad -2 \\ \cline{2-2}
 & \qquad\qquad X + 3 \ = \ r(X)
\end{array}$$

(tänk på det att $2 \cdot 2 = 0$ i $\mathbb{Z}_4$). □

(9.3) Definition. Man säger att $g \in R[X]$ är en **delare** till $f \in R[X]$ om $f = gq$, där $q \in R[X]$. Man skriver då $g|f$. □

Från och med nu förutsätter vi att $R = K$ är en kropp.

(9.4) Definition. Om $f, g \in K[X]$ säger man att $d \in K[X]$ är en **största gemensamma delare** till f och g ($SGD(f,g)$) om

(a) $d|f$ och $d|g$,

(b) $d'|f$ och $d'|g$, där $d' \in K[X]$ implicerar att $d'|d$.

Om $f = g = 0$ definierar man $SGD(0,0) = 0$. □

Genom att utnyttja divisionsalgoritmen kan man beräkna $SGD(f,g)$ för godtyckliga polynom $f, g \in K[X]$ med hjälp av Euklides algoritm (precis som för heltalen).

(9.5) Anmärkning. $SGD(f,g)$ är inte entydig. Om $f \neq 0$ eller $g \neq 0$ och d_1, d_2 är två polynom som uppfyller villkoren i (9.4) så är $d_1|d_2$ och $d_2|d_1$. Alltså är $d_2 = d_1q$, där q har grad 0 ty d_1 och d_2 har samma grad ($\text{grad}\, d_1 \geq \text{grad}\, d_2$ och $\text{grad}\, d_2 \geq \text{grad}\, d_1$). Detta betyder att d_1 och d_2 är lika så när som på en konstant. Genom ett lämpligt val av den konstanten kan vi välja en största gemensamma delare med högsta koefficienten 1. Man kallar en sådan **den största gemensamma delaren**. Två polynom vars största **gemensamma delare är 1 kallas relativt prima**. □

Följande sats visas precis på samma sätt som motsvarande sats för heltalen, men absolutbelopp för heltal måste ersättas med grad för polynom (se (1.5)):

(9.6) Sats. *Om $d = SGD(f, g)$, där $f, g \in K[X]$ så existerar $s, t \in K[X]$ så att*

$$d = fs + gt$$

Med hjälp av (9.6) visar man som för heltalen följande egenskap:

(9.7) Sats. *Om $f|h$, $g|h$ och $SGD(f, g) = 1$, där $f, g, h \in K[X]$ så $fg|h$.*

Bevis. Låt $h = fq_f$, $h = gq_g$ och $1 = fs + gt$. Då är $h = hfs + hgt = fgq_gs + fgq_ft = fg(q_gs + q_ft)$ dvs $fg|h$. □

(9.8) Definition. Man säger att $a \in K$ är ett **nollställe** till $f \in K[X]$ om $f(a) = 0$. □

(9.9) Faktorsatsen. (a) *Resten vid division av $f \in K[X]$ med $X - a$, $a \in K$, är lika med $f(a)$;*

(b) *$a \in K$ är ett nollställe till $f \in K[X]$ då och endast då $X - a|f(X)$.*

Bevis. (a) Enligt divisionsalgoritmen är

$$f(X) = (X - a)q(X) + r,$$

där $\operatorname{grad} r < 1$ dvs r är en konstant. Alltså är $f(a) = r$.

(b) $f(a) = 0 \Leftrightarrow r = f(a) = 0$. □

(9.10) Definition. Man säger att $a \in K$ är ett nollställe av multiplicitet m till $f \in K[X]$ om $(X - a)^m|f(X)$ och $(x - a)^{m+1} \nmid f(X)$. □

(9.11) Sats. *Summan av multipliciteterna av alla nollställen till $f \in K[X]$ är högst lika med $\operatorname{grad} f$.*

Bevis. Låt $a_1, \ldots, a_r$ vara nollställen till f och låt $m_1, \ldots, m_r$ vara deras respektive multipliciteter. Detta betyder att

$$(X - a_1)^{m_1}|f(X), \ldots, (X - a_r)^{m_r}|f(X).$$

Men polynomen $(X - a_i)^{m_i}$ är parvis relativt prima så att

$$(X - a_1)^{m_1} \ldots (X - a_r)^{m_r}|f(X)$$

dvs $\operatorname{grad} f \geq m_1 + \ldots + m_r$. □

(9.12) Derivatan av ett polynom. Låt $f(X) = a_0 + a_1X + \ldots + a_nX^n \in K[X]$. Derivatan av $f(X)$ definieras helt formellt som

$$f'(X) = a_1 + 2a_2X + \ldots + na_nX^{n-1}.$$

De vanliga deriveringsreglerna

$$(f+g)' = f' + g',\ (fg)' = f'g + fg'$$

visas genom en direkt kontroll (se övning 9.7).

(9.13) Sats. *$a \in K$ är ett multipelt nollställe till $f \in K[X]$ (dvs a har multiplicitet > 1) då och endast då $f(a) = f'(a) = 0$.*

Bevis. "$\Rightarrow$" Låt $f(X) = (X-a)^2q(X)$ (multipliciteten av a är minst 2). Då är $f'(X) = 2(X-a)q(X) + (X-a)^2q'(X)$ så att $f(a) = f'(a) = 0$.

"$\Leftarrow$" Antag att $f(a) = f'(a) = 0$ och att multipliciteten av a är 1 dvs $f(X) = (X-a)q(X)$ och $q(a) \neq 0$. Då är $f'(X) = q(X) + (X-a)q'(X)$ så att $f'(a) = q(a) \neq 0$ – en motsägelse. $\square$

Mot primtalen i $\mathbb{Z}$ svarar irreducibla polynom i $K[X]$.

(9.14) Definition. Man säger att ett polynom $f \in K[X]$ är **reducibelt** om $f = gh$, där $g, h \in K[X]$ och $\operatorname{grad} g < \operatorname{grad} f$ samt $\operatorname{grad} h < \operatorname{grad} f$. Ett icke-konstant polynom som inte är reducibelt kallas **irreducibelt**. $\square$

(9.15) Exempel. (a) Varje polynom av grad 1 är irreducibelt.

(b) Ett polynom $f \in K[X]$ av grad 2 eller 3 är reducibelt i $K[X]$ då och endast då f har ett nollställe i K dvs det finns $x_0 \in K$ så att $f(x_0) = 0$. I själva verket, om $f(x_0) = 0$ så är $f(X) = (X - x_0)f_1(X)$ där $f_1(X) \in K[X]$ och $\operatorname{grad} f_1(X) \geq 1$ dvs $f(X)$ är reducibelt. Omvänt om $f(X) = g(X)h(X)$ är en faktoruppdelning av $f(X)$ i två icke-konstanta faktorer så måste någon av dessa ha grad 1. Låt $g(X) = b_0 + b_1X \in K[X]$. Då är $x_0 = -b_0/b_1$ ett nollställe till $f(X)$. Till exempel är $f(X) = X^2 + 1 \in \mathbb{Q}[X]$ irreducibelt i $\mathbb{Q}[X]$ ty det saknar nollställen i $\mathbb{Q}[X]$ ($\pm i \notin \mathbb{Q}$). Det är irreducibelt även i $\mathbb{R}[X]$, men i $\mathbb{C}[X]$ är $X^2 + 1 = (X+i)(X-i)$ så att $X^2 + 1$ är reducibelt i den sistnämda polynomringen.

(c) $f(X) = X^2 + X + 1$ är irreducibelt i $\mathbb{Z}_2[X]$ ty $f(0) = 0^2 + 0 + 1 = 1$ och $f(1) = 1^2 + 1 + 1 = 1$ så att polynomet saknar nollställen i $\mathbb{Z}_2$. Vi har att $X^2 + 1 = (X+1)^2$ i $\mathbb{Z}_2[X]$, så att $X^2 + 1$ är reducibelt i $\mathbb{Z}_2[X]$.

(d) Polynomet $f(X) = X^4 + 4$ saknar rationella (även reella) nollställen. Men man får inte påstå att f är irreducibelt i $\mathbb{Q}[X]$. Detta är ett polynom av grad 4 så att (b) inte är användbar här! I själva verket har vi

$$X^4 + 4 = X^4 + 4X^2 + 4 - 4X^2 = (X^2 + 2)^2 - (2X)^2 = (X^2 + 2X + 2)(X^2 - 2X + 2)$$

dvs $X^4 + 4$ är reducibelt i $\mathbb{Q}[X]$. □

(9.16) Sats. *Om $p \in K[X]$ är irreducibelt och $p|fg$, där $f, g \in K[X]$ så $p|f$ eller $p|g$.*

Bevis. Satsen visas på samma sätt som för heltal (se (1.7)). □

(9.17) Sats. *Varje polynom av grad ≥ 1 i $K[X]$ är en produkt av irreducibla polynom. Om*

$$f = p_1 \dots p_k = p'_1 \dots p'_l,$$

där p_i och p'_i är irreducibla polynom så är $k = l$ och vid en lämplig numrering $p'_i = c_i p_i$, där $c_i \in K$.

Bevis. Satsen bevisas på exakt samma sätt som motsvarande sats om primfaktoruppdelning av heltalen dvs aritmetikens fundamentalsats. □

Vi avslutar med några fakta om irreducibla polynom i olika polynomringar:

(9.18) Exempel. (a) **Ringen $\mathbb{C}[X]$.** Irreducibla polynom är endast alla polynom av grad 1. Detta är innehållet i "(polynom)algebrans fundamentalsats". Om $f(X) \in \mathbb{C}[X]$ så är $f(X) = c(X - z_1) \dots (X - z_n)$ där n är polynomets grad och $z_i \in \mathbb{C}$. Satsen visas enklast med hjälp av analytiska funktioner. Den visades för första gången av C. F. Gauss 1799.

(b) **Ringen $\mathbb{R}[X]$.** Irreducibla är alla polynom av grad 1 och alla polynom $c(X^2 + pX + q)$ med $\Delta = p^2 - 4q < 0$ och $c \neq 0$. Detta följer lätt ur (a) och visades i tidigare kurser (nyckeln till beviset är det faktum att om $f(X)$ har reella koefficienter och $f(z) = 0$ så är även $f(\bar{z}) = 0$, där $\bar{z}$ är det konjugerade talet till z.).

(c) **Ringen** $\mathbb{Q}[X]$. Här finns irreducibla polynom av godtyckliga grader. T.ex. är $X^n - 2$ irreducibelt för varje $n \geq 1$. (se övning 9.5).

(d) **Ringen** $\mathbb{Z}_2[X]$**.** Här finns det också irreducibla polynom av godtyckliga grader (vi bevisar inte detta påstående). Antalet polynom av en fixerad grad är ändligt (2^{n+1} polynom av grad n – räkna!). Man kan tabellera irreducibla polynom (det finns mycket omfattande tabeller med tanke på tillämpningarna). Här följer en kort lista över irreducibla polynom av grad ≤ 5.

grad 1: X, $X+1$

grad 2: X^2+X+1

grad 3: X^3+X+1, X^3+X^2+1

grad 4: X^4+X+1, X^4+X^3+1, $X^4+X^3+X^2+X+1$

grad 5: X^5+X^2+1, X^5+X^3+1, $X^5+X^3+X^2+X+1$,
$X^5+X^4+X^2+X+1$, $X^5+X^4+X^3+X+1$, $X^5+X^4+X^3+X^2+1$

Som exempel visar vi att $p(X)=X^4+X+1$ är irreducibelt. $p(X)$ saknar förstagradsfaktorer ty $p(0)=1$ och $p(1)=1$ dvs $p(X)$ saknar nollställen i $\mathbb{Z}_2$. Antag i så fall att $p(X)$ har en faktoruppdelning $p(X)=p_1(X)p_2(X)$ i en produkt av två irreducibla andragradsfaktorer. Då är $p_1(X)=p_2(X)=X^2+X+1$ ty det finns enbart ett irreducibelt polynom av grad 2. Men $(X^2+X+1)^2=X^4+X^2+1\neq X^4+X+1$ så att $p(X)$ måste vara irreducibelt ($p(X)$ kunde ha varit en produkt av p_1 och p_2 med graderna 1, 3 eller 2, 2). $\square$

ÖVNINGAR

9.1. Bestäm kvoten och resten vid division av $f(X)$ med $g(X)$:

(a) $f(X)=X^3+X^2+1, g(X)=X^2+X+1$ i $\mathbb{Z}_2[X]$;

(b) $f(X)=3X^4+2X^2+4, g(X)=2X^2+4X$ i $\mathbb{Z}_5[X]$.

9.2. Bestäm $SGD(f(X),g(X))$ då

(a) $f(X)=X^4+1, g(X)=X^2+1$ i $\mathbb{Z}_2[X]$;

(b) $f(X)=X^9+1, g(X)=X^6+1$ i $\mathbb{Z}_2[X]$;

(c) $f(X)=X^4+2X^3+X^2+4X+2, g(X)=X^2+3X+1$ i $\mathbb{Z}_5[X]$.

Bestäm s,t sådana att $SGD(f,g)=fs+gt$.

9.3. Faktorisera följande polynom i produkt av irreducibla:

(a) X^4+4 i $\mathbb{Q}[X]$, (e) X^3-2 i $\mathbb{Q}[X]$,
(b) X^4+1 i $\mathbb{R}[X]$, (f) X^3+X+1 i $\mathbb{R}[X]$,
(c) X^7-1 i $\mathbb{Z}_2[X]$, (g) X^2+1 i $\mathbb{Z}_3[X]$,
(d) X^4+2 i $\mathbb{Z}_5[X]$, (h) X^4+X+2 i $\mathbb{Z}_3[X]$.

9.4. Visa att om $p\in K[X]$ är irreducibelt och $p|fg$ så $p|f$ eller $p|g$.

Ledning: Bevis som för heltal.

9.5. Låt $f(X)=a_0+a_1X+\ldots+a_nX^n\in\mathbb{Z}[X]$ och låt p vara ett primtal sådant att $p|a_0, p|a_1, \ldots, p|a_{n-1}, p\nmid a_n$ och $p^2\nmid a_0$. Visa att $f(X)$ är irreduciblet i $\mathbb{Z}[X]$.

Ledning: Påståendet kallas **Eisensteins kriterium**. Antag att $f(X)=g(X)h(X)$ i $\mathbb{Z}[X]$ där $\operatorname{grad} g(X)=k<n$ och $\operatorname{grad} h(X)=l<n, g(X), h(X)\in\mathbb{Z}[X]$ och se vad som händer med $f(X), g(X), h(X)$ vid homomorfismen $\theta:\mathbb{Z}[X]\to\mathbb{Z}_p[X]$. I själva verket är

$f(X)$ också irreducibelt i $\mathbb{Q}[X]$. Om ett polynom med heltaliga koefficienter inte kan faktoriseras i produkt av två polynom av lägre grader i $\mathbb{Z}[X]$ så kan det inte heller faktoriseras på detta sätt i $\mathbb{Q}[X]$. Detta påstående kallas Gauss lemma och dess bevis är inte svårt.

9.6. Definiera $-\infty + n = -\infty$ och $\max(-\infty, n) = n$. Visa att

$$\text{grad}(f(X) + g(X)) = \max(\text{grad} f(X), \text{grad } g(X))$$

och

$$\text{grad}(f(X)g(X)) \leq \text{grad } f(X) + \text{grad } g(X)$$

varvid likheten gäller om $a_n b_m \neq 0$ där $f(X) = a_0 + \ldots + a_n X^n$ och $g(X) = b_0 + \ldots + b_m X^m$ är polynom ur $R[X]$.

9.7. Visa att $(fg)' = f'g + fg'$ då $f, g \in K[X]$.

Ledning: Utnyttja den självklara formeln för $(f + g)'$ och börja beviset med $f = aX^m, g = bX^n$.

Kapitel 10

KROPPSUTVIDGNINGAR

Låt K vara en kropp och $p_0(X) \in K[X]$. Polynomet $p_0(X)$ behöver inte ha något nollställe i K, men det visar sig att det alltid finns en kropp $L \supseteq K$ sådan att $p_0(X)$ har ett nollställe i L. Det är till och med möjligt att konstruera en kropp $L \supseteq K$ så att $p_0(X)$ är en produkt av förstagradsfaktorer med koefficienter i L. Vi visar i detta kapitel hur en sådan kropp L (en kroppsutvidgning av K) kan konstrueras då K och $p_0(X) \in K[X]$ är givna. Först definierar vi kvotringen $K[X]/(p_0(X))$ som kommer att ha en stor betydelse i detta och i efterföljande kapitel.

(10.1) Kvotringen $K[X]/(p_0(X))$. Låt $p_0(X) = a_nX^n + a_{n-1}X^{n-1} + \cdots + a_1X + a_0$ vara ett godtyckligt icke-konstant polynom med koefficienter i K. Låt $p(X) \in K[X]$. Vi skall beteckna med $[p(X)]_{p_0}$ resten vid division av $p(X)$ med $p_0(X)$. Observera att

$$[p(X)]_{p_0} = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$$

där $r_i \in K$, därför att polynomet p_0 har graden n. Vi vill definiera addition och multiplikation av resterna precis som vi gjorde det för addition och multiplikation av rester vid division med ett fixt heltal:

$$[p_1(X)]_{p_0} + [p_2(X)]_{p_0} = [p_1(X) + p_2(X)]_{p_0}$$

och

$$[p_1(X)]_{p_0}[p_2(X)]_{p_0} = [p_1(X)p_2(X)]_{p_0}.$$

Man kontrollerar utan svårigheter att resterna vid division med p_0 bildar en ring med avseende på dessa operationer. Man gör det på samma sätt som för addition och multiplikation av rester vid division med heltal i avsnittet om restgrupper.

Observera att addition av resterna sammanfaller med vanlig addition därför att summan av två rester är också en rest (har graden $< n$), medan produkten av två rester kan ha graden $> n$. Då måste man räkna ut resten av denna produkt vid division med p_0. Vi ger exempel snart, men låt oss först notera att konstanta polynom adderas och multipliceras precis som elementen i K:

$$[a] + [b] = [a + b] \qquad \text{och} \qquad [a][b] = [ab].$$

då $a, b \in K$. För att undvika missförstånd, då vi arbetar med rester och ej polynom, låt oss beteckna $[X] = \alpha$. Vi kommer att utelämna p_0 i $[p(X)]_{p_0}$ då detta är klart från texten. I enlighet med våra additions- och multiplikationsregler har vi då:

$$[r_0 + r_1X + \cdots + r_{n-1}X^{n-1}] = [r_0] + [r_1][X] + \cdots + [r_{n-1}][X]^{n-1} = r_0 + r_1\alpha + \cdots + r_{n-1}\alpha^{n-1}.$$

Dessutom har man:

$$0 = [p_0(X)]_{p_0} = [a_nX^n + a_{n-1}X^{n-1} + \cdots + a_1X + a_0] = a_n\alpha^n + a_{n-1}\alpha^{n-1} + \cdots + a_1\alpha + a_0.$$

Låt oss sammanfatta våra observationer:

(10.2) Sats. *Låt $p_0(X) = a_0 + a_1X + \ldots + a_nX^n, a_n \neq 0$. Varje element i kvotringen $K[X]/(p_0(X))$ kan entydigt skrivas på formen $r_0 + r_1\alpha + \ldots + r_{n-1}\alpha^{n-1}$, där $r(X) = r_0 + r_1X + \ldots + r_{n-1}X^{n-1} \in K[X]$ och $\alpha = [X]$ uppfyller ekvationen $a_0 + a_1\alpha + \ldots + a_n\alpha^n = 0$. Ringen $K[X]/(p_0(X))$ innehåller kroppen K och kommer att betecknas med $K[\alpha]$.*

(10.3) Anmärkning. Satsen säger att $K[\alpha]$ är ett vektorrum över K med en bas 1, α, α^2, …, α^{n-1}. □

(10.4) Exempel. (a) Låt $p_0(X) = 1 + X + X^2 \in \mathbb{Z}_2[X]$. Bå består $\mathbb{Z}_2[X]/(p_0(X))$ av resterna $[a + bX]$, $a, b \in \mathbb{Z}_2$ dvs $[0], [1], [X], [1 + X]$. Låt $[X] = \alpha$. Då kan vi skriva ut resterna som: $0, 1, \alpha, 1+\alpha$. Vi har $[p_0(X)]_{p_0} = [1+X+X^2]_{p_0} = 0$ så att $1+\alpha+\alpha^2 = 0$ dvs $\alpha^2 = \alpha+1$. Additions- och multiplikationstabellerna ser ut så här:

$+$	0	1	α	$1+\alpha$
0	0	1	α	$1+\alpha$
1	1	0	$1+\alpha$	α
α	α	$1+\alpha$	0	1
$1+\alpha$	$1+\alpha$	α	1	0

$\cdot$	0	1	α	$1+\alpha$
0	0	0	0	0
1	0	1	α	$1+\alpha$
α	0	α	$1+\alpha$	1
$1+\alpha$	0	$1+\alpha$	1	α

(b) Låt $p_0(X) = X^2 + 1 \in \mathbb{R}[X]$. $\mathbb{R}[X]/(X^2+1)$ består av alla rester $r = a + bX$, $a, b \in \mathbb{R}$. Låt $[X]_{p_0} = \alpha$. Då är $[r] = [a + bX] = a + b\alpha$. Men $[X^2 + 1]_{p_0} = 0$ så att $\alpha^2 + 1 = 0$ dvs $\alpha^2 = -1$. Vi har alltså:

$$\begin{aligned}(a+b\alpha)+(c+d\alpha) &= (a+c)+(b+d)\alpha \\ (a+b\alpha)(c+d\alpha) &= (ac-bd)+(bc+ad)\alpha\end{aligned}$$

dvs resterna adderas och multipliceras som komplexa tal. Med andra ord är $\mathbb{R}[X]/(X^2+1)$ isomorf med $\mathbb{C}$. □

Nu vill vi veta när $K[X]/(p_0)$ är en kropp.

(10.5) Sats. *$K[X]/(p_0)$ är en kropp då och endast då p_0 är irreducibelt i $K[X]$.*

Bevis. "$\Rightarrow$" Låt p_0 vara irreducibelt och låt $r \in K[X]/(p_0)$, $r \neq 0$, $\operatorname{grad} r < \operatorname{grad} p_0$ och p_0 är irreducibelt. Alltså finns det två polynom $s, t \in K[X]$ så att

$$rs + p_0 t = 1$$

Nu är $[rs + p_0 t]_{p_0} = [r][s] + [p_0][t] = 1$ dvs $[r][s] = 1$ ty $[p_0] = 0$. Alltså är $[s]$ inversen till $[r]$. Detta visar att $K[X]/(p_0)$ är en kropp ty varje $[r] \neq 0$ har invers.

"$\Leftarrow$" Antag att p_0 är reducibelt. Då är $p_0 = r_1 r_2$, där $r_1, r_2 \in K[X]$ grad $r_1 <$ grad p_0 och $\operatorname{grad} r_2 < \operatorname{grad} p_0$. Alltså är $0 = [p_0]_{p_0} = [r_1][r_2]$, vilket betyder att ringen $K[X]/(p_0)$ har nolldelare ty $[r_1] \neq 0$ och $[r_2] \neq 0$. I så fall är $K[X]/(p_0)$ inte en kropp ty kroppar saknar nolldelare*. □

(10.6) Exempel. Både $\mathbb{Z}_2[X]/(X^2+X+1)$ (se (10.4)(a)) och $\mathbb{R}[X]/(X^2+1)$ (se (10.4)(b)) är kroppar. □

Nu kan vi visa att varje polynom med koefficienter i en kropp kan uppdelas i förstagradsfaktorer i en lämplig utvidgning av denna kropp. Vi gör det i två steg.

(10.7) Lemma. *Låt $p_0 \in K[X]$ vara ett irreducibelt polynom. Då existerar en kropp $L \supseteq K$ sådan att p_0 har ett nollställe i L.*

Bevis. Låt $L = K[X]/(p_0)$. Vi vet att L är en kropp som innehåller K. Låt $p_0(X) = a_0 + a_1 X + \ldots + a_n X^n$ och låt $[X]_{p_0} = \alpha$. Då är

$$0 = [p_0]_{p_0} = [a_0 + a_1 X + \ldots + a_n X^n] = a_0 + a_1[X] + \cdots + a_n[X]^n = a_0 + a_1\alpha + \ldots + a_n\alpha^n$$

så att $p_0(\alpha) = 0$. □

*Om L är en kropp och $ab = 0$ för $a, b \in L$ med $a \neq 0$ så är $a^{-1}ab = b = 0$

(10.8) Sats. *Låt $p \in K[X]$ och $\operatorname{grad} p \geq 1$. Då existerar en kropp $L \supseteq K$ sådan att p är en produkt av förstagradsfaktorer i $L[X]$ dvs $p(X) = a(X - \alpha_1) \cdots (X - \alpha_n)$ där $\alpha_i \in L$ och $n = \operatorname{grad} p$.*

Bevis. Vi visar satsen med hjälp av induktion. Om K är en godtycklig kropp och $\operatorname{grad} p = 1$ så är beviset klart. Antag att satsen gäller för alla kroppar och alla polynom av grad $< n$. Låt $\operatorname{grad} p = n$ och låt p_0 vara en irreducibel faktor av p. Enligt (10.7) finns en kropp $L_0 \supseteq K$ sådan att p_0, och följaktligen p, har ett nollställe $\alpha \in L_0$ dvs $p(X) = (X - \alpha)q(X)$, där $q(X) \in L_0[X]$. Då är $\operatorname{grad} q < \operatorname{grad} p$ så att det finns en kropp $L \supseteq L_0 \supseteq K$ sådan att $q(X)$ är en produkt av förstagradsfaktorer med koefficienter i L. Men då är även $p(X)$ en sådan produkt ty $p(X) = (X - \alpha)q(X)$. $\square$

(10.9) Anmärkning. Satsen visades för första gången av L. Kronecker. Den räcker gott och väl för våra syften, men den är inte helt tillfredsställande om man t ex tänker på de komplexa talen: **Varje** icke-konstant polynom med koefficienter i en delkropp K till $\mathbb{C}$ kan skrivas som produkt av förstagradspolynom med komplexa koefficienter. För varje kropp K finns en liknande utvidgning $\bar{K}$ sådan att varje polynom med koefficienter i K sönderfaller i produkt av förstagradsfaktorer med koefficienter i $\bar{K}$. Dessutom kan man hitta $\bar{K}$ så att ingen av dess äkta delkroppar som innehåller K har samma egenskap som $\bar{K}$. $\bar{K}$ kallas **algebraiska höljet** till K. $\bar{K}$ är till och med entydigt bestämd så att om $\bar{K}'$ är en annan kropp med samma egenskaper som $\bar{K}$ så är $\bar{K}'$ isomorf med $\bar{K}$ (man kan välja en isomorfism mellan dessa kroppar så att elementen i K avbildas på sig självt). $\square$

Vi skall avsluta detta avsnitt med en enkel följdsats till satserna (10.2) och (10.5).

(10.10) Följdsats. *Om K är en ändlig kropp med q element och $p_0(X) \in K[X]$ är ett irreducibelt polynom av grad n så är kvotringen $L = K[X]/(p_0(X))$ en kropp med q^n element.*

Bevis. Enligt (10.2) kan varje element i L skrivas entydigt på formen $r_0 + r_1\alpha + \ldots + r_{n-1}\alpha^{n-1}$, där $r_i \in K$ (och $\alpha = [X]_{p_0}$). Eftersom varje r_i antar q olika värden är antalet element i L lika med q^n. Det följer ur (10.5) att L är en kropp. $\square$

Exempel. För att konstruera en kropp med 4 element måste man välja ett irreducibelt polynom av grad 2 över $\mathbb{Z}_2$. Som vi vet är $X^2 + X + 1$ ett sådant polynom och således är $L = \mathbb{Z}_2[X]/(X^2 + X + 1)$ en kropp med 4 element (se (10.4) (a)). $\square$

ÖVNINGAR

10.1. Skriv ut additions- och multiplikationstabellerna för följande ringar:

(a) $\mathbb{Z}_2[X]/(X^2)$, (b) $\mathbb{Z}_2[X]/(X^2 + X)$, (c) $\mathbb{Z}_3[X]/(X^2 + 1)$,

(d) $\mathbb{Z}_2[X]/(X^3 + X + 1)$, (e) $\mathbb{Z}_2[X]/(X^4 + X + 1)$.

10.2. Vilka av följande ringar är kroppar:

(a) $\mathbb{Z}_3[X]/(X^2+2)$, (b) $\mathbb{Z}_5[X]/(X^2+2)$.

10.3. Konstruera en kropp med

(a) 8, (b) 1024, (c) 25, (d) 3125

element.

10.4. Motivera att kvotringen $\mathbb{Z}_2[X]/(X^3+X+1)$ är en kropp och för varje nollskilt element i denna kropp bestäm dess invers.

10.5. Motivera att kvotringen $\mathbb{Z}_2[X]/(X^5+X^2+1) = \mathbb{Z}_2[\alpha]$, där $\alpha = [X]$, är en kropp och bestäm i denna kropp ett element vars potenser genererar den multiplikativa gruppen.

10.6. Bestäm alla lösningar till ekvationen $p_0(X) = 0$ i K då

(a) $p_0(X) = X^2 + X + 1$, $K = \mathbb{Z}_2[X]/(X^2+X+1) = \mathbb{Z}_2[\alpha]$, där $\alpha = [X]$,

(b) $p_0(X) = X^3 + X + 1$, $K = \mathbb{Z}_2[X]/(X^3+X^2+1) = \mathbb{Z}_2[\alpha]$, där $\alpha = [X]$.

10.7. Låt $p_0(X) \in K[X]$ vara ett polynom av grad n. Motivera att om kroppen K har q element så har $K[X]/(p_0(X))$ q^n element.

Kapitel 11

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

$$\begin{aligned} A &\longmapsto 000, \\ B &\longmapsto 111. \end{aligned} \tag{11.1}$$

Situationen har förbättrats avsevärt. Om det inträffar högst ett fel i A eller B så får man någon av 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.

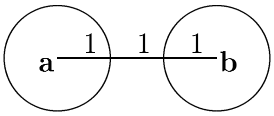
Detta är det enklaste exemplet på en felkorrigerande kod. Rent allmänt kan man beskriva situationen på följande sätt: Man har en mängd av meddelanden X och en metod att översätta dessa meddelanden till sekvenser av 0 och 1. Låt C vara mängden av alla kodord. C innehåller sekvenser av 0 och 1. Man brukar 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 $abc\ldots$ i stället för $(a, b, c, \ldots)$ 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:

(11.2) 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 (11.1) 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 denna egenskap 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 får 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 $\mathbb{Z}_2^n$:

(11.3) 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 $\mathbf{a}$ och $\mathbf{b}$. 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 (11.1) är en så kallad **repetitionskod**, dvs man upprepar varje meddelande ett antal gånger (här 3 gånger). Den metoden är välkänd (och beprövad av varje lärare), men den är tidskrävande och dyrbar. Hammings 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 denna egenskap, 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:

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} \qquad \begin{array}{c|cc} * & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

Därmed 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 associativitet och kommutativitet för både addition och multiplikation, distributivitet för multiplikation med avseende för addition osv. Låt oss lösa ekvationssystemet! Vi får lätt att

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

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

$$\begin{array}{cccccc} x_1 & x_2 & & x_3 & x_4 & x_5 \\ 0 & 0 & & 0 & 0 & 0 \\ 0 & 1 & & 1 & 0 & 1 \\ 1 & 0 & & 1 & 1 & 0 \\ 1 & 1 & & 0 & 1 & 1 \end{array} .$$

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

$$\begin{array}{ccccccccccc} & & x_1 & x_2 & \longmapsto & x_1 & x_2 & x_3 & x_4 & x_5 \\ A & = & 0 & 0 & \longmapsto & 0 & 0 & 0 & 0 & 0 \\ B & = & 0 & 1 & \longmapsto & 0 & 1 & 1 & 0 & 1 \\ C & = & 1 & 0 & \longmapsto & 1 & 0 & 1 & 1 & 0 \\ D & = & 1 & 1 & \longmapsto & 1 & 1 & 0 & 1 & 1 \end{array} \tag{11.4}$$

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:

$$\begin{array}{ccccccc} A & = & 00 & \longmapsto & 00 & 00 & 00 \\ B & = & 01 & \longmapsto & 01 & 01 & 01 \\ C & = & 10 & \longmapsto & 10 & 10 & 10 \\ D & = & 11 & \longmapsto & 11 & 11 & 11 \end{array}$$

Kodorden har alltså längden 6. Kodorden i koden (11.4) 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:

(11.5) 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 högerled 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:

(11.6) 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** för den kod som består 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 (11.2)).

Omvändningen av den sista satsen är också sann (och inte svårt att bevisa):

(11.7) 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.*

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

(11.8) 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 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 (11.4) 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 nedtecknar vi några enkla samband mellan avståndet och vikten:

(11.9) 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 $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$. Påståendet (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 $b_i \neq 0$, vilket visar (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:

(11.10) 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

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

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 **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:

$$\begin{array}{cccccccccc} 0 & 0 & \ldots & 1 & \ldots & 0 & \ldots & 1 & \ldots & 0 \end{array}$$
$$\begin{bmatrix} * & * & \ldots & a & \ldots & * & \ldots & a' & \ldots & * \\ * & * & \ldots & b & \ldots & * & \ldots & b' & \ldots & * \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ * & * & \ldots & x & \ldots & * & \ldots & x' & \ldots & * \end{bmatrix}$$

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.

(11.11) Sats. *En binär matris* **H** *definierar en kod vars vikt är minst* 3 *då och endast då alla kolonner i* **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:

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

Man får lätt

$$\begin{array}{ccccccc} x_1 & = & x_3 & + & x_5 & + & x_7, \\ x_2 & = & x_3 & + & x_6 & + & x_7, \\ x_4 & = & x_5 & + & x_6 & + & x_7. \end{array} \tag{11.12}$$

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:

$$
\begin{array}{ccccccccccccc}
x_3 & x_5 & x_6 & x_7 & \longmapsto & x_3 & x_5 & x_6 & x_7 & x_1 & x_2 & x_4 \\
0 & 0 & 0 & 0 & \longmapsto & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \longmapsto & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & \longmapsto & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & \longmapsto & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & \longmapsto & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & \longmapsto & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & \longmapsto & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & \longmapsto & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & \longmapsto & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & \longmapsto & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & \longmapsto & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & \longmapsto & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & \longmapsto & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & \longmapsto & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & \longmapsto & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & \longmapsto & 1 & 1 & 1 & 1 & 1 & 1 & 1.
\end{array}
\tag{11.13}
$$

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 (11.13). 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 (11.13) 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 (11.12). 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 välkänd 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 Hammings 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.

(11.14) Definition. Man säger att en kod C **detekterar** t fel om ett mottaget ord inte ses 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 sändats. □

(11.15) 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 et 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 struktur menar vi att summan av två kodord återigen är ett kodord) är betydligt svårare att hantera.

Lika enkelt som i sats (11.11) kan vi konstatera att konstruktion av en linjär kod som rättar t fel innebär konstruktion av en matris $\mathbf{H}$ med följande egenskap:

(11.16) 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 (11.11) 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 – Hammingmatriserna 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.

(11.17) 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 $\mathbf{G}\boldsymbol{a} + \mathbf{G}\boldsymbol{b} = \mathbf{G}(\boldsymbol{a}+\boldsymbol{b}) \in C$. □

(11.18) 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:

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

(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

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

□

(11.19) Anmärkning. Man kan utan större problem visa att varje gruppkod har en generatormatris. Se vidare övning 11.9 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

$$\mathbf{G} = \left[\begin{array}{cccc} \mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_k \end{array}\right] = \left[\begin{array}{cccc} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nk} \end{array}\right],$$

där $\mathbf{g}_i = (g_{1i}, g_{2i}, \ldots, g_{ni})^t$ så är $\mathbf{G}\boldsymbol{a} = a_1\mathbf{g}_1 + a_2\mathbf{g}_2 + \ldots + a_k\mathbf{g}_k$ då $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$. 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 $\mathbf{G}\boldsymbol{a}$ är olika då $\boldsymbol{a} \in \mathbb{Z}_2^k$. 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 eftersom kodorden $\mathbf{G}\boldsymbol{a}$ då har informationssymbolerna på de första k platserna och kontrollsymbolerna (checksymbolerna) på de sista $n - k$.

□

I praktiska sammanhang är det oftast mycket viktigt att relativt lätt kunna 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 kan vi genomföra avkodning på ett mycket enkelt sätt:

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

ty $\hat{\boldsymbol{a}} + \boldsymbol{\varepsilon} = \boldsymbol{a} + \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} = \boldsymbol{a}$ (tänk på 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.

(11.21) 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:

(11.22) 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 tabellera 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 (11.20).

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 $2t$. □

(11.23) 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 $\mathbf{H}_3\boldsymbol{\varepsilon}$:

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

Nu observerar vi att felsyndromen helt enkelt är kolonnerna i matrisen $\mathbf{H}_3$ – felet i 1:a koordinatan 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.

(11.24) 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 $|C| = 2^k$). □

(11.25) Anmärkning. Talen

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

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 fler meddelanden kan den överföra. Men det finns en klar motsättning mellan dessa strävanden. En ganska intensiv matematisk forskning är inriktad på 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 som den ryske matematikern 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 redan ursprungligen som en matematisk teori med syfte att lösa mycket konkreta tekniska problem. Under de år som har gått sedan 1948 har flera tusen väsentliga bidrag till den algebraiska kodningsteorin publicerats, 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, särskilt i samband med utvecklingen av datatekniken.

ÖVNINGAR

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

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

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

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

11.3. 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?

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

11.5. 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

$$\boldsymbol{ab} = (a_1b_1, a_2b_2, \ldots, a_nb_n).$$

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

11.6. 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 11.3. 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.

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

11.8. 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.

11.9. (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.

11.10. Bestäm generatormatriser för koderna i övning 11.2.

11.11. Bestäm kontrollmatriser för koderna i uppgift 11.3.

11.12. 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 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**).

11.13. 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 **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 11.3.

11.14. **Perfekta koder.** En gruppkod kallas **perfekt** om till varje vektor $\mathbf{x} \in \mathbb{Z}_2^n$ existerar exakt ett kodord vars avstånd från **x** är högst t. Visa att Hammingkoderna är perfekta (med $t = 1$).

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 at 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)-Golaykoden har man:

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

11.15. 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).

Kapitel 12

TIPSPROBLEMET

Tipsproblemet handlar om möjligheten att så billigt som möjligt vinna på tipset. Frågan är sådan: hur många tipskuponger skall man fylla i för att garantera att man får rätt i $n-1$ matcher på n möjliga? På en vanlig tipskupong (med 13 matcher) gäller alltså frågan hur många tipsrader måste man fylla i för att ha 12 rätt.

Låt oss börja med ett specialfall som gäller 6 rätt i 7 matcher. Vi skall också förutsätta att i dessa 7 matcher har man antingen 1 (vinst av hemmalaget) eller $\times$ (oavgjort). Vi kan lösa problemet på följande sätt: Låt oss bortse från en av matcherna (t ex den sista) och fylla i alla möjliga utfall av de återstående 6. Då måste vi fylla i $2^6 = 64$ tipsrader.

Det visar sig att det räcker att fylla i enbart 16 tipsrader i enlighet med alla kodord i Hammingkoden för att få samma effekt dvs 6 rätt. Nedan skriver vi ut alla kodord

$$\begin{array}{ccccccc}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & 1 & 1 & 1 \\
\times & \times & 1 & \times & \times & 1 & 1 \\
\times & \times & 1 & 1 & 1 & \times & \times \\
\times & 1 & \times & \times & 1 & \times & 1 \\
\times & 1 & \times & 1 & \times & 1 & \times \\
\times & 1 & 1 & \times & 1 & 1 & \times \\
\times & 1 & 1 & 1 & \times & \times & 1 \\
1 & \times & \times & \times & 1 & 1 & \times \\
1 & \times & \times & 1 & \times & \times & 1 \\
1 & \times & 1 & \times & 1 & \times & 1 \\
1 & \times & 1 & 1 & \times & 1 & \times \\
1 & 1 & \times & \times & \times & 1 & 1 \\
1 & 1 & \times & 1 & 1 & \times & \times \\
1 & 1 & 1 & \times & \times & \times & \times \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

i Hammingkoden som svarar mot matrisen $\mathbf{H}_3$ med $\times$ i stället för 0 (se (11.14) i Kapitel 11).

Varför fungerar den konstruktionen? Kodorden utgör 16 av $2^7 = 128$ vektorer i $\mathbb{Z}_2^7$. Varje kodord har sin cirkel med radie 1 som inte överlappar med andra cirklar som svarar mot andra kodord. Varje sådan cirkel består av 8 vektorer (kodordet självt och 7 vektorer som skiljer sig från detta på ett ställe). Dessa cirklar innehåller sammanlagt $16 \cdot 8 = 128$ vektorer dvs alla vektorer i $\mathbb{Z}_2^7$. Varje tipsrad är en sådan vektor och varje vektor ligger på ett avstånd ≤ 1 från ett kodord. Alltså garanterar alla kodord i Hammingkoden att varje möjligt utfall kommer att skilja sig från ett kodord på högst ett ställe. Detta betyder att med all säkerhet kommer vi att ha minst 6 rätt om vi väljer som tipsrader kodorden i Hammingkoden av längden 7 (pröva!).

Hur är det med det riktiga tipsproblemet för 13 matcher? Det naiva svaret är att det räcker med 3^{12} tipsrader (man kan bortse från en match och fylla i alla möjliga tipsrader för 12 matcher). En lämplig Hammingkod ger i detta fall 3^{10} tipsrader – en mycket stor skillnad! Men detta svar döljer ganska oväntade matematiska problem. För det första, om man vill lösa problemet för 3 olika utfall av en match $(1, \times, 2)$, måste man utnyttja restaritmetiken $\mathbb{Z}_3$ i stället för $\mathbb{Z}_2$. Den består av alla rester vid division med tre dvs $\mathbb{Z}_3 = \{0, 1, 2\}$ med följande reknelagar:

$$\begin{array}{c|ccc} + & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{array} \qquad \begin{array}{c|ccc} * & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{array}$$

För att konstruera Hammingkoder måste man välja lämpliga matriser med ledning av samma principer som i restaritmetiken $\mathbb{Z}_2$. Lämpliga matriser måste sakna nollkolonner för att utesluta lösningar av vikten 1. Lösningar av vikten 2 finns då och endast då en kolonn är en multipel av en annan (för att kontrollera det gå tillbaka till (11.11) i Kapitel 11 och tänk på vad som händer om högst upp i stället för en vektor med två koordinater lika med 1 har man en vektor med två koordinater lika med 1 eller 2). Alltså definierar en matris $\mathbf{H}$ över $\mathbb{Z}_3$ en ettfelkorrigerande kod då och endast då den saknar nollkolonner och proportionella kolonner. Ett exempel på en sådan matris (som också konstruerades av Golay och Hamming) är följande:

$$\begin{array}{cc} & \begin{array}{ccccccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} \end{array} \\ \mathbf{H_{3,13}} = & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{bmatrix} \end{array}$$

Kolonnerna i den matrisen är alla $\neq 0$ tal med 3 siffror i talsystemet i bas 3 med ettan som första betydande siffra. Matrisen $\mathbf{H_{3,13}}$ definierar ett ekvationssystem med 13 obekanta. Lösningarna beror på 10 parametrar (x_1, x_2 och x_5 är kontrollsymboler medan de övriga 10 variablerna antar helt godtyckliga värden 0, 1 och 2). Man får 3^{10} vektorer i rummet $\mathbb{Z}_3^{13}$ som innehåller 3^{13} vektorer. Varje lösning definierar sin cirkel bestående av alla vektorer i $\mathbb{Z}_3^{13}$ som skiljer sig från denna på högst ett ställe. En sådan cirkel har $13 + 13 = 26$ punkter. Tillsammans med lösningen i dess centrum ger detta 27 vektorer. Antalet cirklar är 3^{10} och tillsammans omfattar de $3^{10} \times 27 = 3^{13}$ vektorer. Alltså ligger varje vektor i $\mathbb{Z}_3^{13}$ (dvs varje möjlig tipsrad i det svenska tipset) på ett avstånd ≤ 1 från exakt ett kodord. Alltså räcker det med $3^{10} = 59049$ tipsrader i stället för $3^{12} = 531441$ för att ha 12 rätt på 13 matcher.

Det faktum att man löser tipsproblemet för det svenska tipset är i själva verket en ren slump. Låt oss observera att Hammingkoderna i restaritmetiken $\mathbb{Z}_2$ består av vektorer av längden $2^n - 1$ dvs man har sådana koder för 1, 3, 7, 15,... matcher. När man ersätter $\mathbb{Z}_2$ med $\mathbb{Z}_3$ har man i stället Hammingkoderna av längder $(1/2)(3^n - 1)$ dvs man har sådana koder för 1, 4, 13, 40,... matcher. Det gör att man kan lösa problemet för det svenska tipset. Men problemet är inte löst för det engelska (antalet matcher är 12). Det finns ett ganska långt arbete där det visas att lösningen av tipsproblemet för 5 matcher är 27 (dvs man måste fylla i 27 tipsrader för att få 4 rätt på 5 matcher). Men problemet är inte löst för 6 matcher (möjligen är svaret 81).

Men låt oss avsluta med ett riktigt lönande och realistiskt tips genom att betrakta fallet av 4 matcher. Om man vill ha 3 rätt på 4 matcher så ger den naiva lösningen (dvs alla möjliga tipsrader för 3 matcher) svaret 27. Låt oss i stället ta Hammingmatrisen

$$\mathbf{H}_{\mathbf{2},\mathbf{4}} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix}$$

och lösa (över $\mathbb{Z}_3$) ekvationssystemet

$$\begin{array}{ccccccccc} & & x_2 & + & x_3 & + & x_4 & = & 0, \\ x_1 & + & & & x_3 & + & 2x_4 & = & 0. \end{array}$$

Man får lätt

$$\begin{array}{ccccc} x_1 & = & 2x_3 & + & x_4, \\ x_2 & = & 2x_3 & + & 2x_4. \end{array}$$

(observera att $-1 = 2$ ty $1 + 2 = 0$). Nu väljer man godtyckliga värden 0,1,2 på x_3 och x_4 och man räknar ut x_1 och x_2.

Man får 9 vektorer i $\mathbb{Z}_3^4$:

(12.1)

x_3	x_4	x_2	x_1
$\times$	$\times$	$\times$	$\times$
$\times$	1	2	1
$\times$	2	1	2
1	$\times$	2	2
1	1	1	$\times$
1	2	$\times$	1
2	$\times$	1	1
2	1	$\times$	2
2	2	2	$\times$

(vi skriver $\times$ i stället för 0). Dessa 9 tipsrader ger 3 rätt på 4 matcher. I själva verket har varje kodord sin cirkel bestående av 9 vektorer varvid 8 av dem skiljer sig från kodordet på ett ställe. Vi har 9 sådana cirklar dvs $9 \cdot 9 = 81$ vektorer i $\mathbb{Z}_3^4$. Detta betyder att vi får alla vektorer i $\mathbb{Z}_3^4$ så att varje vektor skiljer sig från ett kodord (12.1) på högst ett ställe. Alltså är minst 3 matcher av 4 rätt tippade. I stället för 27 tipsrader kan vi fylla i enbart 9 och spara (i dagens priser) 18 kronor (om man bara inte spelar för mycket!).

Det faktum att Hamming-koderna kan tillämpas på tipsproblemet beror på deras ganska sällsynta egenskaper. De är så kallade **perfekta koder**. En t-felkorrigerande kod $C \subseteq \mathbb{Z}_p^n$ kallas perfekt om för varje vektor $\mathbf{x} \in \mathbb{Z}_p^n$ existerar exakt ett kodord $\mathbf{a}$ sådant att avståndet mellan $\mathbf{x}$ och $\mathbf{a}$ är högst t. Vi har utnyttjat den egenskapen flera gånger (med $t = 1$) under vår diskussion av tipsproblemet. Perfekta koder har studerats mycket intensivt och ett av de viktigaste resultaten om sådana koder visades 1973 av A. Tietäväinen. Med ledning av tidigare resultat av J.H. van Lint visade han att alla icke-triviala perfekta koder har exakt samma antal kodord, längd och felkorrigeringsförmåga som Hamming-koderna eller en av två så kallade Golay-koder med standardbeteckningar $\mathcal{G}_{11}$ och $\mathcal{G}_{23}$. De två Golay-koderna tillhör de mest intressanta bland alla kända koder. Koden $\mathcal{G}_{11}$ består av 3^6 kodord i $\mathbb{Z}_3^{11}$ och korrigerar 2 fel. Dessa parametrar gör att koden är mycket lämplig i tipssammanhang. Dess matris kan ges på följande form:

$$\mathbf{G}_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 2 \end{bmatrix}$$

Kodorden är alla 3^6 lösningar i $\mathbb{Z}_3^{11}$ till ekvationssystemet bestående av 6 ekvationer med 11 obekanta vars koefficienterna utgörs av raderna i denna matris. Man får säkert 9 rätt på 11 matcher vid användningen av den koden. Koden $\mathcal{G}_{23}$ består av 2^{12} kodord i $\mathbb{Z}_2^{23}$ och korrigerar 3 fel.

Kapitel 13

POLYNOMKODER

Vi ägnar detta avsnitt åt konstruktioner av en stor klass av riktigt bra koder – s k BCH-koder*. Konstruktionen bygger på våra tidigare kunskaper om koder, polynomringar och kroppsutvidgningar. Vi börjar med en allmän definition av polynomkoder. BCH-koder hör just till den här klassen. Vi begränsar oss till polynom med koefficienter i kroppen med 2 element, men polynomkoder kan konstrueras på liknande sätt med hjälp av andra polynomringar över ändliga kroppar.

Principen för konstruktion av polynomkoder är mycket enkel: Man tolkar en binär följd som ett polynom, t ex

$$\begin{aligned}
101 &\mapsto 1\cdot 1+0\cdot X+1\cdot X^2=1+X^2,\\
111 &\mapsto 1\cdot 1+1\cdot X+1\cdot X^2=1+X+X^2,\\
1011 &\mapsto 1\cdot 1+0\cdot X+1\cdot X^2+1\cdot X^3=1+X^2+X^3,\\
0101 &\mapsto 0\cdot 1+1\cdot X+0\cdot X^2+1\cdot X^3=X+X^3.
\end{aligned}$$

Därefter fixerar man ett polynom $g(X)\in\mathbb{Z}_2[X]$ (ett generatorpolynom), och man ordnar mot en dataföljd i form av ett polynom $p(X)\in\mathbb{Z}_2[X]$ dess kodpolynom $p(X)g(X)$. Koefficienterna av produkten ger kodordet.

(13.1) Exempel. Låt $g(X)=1+X+X^2$. För att kodera 1011 tolkar vi den följden som polynomet $p(X)=1+X^2+X^3$ och räknar ut produkten $p(X)g(X)=(1+X^2+X^3)(1+X+X^2)=1+X+X^5$, dvs mot 1011 svarar kodordet 110001. $\square$

Helt allmänt antar vi följande definition:

*Dessa koder konstruerades av R. C. Bose, D. K. Ray-Chaudhuri (1960) och A. Hocquenghem (1959).

(13.2) Definition. Med en (m,n)-**polynomkod** $C \subseteq \mathbb{Z}_2^n$ med **generatorpolynomet** $g(X) = a_0 + a_1X + \ldots + a_kX^k$, där $k = n - m$, menas den kod som man får då varje dataföljd $x_0x_1 \ldots x_{m-1}$ betraktas som ett polynom

$$p(X) = x_0 + x_1X + x_2X^2 + \ldots + x_{m-1}X^{m-1}$$

och man mot $p(X)$ ordnar produkten $p(X)g(X)$ vars koefficienter sedan tolkas som ett kodord av längden n. □

(13.3) Exempel. För att definiera en $(2,5)$-kod $C \subseteq \mathbb{Z}_2^5$ måste man ha ett polynom $g(X)$ av grad $k = 5 - 2 = 3$. Tag t ex $g(X) = 1 + X + X^3$. Då är

$$x_0x_1 \mapsto x_0+x_1X \mapsto (x_0+x_1X)(1+X+X^3) \mapsto y_0+y_1X+y_2X^2+y_3X^3+y_4X^4 \mapsto y_0y_1y_2y_3y_4,$$

där $y_0 = x_0, y_1 = x_0 + x_1, y_2 = x_1, y_3 = x_0, y_4 = x_1$. Alltså

$$\begin{aligned} 00 &\mapsto 00000, \\ 01 &\mapsto 01101, \\ 10 &\mapsto 11010, \\ 11 &\mapsto 10111. \end{aligned}$$

□

(13.4) Anmärkning. Polynomkoder är gruppkoder. Samma effekt som multiplikation med $1 + X + X^3$ kan man nå genom att multiplicera med en lämplig (5×2)-matris. En sådan generatormatris i exempel (13.3) är:

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

Allmänt kontrollerar man utan större problem att multiplikation av polynom:

$$x_0 + x_1X + \ldots + x_{m-1}X^{m-1} \mapsto (x_0 + x_1X + \ldots + x_{m-1}X^{m-1})(a_0 + a_1X + \ldots + a_kX^k)$$

ger samma resultat som matrismultiplikation:

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{bmatrix} \mapsto \begin{bmatrix} a_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \cdot & \cdot & \cdots & a_0 \\ \cdot & \cdot & \cdots & a_1 \\ \vdots & \vdots & & \vdots \\ a_k & a_{k-1} & \cdots & \cdot \\ 0 & a_k & \cdots & \cdot \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & a_k \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{bmatrix}.$$

Generatormatrisen för koden får man alltså genom att skriva ut koefficienterna för generatorpolynomet i första kolonnen och sedan flytta dem neråt ett steg i taget tills man får en $(n \times m)$-matris. □

(13.5) Anmärkning. En polynomkod $C \subseteq \mathbb{Z}_2^n$ med ett generatorpolynom $g(X)$ definierar en linjär transformation $p(X) \mapsto p(X)g(X)$ från vektorrummet av alla polynom vars grad är $\leq m-1$ till vektorrummet av alla polynom vars grad är $\leq n-1$. Generatormatrisen G är just transformationsmatrisen i basen $1, X, \ldots, X^{m-1}$ för det första rummet och $1, X, \ldots, X^{n-1}$ för det andra. □

För att kunna konstruera BCH-koder behöver vi polynom i ringen $\mathbb{Z}_2[X]$ som har en speciell egenskap. Låt oss repetera från avsnittet om kroppsutvidgningar att restringen $K = \mathbb{Z}_2[X]/(p(X)) = \mathbb{Z}_2[\alpha]$, där $\alpha = [X]_{p(X)}$, är en kropp då och endast då polynomet $p(X)$ är irreducibelt. De polynom som vi är intresserade av har följande egenskap:

(13.6) Definition. Man säger att ett irreducibelt polynom $p(X) \in \mathbb{Z}_2[X]$ är **primitivt** om varje nollskilt element i kroppen $K = \mathbb{Z}_2[X]/(p(X)) = \mathbb{Z}_2[\alpha]$ är en potens av α. □

Man kan också uttrycka den egenskapen så att den multiplikativa gruppen $K^* = K \setminus \{0\}$ genereras av α [†]. Låt oss betrakta ett exempel:

(13.7) Exempel. Vi skall visa på två olika sätt att polynomet $X^3 + X + 1$ är primitivt. Låt $K = \mathbb{Z}_2[X]/(X^3 + X + 1) = \mathbb{Z}_2[\alpha]$, där $\alpha^3 + \alpha + 1 = 0$ dvs $\alpha^3 = 1 + \alpha$. Kroppen K har 8 element $a + b\alpha + c\alpha^2$, där $a, b, c \in \mathbb{Z}_2$. Vi har följande potenser av α: $\alpha^1 = \alpha$, α^2, $\alpha^3 = 1 + \alpha$ och vidare successivt:

[†] Varje ändlig delgrupp till den multiplikativa gruppen i en kropp K är cyklisk, dvs den består av potenserna av ett fixt element. Vi kommer inte att bevisa eller behöva denna sats.

$$\begin{aligned}
\alpha^4 &= \alpha+\alpha^2,\\
\alpha^5 &= \alpha^2+\alpha^3 = 1+\alpha+\alpha^2,\\
\alpha^6 &= \alpha+\alpha^2+\alpha^3 = 1+\alpha^2,\\
\alpha^7 &= \alpha+\alpha^3 = 1.
\end{aligned}$$

Detta visar att vi verkligen får 7 olika potenser av α, dvs alla nollskilda element i K.

Nu ger vi ett annat bevis som inte bygger på beräkningar: Eftersom K^* är en grupp med 7 element måste den vara cyklisk (7 är ett primtal). Dessutom genereras den av ett godtyckligt element $\neq 1$. Alltså ger potenserna av α alla element i K^*. □

(13.8) Anmärkning. Om $p(X)$ är ett irreducibelt polynom av r-te graden så består kroppen $K = \mathbb{Z}_2[X]/(p(X)) = \mathbb{Z}_2[\alpha]$ av 2^r element (se kapitlet Kroppsutvidgningar). Detta betyder att polynomet $p(X)$ är primitivt om dess nollställe α har ordningen $2^r - 1$ i gruppen K^*. Gruppen $K^* =< \alpha >$ består av potenserna $\alpha, \alpha^2, \ldots, \alpha^{2^r-2}, \alpha^{2^r-1} = 1$. Observera att alla nollskilda element i kroppen K är lösningar till ekvationen $X^{2^r-1} - 1 = 0$ eftersom antalet element i gruppen K^* är just $2^r - 1$. □

Eftersom primitiva polynom har en stor betydelse för konstruktioner av BCH-koder finns det omfattande tabeller av sådana polynom. Vi ger här en mycket kort tabell som vi kan utnyttja för våra kodkonstruktioner.

(13.9) Primitiva polynom av grad ≤ 5:

$n = 2:\quad 1 + X + X^2,$

$n = 3:\quad 1 + X + X^3,\ 1 + X^2 + X^3,$

$n = 4:\quad 1 + X + X^4,\ 1 + X^3 + X^4,$

$n = 5:\quad 1 + X^2 + X^5,\ 1 + X^3 + X^5,\ 1 + X + X^2 + X^3 + X^5,\ 1 + X + X^2 + X^4 + X^5,$

$1 + X + X^3 + X^4 + X^5,\ 1 + X^2 + X^3 + X^4 + X^5.$

□

Nu är vi beredda att definiera BCH-koder.

(13.10) Definition av BCH-koder. Med en BCH-kod menar man en kod som är konstruerad på följande sätt:

1. Man bestämmer det minimiavstånd $2t+1$ mellan kodorden som man vill ha och ett heltal r sådant att $2^r > 2t+1$.

2. Man väljer ett primitivt polynom $g_1(X)$ av grad r och dess nollställe $\alpha \in K$, där $K = \mathbb{Z}_2[X]/(p(X)) = \mathbb{Z}_2[\alpha]$.

3. Man bestämmer irreducibla polynom $g_2(X), g_3(X), \ldots g_{2t}(X)$ som har $\alpha^2, \alpha^3, \ldots, \alpha^{2t}$ som sina nollställen.

4. Man räknar ut produkten $g(X)$ av **olika** polynom bland $g_i(X)$, för $i = 1, \ldots, 2t$

5. Man definierar den (m,n)-polynomkod $C \subseteq \mathbb{Z}_2^n$, där $n = 2^r - 1$ och $m = 2^r - 1 - \text{grad}(g(X))$, som genereras av polynomet $g(X)$.

(13.11) Anmärkning. Man kan visa att det finns primitiva polynom av varje given grad så att valet av polynomet g_1 i punkt 2 alltid är möjligt. Man kan också visa att polynomet $g(X)$ konstruerat i punkten 4 alltid har grad mindre än $2^r - 1$ (se övning 13.8). □

(13.12) Huvudsatsen om BCH-koder. *Vikten av en BCH-kod konstruerad i enlighet med definitionen* (13.10) *är minst* $2t+1$ *och antalet kontrollsymboler i den* (*dvs graden av generatorpolynomet* $g(X)$) *är högst* tr.

Innan vi bevisar satsen ger vi ett exempel.

(13.13) Exempel. Vi skall konstruera en kod med minimiavståndet mellan kodorden lika med 5. Man måste välja r så att $2^r > 2\cdot 2+1 = 5$. Vi väljer $r = 4$ och $g_1(X) = 1+X^3+X^4$. Låt $K = \mathbb{Z}_2[X]/(g_1(X)) = \mathbb{Z}_2[\alpha]$. Vi har $g_1(\alpha) = 0$. Den likheten medför att $g_1(\alpha^2) = g_1(\alpha^4) = 0$. I själva verket gäller det att om $h(X)$ är ett godtyckligt polynom och β är dess nollställe, så är också β^2 dess nollställe, ty om $h(X) = a_0 + a_1X + \ldots + a_sX^s$ så är

$$
\begin{aligned}
h(\beta^2) &= a_0 + a_1\beta^2 + a_2(\beta^2)^2 + \ldots + a_s(\beta^2)^s = \\
&= [a_0 + a_1\beta + a_2\beta^2 + \ldots + a_s\beta^s]^2 = h(\beta)^2 \ {}^{\ddagger}.
\end{aligned}
\tag{13.14}
$$

Eftersom $g_1(\alpha) = g_1(\alpha^2) = g_1(\alpha^4) = 0$, så är $g_1(X) = g_2(X) = g_4(X)$. Vi måste beräkna polynomet $g_3(X)$ som har α^3 som sitt nollställe. Vi vet att graden av $g_3(X)$ är mindre än eller lika med 4, dvs α^3 är ett nollställe till ett av polynomen $1+X+X^2$, $1+X+X^3$, $1+X^2+X^3$, $1+X+X^4$, $1+X^3+X^4$ eller $1+X+X^2+X^3+X^4$ (se (13.9)). Att bestämma vilket polynom bland dessa som har α^3 som sitt nollställe (det finns enbart ett enligt övning 13.8) är något arbetsamt, men enkelt. Vi vet att α är ett nollställe till $g_1(X)$, dvs $\alpha^4 + \alpha^3 + 1 = 0$. Alltså har vi:

‡Här utnyttjar vi märkliga egenskaper hos aritmetiken modulo 2. Eftersom $1+1=0$, dvs $2=0$, har vi

$$(x_1 + x_2 + \ldots + x_s)^2 = x_1^2 + x_2^2 + \ldots + x_s^2.$$

Dessutom har vi $a_i^2 = a_i$, ty $a_i = 0$ eller 1.

$$\alpha^4 = 1 + \alpha^3.$$

Vi utnyttjar den likheten och räknar ut andra potenser av α:

$$\begin{aligned}
\alpha^5 &= \alpha \cdot \alpha^4 = \alpha + \alpha^4 = 1 + \alpha + \alpha^3, \\
\alpha^6 &= \alpha \cdot \alpha^5 = \alpha + \alpha^2 + \alpha^4 = 1 + \alpha + \alpha^2 + \alpha^3, \\
\alpha^7 &= \alpha \cdot \alpha^6 = \alpha + \alpha^2 + \alpha^3 + \alpha^4 = 1 + \alpha + \alpha^2, \\
\alpha^8 &= \alpha \cdot \alpha^7 = \alpha + \alpha^2 + \alpha^3, \\
\alpha^9 &= \alpha \cdot \alpha^8 = \alpha^2 + \alpha^3 + \alpha^4 = 1 + \alpha^2, \\
\alpha^{10} &= \alpha \cdot \alpha^9 = \alpha + \alpha^3, \\
\alpha^{11} &= \alpha \cdot \alpha^{10} = \alpha^2 + \alpha^4 = 1 + \alpha^2 + \alpha^3, \\
\alpha^{12} &= \alpha \cdot \alpha^{11} = \alpha + \alpha^3 + \alpha^4 = 1 + \alpha.
\end{aligned}$$

Vi avbryter här eftersom den högsta potens av α som vi behöver är α^{12} (α^3 kan vara ett nollställe till ett polynom vars grad är högst lika med 4) men det finns inte många potenser kvar, ty $\alpha^{15} = 1$ (varför?). Nu kan vi lätt kontrollera att α^3 är ett nollställe till $1 + X + X^2 + X^3 + X^4$:

$$\begin{aligned}
1 + \alpha^3 + (\alpha^3)^2 + (\alpha^3)^3 + (\alpha^3)^4 = \;& 1 + \alpha^3 + \alpha^6 + \alpha^9 + \alpha^{12} = \\
& 1 + \alpha^3 + (1 + \alpha + \alpha^2 + \alpha^3) + (1 + \alpha^2) + (1 + \alpha) = 0,
\end{aligned}$$

ty $1 + 1 = 2 = 0$. Om vi inte har tur att hitta det rätta polynomet med en gång, kan vi pröva oss fram. T ex visar vi att α^3 inte är ett nollstället till $1 + X + X^4$. Antag motsatsen, dvs

$$1 + \alpha^3 + (\alpha^3)^4 = 1 + \alpha^3 + \alpha^{12} = 1 + \alpha^3 + (1 + \alpha) = \alpha + \alpha^3 = \alpha(1 + \alpha^2) = \alpha(1 + \alpha)^2 = 0.$$

Alltså är $\alpha = 0$ eller $\alpha = 1$. Detta strider mot likheten $1 + \alpha^3 + \alpha^4 = 0$ som α uppfyller, ty $\alpha = 0$ eller $\alpha = 1$ ger $1 = 0$.

Nu kan vi avsluta vår konstruktion genom att beräkna produkten $g(X)$ av olika polynom bland $g_1(X)$, $g_2(X)$, $g_3(X)$ och $g_4(X)$, dvs

$$g(X) = g_1(X) g_3(X) = (1 + X^3 + X^4)(1 + X + X^2 + X^3 + X^4) = 1 + X + X^2 + X^4 + X^8$$

Polynomet $g(X)$ genererar en (7,15) BCH-kod $C \subseteq \mathbb{Z}_2^{15}$ som korrigerar minst 2 fel, där

$$\begin{aligned} x_0x_1\dots x_6 \mapsto p(X) &= x_0 + x_1X + \dots + x_6X^6 \mapsto p(X)g(X) = \\ &= y_0 + y_1X + \dots + y_{14}X^{14} \mapsto y_0y_1\dots y_{14}. \end{aligned}$$

T ex

$$\begin{aligned} 1010001 &\mapsto 1 + X^2 + X^6 \mapsto (1 + X^2 + X^6)(1 + X + X^2 + X^4 + X^8) = \\ &= 1 + X + X^3 + X^7 + X^{14} \mapsto 110100010000001. \end{aligned}$$

□

(13.15) Bevis av huvudsatsen om BCH-koder. Kodorden har formen $p(X)g(X)$, där $p(X)$ har grad $< m$. Detta innebär att varje kodpolynom är lika med 0 då man sätter in $X = \alpha, \alpha^2, \dots, \alpha^{2t}$ (ty $\alpha, \alpha^2, \dots, \alpha^{2t}$ är nollställen till polynomet $g(X)$ som är delbart med $g_1(X), g_2(X), \dots, g_{2t}(X)$). Vi påstår att $p(X)g(X), p \not\equiv 0$[§], måste ha minst $2t+1$ koefficienter som är lika med 1, dvs minimivikten av kodorden är minst $2t+1$. Låt oss anta motsatsen, dvs att för något $p(X), p \not\equiv 0$, är

$$p(X)g(X) = x_1X^{r_1} + x_2X^{r_2} + \dots + x_{2t}X^{r_{2t}}, \;\; r_1 < r_2 < \dots < r_{2t} \le n-1,$$

där $x_i = 0$ eller 1 och inte alla x_i är $= 0$. Om man sätter in $X = \alpha, \alpha^2, \dots, \alpha^{2t}$, får man ett linjärt homogent ekvationssystem:

$$\begin{aligned} \alpha^{r_1}x_1 + \alpha^{r_2}x_2 + \cdots + \alpha^{r_{2t}}x_{2t} &= 0, \\ (\alpha^2)^{r_1}x_1 + (\alpha^2)^{r_2}x_2 + \cdots + (\alpha^2)^{r_{2t}}x_{2t} &= 0, \\ &\;\;\vdots \\ (\alpha^{2t})^{r_1}x_1 + (\alpha^{2t})^{r_2}x_2 + \cdots + (\alpha^{2t})^{r_{2t}}x_{2t} &= 0. \end{aligned}$$

Systemet har $2t$ ekvationer med $2t$ obekanta och en icke-trivial lösning $x_1, x_2, \dots, x_{2t}$. Alltså måste systemets determinant

$$\begin{vmatrix} \alpha^{r_1} & \alpha^{r_2} & \dots & \alpha^{r_{2t}} \\ (\alpha^2)^{r_1} & (\alpha^2)^{r_2} & \dots & (\alpha^2)^{r_{2t}} \\ \dots & \dots & \dots & \dots \\ (\alpha^{2t})^{r_1} & (\alpha^{2t})^{r_2} & \dots & (\alpha^{2t})^{r_{2t}} \end{vmatrix}$$

vara lika med 0. Denna determinant är ett specialfall av en mycket känd determinant:

$$\begin{vmatrix} x_1 & x_2 & \dots & x_m \\ x_1^2 & x_2^2 & \dots & x_m^2 \\ \vdots & \vdots & & \vdots \\ x_1^m & x_2^m & \dots & x_m^m \end{vmatrix} = x_1x_2\cdots x_m \prod_{1\le i<j\le m} (x_i - x_j)$$

[§] Med $p \not\equiv 0$ betecknar vi ett polynom som inte är lika med nollpolynomet.

som kallas Vandermondes determinant. Här är $x_1 = \alpha^{r_1}$, $x_2 = \alpha^{r_2}, \ldots, x_m = \alpha^{r_m}$, $m = 2t$. Eftersom alla potenser $\alpha^{r_1}, \alpha^{r_2}, \ldots, \alpha^{r_{2t}}$ är olika (se (13.8)) får vi att determinanten är skild från 0. Den motsägelsen som visar att vikten av kodorden måste vara $\geq 2t + 1$.

Enligt (13.14) har vi $g_j(\alpha^{2j}) = g_j(\alpha^j)^2 = 0$ så att $g_{2j}(X) = g_j(X)$. Alltså när man räknar polynomet $g(X)$, kan man stryka minst hälften av polynomen $g_1(X)$, $g_2(X), \ldots, g_{2t}(X)$. Produkten av de återstående har graden $\leq t \cdot r$, eftersom antalet polynom är t och graden av varje polynom $g_j(X)$ är $\leq r$ (se övning 13.9(b)). Alltså är graden av $g(X) \leq tr$.

(13.16) Anmärkning. BCH-koder av måttlig längd (några tusen symboler i kodorden) tillhör de bästa bland alla kända koder. Om man jämför hastigheten 7/15 av koden ur exempel (13.13), med hastigheten 1/5 av den repetitionskod som korrigerar 2 fel genom att upprepa varje information 5 gånger, så är skillnaden märkbar. Om man vill korrigera 2 fel i informationsföljder av längden 7 med hjälp av repetitionskoden måste man sända 35 signaler medan den konstuerade BCH-koden enbart behöver 15 signaler! En av de första BCH-koder som hade stora tillämpningar var en (92, 127)-kod med generatorpolynomet

$$\begin{aligned} g(X) &= (1 + X + X^7)(1 + X + X^2 + X^3 + X^7)(1 + X^2 + X^3 + X^4 + X^5 + X^7) \\ &\quad (1 + X + X^2 + X^4 + X^5 + X^6 + X^7)(1 + X + X^2 + X^3 + X^4 + X^5 + X^7). \end{aligned}$$

Den korrigerar 5 fel och dess hastighet är 92/127 (jämför med 1/11 för den repetitionskod som korrigerar 5 fel!). Det räcker att endast sända 127 signaler i stället för $11 \cdot 92 = 1012$ då man repeterar varje information 11 gånger. Här följer en liten tabell med några exempel på kända BCH-koder:

m	n	t	m	n	t
4	7	1	36	63	5
			30	63	6
11	15	1	24	63	7
7	15	2	18	63	10
5	15	3			
			92	127	5
26	31	1	64	127	10
21	31	2			
16	31	3	231	255	3
11	31	5	215	255	5
6	31	7	139	255	15

□

Vi skall avsluta detta avsnitt med ett exempel som visar hur man kan få Hammingkoderna som ett specialfall av BCH-koder.

(13.17) Exempel. Om man vill ha en kod av vikten ≥ 3, så måste man välja r så att $2^r > 3$, dvs $r \geq 2$. Detta innebär att man kan välja ett godtyckligt primitivt polynom $g_1(X)$ av grad $r \geq 2$. Eftersom $g_2(X) = g_1(X)$ (som vanligt), så är $g(X) = g_1(X)$. Vi får alltså en (m,n)-kod $C \subseteq \mathbb{Z}_2^n$, där $m = 2^r - 1 - r$ och $n = 2^r - 1$, som korrigerar 1 fel. Detta är helt enkelt Hammingkoden, ty kontrollmatrisen för den koden har $n = 2^r - 1$ olika kolonner och alla dessa kolonner har längden r – det finns bara en sådan matris (så när som på omkastning av kolonnernas ordningsföljd). Sådana matriser definierar just Hammingkoderna (se Kapitel 11). Om t ex $g(X) = 1 + X + X^3$, då får vi $C \subset \mathbb{Z}_2^7$, där den generatormatris som svarar mot $g(X)$ är (se (13.4))

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

□

ÖVNINGAR

13.1. Definiera en (3,6)-polynomkod $C \subseteq \mathbb{Z}_2^6$. Ange generatorpolynomet och skriv ut alla kodord.

13.2. Skriv ut alla kodord för den (m,n)-kod $C \subseteq \mathbb{Z}_2^n$ som genereras av polynomet $g(X)$ då

a) $g(X) = 1 + X + X^2, \quad m = 3.$

b) $g(X) = 1 + X^2 + X^3, \quad n = 6.$

Definiera dessa koder som matriskoder (välj lämpliga generatormatriser).

13.3. Konstruera alla möjliga BCH-koder med hjälp av polynomet $g_1(X) = 1 + X + X^3$.

13.4. Låt $K = \mathbb{Z}_2[X]/(X^4 + X + 1) = \mathbb{Z}_2[\alpha]$, där $\alpha = [X]$.

(a) Motivera att α genererar gruppen K^*.

(b) Visa att $1 + X + X^2 + X^3 + X^4$ är irreducibelt och α^3 är dess nollställe.

(c) Konstruera BCH-koder som korrigerar 1, 2 och 3 fel med hjälp av polynomet $g_1(X) = 1 + X + X^4$.

13.5. (a) Kontrollera att polynomen i (13.9) verkligen är primitiva.

(b) Visa att polynomet $1 + X + X^2 + X^3 + X^4$ inte är primitivt.

13.6. Konstruera en $(m, 15)$-BCH-kod som korrigerar 3 fel med så stort m som möjligt.

13.7. Låt $C \subseteq \mathbb{Z}_2^n$ vara en (m,n)-polynomkod som genereras av ett polynom $g(X)$ sådant att $g(1) = 0$. Visa att vikten av den koden är jämn.

13.8. *Minimalpolynom.* Låt $K \subseteq L$ vara två kroppar och $\alpha \in L$ ett nollställe till ett irreducibelt polynom $g(X) \in K[X]$ med högsta koefficienten lika med 1.

(a) Visa att $g(X)$ är en delare till varje polynom $h(X) \in K[X]$ som har α som ett nollställe (så att $g(x)$ har minsta möjliga grad bland alla polynom i $K[X]$ som har α som sitt nollställe).

(b) Visa att två olika irreducibla polynom i $K[X]$ med högsta koefficienter lika med 1 saknar gemensamma nollställen i L.

Anmärkning. Polynomet $g(X)$ i (a) kallas **minimalpolynomet** för α över K. T ex är $X^2 - 2$ minimalpolynomet för $\sqrt{2}$ över de rationella talen

13.9. Låt $p(X) \in \mathbb{Z}_2[X]$ vara ett irreducibelt polynom av grad r och $K = \mathbb{Z}_2[X]/(p(X)) = \mathbb{Z}_2[\alpha]$. Låt $\beta \in K^*$.

(a) Visa att β är ett nollställe till $X^{2^r-1} - 1$ och följaktligen till ett irreducibelt polynom $g(X) \in \mathbb{Z}_2[X]$ som dividerar $X^{2^r-1} - 1$. Motivera också att $g(X)$ är minimalpolynomet för β över $\mathbb{Z}_2$ (se förra övningen).

(b) Låt $g(X)$ vara minimalpolynomet för β över $\mathbb{Z}_2$. Visa att graden av $g(X)$ är $\leq r$.

Ledning. Observera att $1, \beta, \beta^2, \ldots, \beta^r$ måste vara linjärt beroende över K därför att dimensionen av K som vektorrum över $\mathbb{Z}_2$ är lika med r (se kapitlet Kroppsutvidgningar).

Anmärkning. Man kan visa att graden av $g(X)$ t o m är en delare till r.

(c) Motivera att polynomet $g(X)$ i punkten 4 av definitionen (13.10) har graden $< 2^r - 1$.

Kapitel 14

NÅGOT OM KRYPTERING

Behovet av att skydda information har funnits mycket länge, men först i samband med utvecklingen av datatekniken har det blivit ett allmänt problem för alla moderna samhällen. Stora datamängder måste ofta skyddas från obehörigt intrång med hjälp av en lämplig kryptering.

Krypteringsproblemet är mycket mera komplicerat matematiskt än rent tekniskt. Situationen kan beskrivas så att det finns två mängder: mängden av meddelanden X och mängden av deras krypterade motsvarigheter Y. Det finns två funktioner:

$$E : X \longrightarrow Y \quad \text{och} \quad D : Y \longrightarrow X.$$

Den första E är krypteringsfunktionen, och den andra, D är dekrypteringsfunktionen. E krypterar x till $E(x)$, och D dekrypterar $E(x)$ till x, dvs $(D \circ E)(x) = D(E(x)) = x$ (E och D är varandras inverser). Problemet är att konstruera E och D på ett sådant sätt att E är relativt enkel att definiera, och D är mycket svår att rekonstruera av den som inte har tillgång till dess definition.

Mera formellt kan X betraktas som mängden av informationsvektorer $\mathbf{a} = a_1a_2 \ldots a_n$, där a_i tillhör en ring R (t ex $\mathbb{Z}_2$, dvs $a_i = 0$ eller 1). En krypteringsfunktion ersätter $\mathbf{a}$ med en vektor $\mathbf{a}' = a_1'a_2' \ldots a_m'$ tillhörande mängden Y. Man kan inte konstruera $\mathbf{a}'$ helt slumpmässigt därför att det måste finnas ett effektivt sätt att få tillbaka $\mathbf{a}$. Samtidigt måste övergången från $\mathbf{a}'$ till $\mathbf{a}$ vara komplicerad för att göra det mycket svårt för obehöriga att komma åt $\mathbf{a}$.

Låt oss börja med ett exempel som är drygt 2000 år gammalt:

(14.1) Exempel. (**Caesarkrypto**[*]) Låt oss numrera alla bokstäver (för enkelhets skull i det engelska alfabetet) med 0, 1, 2, . . . , 25:

[*]Detta krypto användes av Julius Caesar.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Caesarkryptot är definierat som en funktion $E(x) = x + a$, där x, $a \in \mathbb{Z}_{26}$ (dvs man adderar modulo 26). Tag t ex $a = 3$. Då är $E(x) = x+3$ så att $E(0) = 3$, $E(1) = 4$, $E(24) = 2, \ldots$, dvs A krypteras till D, B till E, Y till C osv. Ordet MATEMATIK förvandlas till PDWHPDWLN.

Dekrypteringen är mycket enkel. Man kan använda dekrypteringsfunktionen $D(x) = x + 23$ därför att D är inversen till E dvs:

$$x \longmapsto x + 3 \longmapsto (x + 23) + 3 = x.$$

Med andra ord $D(E(x)) = x$ vilket följer ur likheten $23 + 3 = 0$. Till exempel dekrypteras PDW, dvs 15,3,22, till $15 + 23 = 12$, $3 + 23 = 0$, $22 + 23 = 19$, dvs MAT. $\square$

Caesarkryptot är mycket enkelt och kan knappast uppfylla dagens krav på säkra krypteringssystem. Men själva krypteringsmetoden visar vikten av en algebraisk struktur vid konstruktioner av krypterings- och dekrypteringsfunktioner.

Vi skall beskriva några krypteringstekniker som bygger på restaritmetiker, dvs grupper och ringar relaterade till $\mathbb{Z}_n$. För att kunna göra det behöver vi två mycket berömda satser ur talteorin.

Vi vet redan att $\mathbb{Z}_n$ är en grupp med avseende på addition modulo n. Låt

$$\mathbb{Z}_n^* = \{r \in \mathbb{Z}_n : SGD(r,n) = 1\}.$$

T ex är $\mathbb{Z}_4^* = \{1,3\}$, $\mathbb{Z}_5^* = \{1,2,3,4\}$ och $\mathbb{Z}_6^* = \{1,5\}$. Vi vet att $\mathbb{Z}_n^*$ är grupper med avseende på multiplikation modulo n (se Prop. (5.5)). Ordningen $|\mathbb{Z}_n^*|$ betecknas med $\varphi(n)$. Funktionen $\varphi(n)$ kallas Eulers funktion. Man har alltså

$$\varphi(n) = \text{ antalet heltal } \quad r \quad \text{sådana att} \quad 0 \leq r < n \quad \text{och} \quad SGD(r,n) = 1.$$

Definitionen ger $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(5) = 4$, $\varphi(6) = 2$, $\varphi(7) = 6$, $\varphi(8) = 4$, $\varphi(9) = 6$, $\varphi(10) = 4$ osv, jfr Prop (5.8).

(14.2) Eulers sats[†]. *Låt a och $n \geq 1$ vara heltal sådana att $SGD(a,n) = 1$. Då gäller*

$$a^{\varphi(n)} \equiv 1 \ (mod \ \ n).$$

Bevis. $\varphi(n)$ är ordningen av gruppen $\mathbb{Z}_n^*$. För varje element r i denna grupp gäller alltså $r^{\varphi(n)} = 1$. Villkoret $SGD(a,n) = 1$ säger att $a \equiv r \pmod{n}$ för något $r \in \mathbb{Z}_n^*$. Alltså är $a^{\varphi(n)} \equiv r^{\varphi(n)} \equiv 1 \pmod{n}$. □

Ett mycket viktigt specialfall av Eulers sats får man då $n = p$ är ett primtal. I sådant fall är $\varphi(p) = p - 1$ ty $\mathbb{Z}_p^* = \{1,\ 2,\ \ldots,\ p-1\}$ (alla element r i $\mathbb{Z}_p$ med undantag av 0 uppfyller $SGD(r,n) = 1$).

(14.3) Fermats lilla sats[‡]. *Låt a vara ett heltal och p ett primtal. Då är*

$$a^p \equiv a \ (mod \ \ p).$$

Bevis. Om $p \not| a$ (dvs $SGD(a,p) = 1$) så är $p | a^{p-1} - 1$ enligt Eulers sats. Alltså gäller $p | a^p - a = a(a^{p-1} - 1)$ både då $p \not| a$ och $p | a$. □

För våra tillämpningar behöver vi en generalisering av Fermats lilla sats:

(14.4) Sats. *Låt $n = p_1 p_2 \cdots p_k$ vara en produkt av olika primtal. Då är*

$$a^{\varphi(n)+1} \equiv a \ (mod \ \ n).$$

Bevis. Vi har $\varphi(n) = (p_1 - 1)(p_2 - 1) \ldots (p_k - 1)$ enligt Proposition (5.8). Om $p_i \not| a$ så är

$$a^{\varphi(n)} - 1 = (a^{\frac{\varphi(n)}{p_i - 1}})^{p_i - 1} - 1$$

delbart med p_i enligt Eulers sats ty $p_i \not| a^{\frac{\varphi(n)}{p_i-1}}$. Alltså är $p_i | a(a^{\varphi(n)} - 1) = a^{\varphi(n)+1} - a$ både då $p_i | a$ och $p_i \not| a$. Men $p_1,\ p_2,\ \ldots,\ p_k$ är olika primtal så att $n = p_1 p_2 \cdots p_k | a^{\varphi(n)+1} - a$. □

Nu kan vi diskutera den mest kända av alla krypteringsmetoder, vilken kallas RSA-krypteringssystem. RSA kommer från namnen Rivest, Shamir, Adleman. Dessa matematiker publicerade systemet 1978. Grunden för RSA-systemet är följande sats:

[†]Leonard Euler 1707 - 1783.
[‡]Pierre Fermat 1601-1665.

(14.5) Sats. *Låt $n = p_1 p_2 \cdots p_k$ där p_i är olika primtal. Låt e vara ett positivt heltal sådant att $SGD(e, \varphi(n)) = 1$ och låt d uppfylla kongruensen $ed \equiv 1 \pmod{\varphi(n)}$. Då har funktionen:*

$$E : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n \ ,$$

där $E(r) = r^e$, inversen

$$D : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n \ ,$$

där $D(r) = r^d$.

Bevis. Vi vet att förutsättningen $SGD(e, \varphi(n)) = 1$ garanterar att d existerar ty $\mathbb{Z}^*_{\varphi(n)}$ är en grupp. För att visa att D är inversen till E kontrollerar man att $D \circ E$ är identiteten på $\mathbb{Z}_n$ dvs $(D \circ E)(r) = r^{ed} = r$. Men $ed = 1 + q \cdot \varphi(n)$, $q \geq 1$ så att:

$$(D \circ E)(r) = r^{1+q\varphi(n)}.$$

Vi visar induktivt att $r^{1+q\varphi(n)} = r$. Likheten gäller då $q = 1$ enligt (14.4). Antag att

$$r^{1+(q-1)\varphi(n)} = r$$

i $\mathbb{Z}_n$. Då är

$$r^{1+q\varphi(n)} = r^{1+\varphi(n)+(q-1)\varphi(n)} = r^{1+\varphi(n)} r^{(q-1)\varphi(n)} = r \cdot r^{(q-1)\varphi(n)} = r^{1+(q-1)\varphi(n)} = r.$$

Alltså gäller likheten $r^{1+q\varphi(n)} = r$ för alla q dvs $(D \circ E)(r) = r$ då $r \in \mathbb{Z}_n$. □

(14.6) RSA-kryptering. Vi ger en beskrivning av metoden enbart då $n = pq$ är en produkt av två olika primtal. Metoden fungerar på samma sätt då n är produkt av ett godtyckligt antal olika primtal.

(a) Välj två olika primtal p, q och beräkna $n = pq$ (p, q är vanligen mycket stora, säg av storleksordningen 10^{100}).

(b) Beräkna $\varphi(n) = (p-1)(q-1)$ och välj e så att $SGD(e, \varphi(n)) = 1$. Beräkna även d så att $ed \equiv 1 \pmod{\varphi(n)}$($d$ räknas med hjälp av Euklides algoritm – se kapitlet “Delbarhet och primtal”).

(c) Publicera n, e och en “ordbok" för översättning av meddelanden till $r \in \mathbb{Z}_n$ (t ex $A = 10$, $B = 11$, …, $Z = 35$ då $n > 35$).

(d) Den som vill sända meddelanden till Dig krypterar med hjälp av (den kända) funktionen $E(r) = r^e$. Du är den ende (förhoppningsvis) som kan dekryptera med hjälp av funktionen $D(r) = r^d$(d är hemligt och $(D \circ E)(r) = D(r^e) = r^{ed} = r$ enligt (14.5)).

(14.7) Exempel. Klartext ALGEBRA kodat med $A = 10$, $B = 11$, ..., $Z = 35$ är

$$1021,\ 1614,\ 1127,\ 10.$$

Låt $n = pq = 47 \cdot 167 = 7849$. Då är $\varphi(n) = 46 \cdot 166 = 7636$. Låt oss välja $e = 29$ ($SGD(29, 46 \cdot 166) = 1$). Då är $e^{-1} = 29^{-1} = 4213$ (i $\mathbb{Z}^*_{\varphi(n)}$). Kryptering enligt RSA-metoden ger:

$$1178,\ 1929,\ 3383,\ 4578,$$

dvs $1021^{29} \equiv 1178 \pmod{7849}$ osv. Vid dekryptering räknar man: $1178^{4213} \equiv 1021 \pmod{7849}$ osv. □

(14.8) Anmärkning. RSA-systemet tillhör s.k. öppen-nyckelkrypton dvs krypteringsfunktionen E är allmänt känd. Vad gör den som vill beräkna inversen $D = E^{-1}$? För att beräkna d måste man lösa ekvationen $ed \equiv 1 \pmod{\varphi(n)}$. För att göra det måste man känna $\varphi(n)$. För att beräkna $\varphi(n) = (p-1)(q-1)$ måste man (med all sannolikhet) känna p och q. Men för att få p och q ur $n = pq$ (som är känt) måste man faktorisera n. Faktoriseringsproblemet är mycket svårt att lösa. De bästa kända algoritmerna för primtalsfaktorisering av ett heltal n kräver c:a $n^{1/5}$ räkneoperationer om man vill hitta en primfaktor till n. Om $p, q \approx 10^{100}$ så är $n \approx 10^{200}$. Om en räkneoperation tar $1\mu s$ så krävs det $10^{40}\mu s \approx 3 \cdot 10^{26}$ år för att genomföra beräkningarna för n (10^6 datorer var och en kapabel att utföra en räkneoperation på $1\mu s$ skulle behöva $3 \cdot 10^{20}$ år för att klara dessa beräkningar!). Men det finns inte något bevis att faktororiseringsproblemet är så pass svårt. Det är alltså möjligt att det finns bättre algoritmer som inte är kända nu. Å andra sidan ökar datorernas beräkningskapacitet dramatiskt och redan nu är man mycket försiktig med valet av p och q [§]. □

(14.9) Anmärkning. RSA-systemet kan även användas för äkthetskontroll. Den som känner D (se (14.6)) kan signera dokument med $D(r) = r^d$. Den som vill kontrollera äktheten av signaturen räknar ut $E \circ D(r) = r^{de} = r$ (E är allmänt känd). □

Vi skall beskriva några andra krypteringsmetoder.

(14.10) Hillkryptot [¶]. Låt $R = \mathbb{Z}_n$ och låt H vara en $(N \times N)-$matris vars element tillhör $\mathbb{Z}_n$ och sådan att $\det(H) \in \mathbb{Z}^*_n$. En sådan matris har invers H^{-1} (den kan beräknas på samma sätt som inversen till en reell matris). Låt

$$E : \mathbb{Z}_n^N \longrightarrow \mathbb{Z}_n^N$$

[§] Nyligen visades att man ibland kan forcera RSA-kryptot i sådana fall som tidigare uppfattades som omöjliga att klara.

[¶] L.S. Hill publicerade sina arbeten om detta krypto 1929-1931.

där $E(x) = Hx$ för $x = (x_1, \ldots, x_N)^t \in \mathbb{Z}_n^N$. E är en isomorfism eftersom E har inversen

$$D : \mathbb{Z}_n^N \longrightarrow \mathbb{Z}_n^N$$

där $D(x) = H^{-1}x$ (dvs $(E \circ D)(x) = E(Hx) = H^{-1}Hx = x$).

Låt oss betrakta ett exempel då $n = 26$ och

$$E : \mathbb{Z}_{26}^2 \longrightarrow \mathbb{Z}_{26}^2,$$

där

$$E(\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]) = \left[\begin{smallmatrix} 2 & 1 \\ 23 & 24 \end{smallmatrix}\right] \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] = \left[\begin{smallmatrix} 2a+b \\ 23a+24b \end{smallmatrix}\right].$$

Genom att översätta $A = 0$, $B = 1$, $\ldots$, $Z = 25$ kan man kryptera par av bokstäver:

$$E(AL) = E(\left[\begin{smallmatrix} 0 \\ 11 \end{smallmatrix}\right]) = \left[\begin{smallmatrix} 2 & 1 \\ 23 & 24 \end{smallmatrix}\right] \left[\begin{smallmatrix} 0 \\ 11 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 11 \\ 4 \end{smallmatrix}\right] = LE.$$

Dekrypteringen sker med hjälp av $H^{-1} = H$ (kontrollera att $HH = I$!). T.ex.

$$D(LE) = D(\left[\begin{smallmatrix} 11 \\ 4 \end{smallmatrix}\right]) = \left[\begin{smallmatrix} 2 & 1 \\ 23 & 24 \end{smallmatrix}\right] \left[\begin{smallmatrix} 11 \\ 4 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 \\ 11 \end{smallmatrix}\right] = AL.$$

Ofta väljer man H just så att $H^{-1} = H$. Då kan H användas som både krypterings- och dekrypteringsnyckel. En matris H sådan att $H^2 = I$ (identitetsmatrisen) kallas involutiv (eller involutionsmatris). När $n = 26$ och $N = 2$ finns det 736 sådana matriser, men för $N = 3$ är deras antal 1 360 832. Det finns flera varianter av Hillkryptot. Se vidare övningar. □

(14.11) Merkle-Hellmans kappsäckskrypto[‖]. Låt $a_1, a_2, \ldots, a_n \in \mathbb{Z}_m$ och $w \in \mathbb{Z}_m^*$. Definiera

$$E_w : \mathbb{Z}_m^n \longrightarrow \mathbb{Z}_m$$

så att

$$E_w(\mathbf{x}) = x_1wa_1 + x_2wa_2 + \cdots + x_nwa_n = w(\mathbf{x} \cdot \mathbf{a}^t),$$

[‖]R.C. Merkle och M.E. Hellman publicerade sina arbeten om detta krypto 1979 - 1982. 1982 visade A. Shamir att säkerheten av detta krypto är dålig.

där $\mathbf{x} \cdot \mathbf{a}^t$ är skalärprodukten av $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_n) \in \mathbb{Z}_m^n$ och $\mathbf{a} = (a_1,\ a_2,\ \ldots,\ a_n)$. Avbildningen E_w är en grupphomomorfism ty

$$E_w(\mathbf{x}+\mathbf{y}) = w((\mathbf{x}+\mathbf{y})\cdot\mathbf{a}^t) = w(\mathbf{x}\cdot\mathbf{a}^t) + w(\mathbf{y}\cdot\mathbf{a}^t) = E_w(\mathbf{x}) + E_w(\mathbf{y}),$$

där $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_m^n$. Funktionen E_w är inte injektiv på $\mathbb{Z}_m^n$ (om $n > 1$) men om man lämpligt väljer $\mathbf{a}$ som s.k. ordnad kappsäck så är den injektiv för vektorer $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_n)$ sådana att $x_i = 0$ eller 1. Ett sådant val är t.ex. $a_i = 2^{i-1}$, $i = 1,\ \ldots,\ n$ och $m > 2^n$. Med ett hemligt val av $w \in \mathbb{Z}_m^*$ får man då ett Merkle-Hellmans kappsäckskrypto ($\mathbf{a} = (a_1, a_2, \ldots, a_n)$ är en ordnad kappsäck och $w\mathbf{a} = (wa_1, wa_2, \ldots, wa_n)$ är oordnad). Man dekrypterar med hjälp av $v \in \mathbb{Z}_m^*$, där $vw \equiv 1(\text{mod } m)$ ty $vE_w(\mathbf{x}) = vw\mathbf{x}\cdot\mathbf{a}^t = \mathbf{x}\cdot\mathbf{a}^t$ och $\mathbf{x}\cdot\mathbf{a}^t$ bestämmer entydigt $\mathbf{x}$. □

ÖVNINGAR

I alla övningar är $A = 1$, $B = 2$, . . . , $Z = 26$ och mellanrum = 27.

14.1. Låt $n = 3 \cdot 11 = 33$.

(a) Låt krypteringsnyckeln vara $e = 3$. Kryptera DISKRET MATEMATIK.

(b) Bestäm dekrypteringsnyckeln d.

(c) Dekryptera 19, 1, 4, 12, 26.

14.2. För att kontrollera äktheten av dokument som skickas från MATEMATIK AB använder man krypteringsnyckeln $n = 221$, $e = 7$ (känd för alla mottagare). Kontrollera äktheten av ett dokument med signaturen:

$$208,\ 1,\ 45,\ 112,\ 208,\ 1,\ 45,\ 76,\ 54.$$

14.3. Låt $H = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 5 \end{smallmatrix}\right]$ och låt $E : M_2(\mathbb{Z}_{28}) \longrightarrow M_2(\mathbb{Z}_{28})$ **, där $E(X) = HX$.

(a) Visa att E är en grupphomomorfism.

(b) Man definierar ett Hillkrypto så att $X = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ svarar mot 4 bokstäver a, b, c, d i en klartext. Kryptera TEXT och konstruera en dekrypteringsfunktion

$$D : M_2(\mathbb{Z}_{28}) \longrightarrow M_2(\mathbb{Z}_{28})$$

(inversen till E). Dekryptera: $\left[\begin{smallmatrix} 19 & 18 \\ 21 & 27 \end{smallmatrix}\right]$.

** $M_2(R)$ betecknar alla 2×2-matriser med element i R.

Kapitel 15

BOOLESKA ALGEBROR

(15.1) Definition. Men en Boolesk algebra menar man en mängd $\mathcal{B}$ med addition och multiplikation som mot $x, y \in \mathcal{B}$ ordnar deras summa $x + y \in \mathcal{B}$ och deras produkt $xy \in \mathcal{B}$ samt konjugering som mot $x \in \mathcal{B}$ ordnar dess konjugat $\bar{x} \in \mathcal{B}$, så att för varje x, y och $z \in \mathcal{B}$

(a)	$x + y = y + x;\ xy = yx,$	(kommutativitet)
(b)	$x + (y + z) = (x + y) + z;\ x(yz) = (xy)z,$	(associativitet)
(c)	$x(y + z) = xy + xz;\ x + yz = (x + y)(x + z),$	(distributivitet)
(d)	Det finns element $0, 1 \in \mathcal{B}$ sådana att $x + 0 = x;\ x1 = x,$ $x + 1 = 1;\ x0 = 0,$ $x + \bar{x} = 1;\ x\bar{x} = 0,$	
(e)	$\overline{x + y} = \bar{x}\bar{y};\ \overline{xy} = \bar{x} + \bar{y},$	(de Morgans lagar)
(f)	$\bar{\bar{x}} = x$	(involution)

□

(15.2) Exempel. (a) Den enklaste Booleska algebran (av intresse) består av enbart två element: $\mathcal{B} = \{0, 1\}$ med addition, multiplikation och konjugat som definieras på följande sätt:

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

och $\bar{0} = 1, \bar{1} = 0$. Den algebran kommer vi att beteckna med $\mathbb{B}$. Det finns en ännu enklare Boolesk algebra: $\mathcal{B} = \{0\}$ med $0 + 0 = 0, 00 = 0$ och $\bar{0} = 0$ (här är $0 = 1$). Den kallas trivial och är helt ointressant.

(b) Låt U vara en fixerad mängd (universum) och låt $\mathcal{B}$ bestå av alla delmängder till U. Vi definierar addition i $\mathcal{B}$ som unionen $A \cup B$, multiplikation i $\mathcal{B}$ som snittet $A \cap B$ och konjugat i $\mathcal{B}$ som komplementet $\bar{A}$. Vi får då en Boolesk algebra om 0 betecknar den tomma mängden

$\emptyset$ och 1 hela mängden U. Det faktum att alla villkor (a) – (e) är uppfyllda följer direkt och enkelt från räknelagar för mängder.

(c) Om X är en mängd och $\mathcal{B}$ är en Boolesk algebra så är det möjligt att konstruera en ny Boolesk algebra som består av alla funktioner $f : X \to \mathcal{B}$. Sådana funktioner kan man addera och multiplicera:

$$(f+g)(x) = f(x) + g(x), \; (fg)(x) = f(x)g(x)$$

samt bilda konjugatet $\bar{f}$ genom att definiera $\bar{f}(x) = \overline{f(x)}$. Lägg märke till att funktionsvärden $f(x)$ och $g(x)$ adderas, multipliceras och konjugeras i $\mathcal{B}$.

Låt oss betrakta ett mycket viktigt specialfall. Om $X = \mathbb{Z}_2^2$ är mängden av alla (binära) vektorer $00, 01, 10, 11$ och $\mathbb{B}$ den Booleska algebran ur exempel (15.2) (a), så ordnar varje funktion $f : \mathbb{Z}_2^2 \to \mathbb{B}$ antingen 0 eller 1 mot varje vektor x_1x_2, där $x_i \in \{0, 1\}$. Vi känner flera sådana funktioner: AND, OR, NAND, NOR osv. Om vi t.ex. tar $f =$ AND och $g =$ OR så har vi funktionstabellerna:

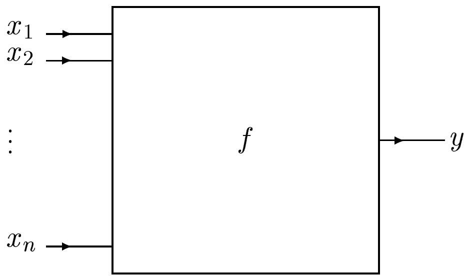
x_1	x_2	$f(x_1, x_2)$	$g(x_1, x_2)$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

För dessa funktioner kan vi bilda $f + g, fg$ och $\bar{f}, \bar{g}$. Då är

x_1	x_2	$(f+g)(x_1, x_2)$	$(fg)(x_1, x_2)$	$\bar{f}(x_1, x_2)$	$\bar{g}(x_1, x_2)$
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	1	1	0	0

Vi ser att $\bar{f} =$ NAND och $\bar{g} =$ NOR.

(15.3) Definition. Om $\mathbb{Z}_2^n =$ mängden av alla binära vektorer $\mathbb{x} = x_1x_2 \ldots x_n$ av längden n och $\mathbb{B}$ är algebran ur exempel (15.2) (a) så kallar vi varje funktion $f : \mathbb{Z}_2^n \to \mathbb{B}$ för en **transmissionsfunktion** eller en **Boolesk funktion**. Den kan tolkas som en grind med n ingångar $x_1, x_2, \ldots, x_n$ och en utgång y:



Mot varje uppsättning av signaler $x_1, x_2, \dots, x_n$ dvs mot varje binär vektor $\mathbb{x} = x_1 x_2 \dots x_n$ svarar en signal $y \in \mathbb{B}$ dvs $y = 0$ eller 1. Den Booleska algebran av alla funktioner $f : \mathbb{Z}_2^n \to \mathbb{B}$ kommer att betecknas med $\mathcal{F}_n$. □

(d)[*] Låt X vara mängden av 3 variabler; P, Q, R. Ur dessa tre variabler kan man bilda olika uttryck genom att använda $\wedge$ (konjunktion), $\vee$ (disjunktion) och $\neg$ (negation). Vi placerar parenteser för att visa i vilken ordning tillämpar vi dessa operationer. På så sätt får vi t.ex. $P \wedge Q, (P \wedge Q) \vee R, \neg(P \wedge Q) \vee R$ osv. Här kan vi känna igen induktionsmetoden: Vi har ett alfabet P, Q, R och vi konstruerar de korrekta uttrycken med hjälp av $\wedge, \vee, \neg$ (P, Q, R är vår bas; om U_1 och U_2 är välkonstruerade så är också $U_1 \wedge U_2, U_1 \vee U_2, \neg U_1, \neg U_2$). I mängden av alla så konstruerade uttryck identifierar vi nu sådana som har exakt samma logiska värden för varje val av de logiska värden 0, 1 för P, Q, R. T.ex. skall $P \wedge Q$ och $Q \wedge P$ identifieras, $\neg(P \wedge Q)$ och $\neg P \vee \neg Q$ likaså. Vi skall skriva $P \wedge Q = Q \wedge P$, $\neg(P \wedge Q) = \neg P \vee \neg Q$. Uttrycket $P \wedge \neg P$ ger alltid värdet 0. Vi skall skriva $P \wedge \neg P = 0$. På liknande sätt är $P \vee \neg P = 1$. Nu kan vi bilda en Boolesk algebra $\mathcal{B}$. Den består av alla korrekta uttryck ur P, Q, R som vi identifierar i enlighet med vår konstruktion. I $\mathcal{B}$ adderar, multiplicerar och konjugerar vi med hjälp av $\wedge, \vee$ och $\neg$. 1 betecknar alla tautologier och 0 alla kontradiktioner. Detta exempel visar på sambandet mellan Booleska algebror och satskalkylen. Man behöver inte starta med just tre variabler P, Q, R – man kan ta ett godtyckligt antal variabler $P_1, P_2, \dots, P_n$ och upprepa hela konstruktionen. □

Vi skall koncentrera oss på exempel (15.2) (c) och transmissionsfunktioner. Innan vi gör det skall vi analysera definitionen av Booleska algebror. Valet av axiomen (a) – (f) är ett av många möjliga. Dessa räknelagar möjliggör räkneoperationer och implicerar andra räknelagar. T.ex. $x + x = x$, $x + xy = x$, $x^2 = x$, $x(x + y) = x$ osv. Sådana formler kan man härleda ur (a) – (f).

(15.4) Exempel. Vi skall visa att $x + x = x$ för varje element x i en Boolesk algebra. Vi har:

$$x + x = x1 + x1 = x(1 + 1) = x1 = x,$$

där den första likheten gäller enligt (15.1) (d), den andra enligt (15.1) (c), och den tredje och fjärde enligt (15.1) (d). □

[*]Detta exempel är litet svårare och man kan hoppa över det.

(15.5) Exempel. Visa att $x + xy = x$. Vi har:

$$x + xy = x1 + xy = x(1 + y) = x1 = x,$$

där den första likheten gäller enligt (15.1) (d), den andra enligt (15.1) (c), och den tredje och fjärde enligt (15.1) (d). □

I själva verket är det t.o.m. möjligt att härleda en del av villkoren (a) – (f) ur de övriga. Vi har valt en sådan uppsättning av räknelagar för att inte ägna för mycket tid åt att "leka med axiomen" som i de två senaste exemplen. Det är ett intressant problem hur man kan begränsa antalet räknelagar i definitionen och samtidigt kunna härleda dem som har utelämnats. Detta är dock ett mera teoretiskt än praktiskt problem. Men det finns en mycket viktig praktisk aspekt av Definitionen (15.1). Den kallas **dualitetsprincipen för Booleska algebror** och baseras på observationen att uppsättningen av villkoren (a) – (f) övergår i samma uppsättning då man byter addition mot multiplikation, multiplikation mot addition, 0 mot 1 och 1 mot 0. Som en konsekvens av detta får vi att varje formel som gäller i en Boolesk algebra fortfarande gäller då man gör dessa byten. Detta är just dualitetsprincipen.

(15.6) Exempel. Vi visade i exempel (15.4) att $x + x = x$. Alltså är det också sant att $xx = x$ (xx betecknas med x^2 dvs $x^2 = x$). I exempel (15.5) visade vi att $x + xy = x$. Alltså gäller även att $x(x + y) = x$. □

De viktigaste tillämpningarna av Booleska algebror kommer ur sambandet mellan deras räknelagar och principer för konstruktioner av digitala kretsar. För att kunna dra full nytta av sambandet måste vi bevisa en sats om Booleska funktioner. Innan vi gör det, låt oss fixera beteckningarna. Om $\mathcal{B}$ är en Boolesk algebra och $x \in \mathcal{B}$ så skall vi skriva

$$x^1 = x \quad \text{och} \quad x^0 = \bar{x}.$$

En viktig egenskap hos sådana potenser är följande:

$$i^j = \begin{cases} 1 & \text{om} \quad i = j, \\ 0 & \text{om} \quad i \neq j, \end{cases}$$

(kontrollera likheter i 4 möjliga fall: $0^0, 0^1, 1^0, 1^1$!).

(15.7) Sats. *Varje Boolesk funktion $f(x_1, x_2, \ldots, x_n)$ kan entydigt skrivas på formen*

$$f(x_1, x_2, \ldots, x_n) = \sum_{i_1, i_2, \ldots, i_n} a_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},$$

där man summerar över alla binära vektorer $i_1 i_2 \ldots i_n$ och $a_{i_1 i_2 \ldots i_n} = 0$ eller 1. I den enda framställningen av f är

$$a_{i_1 i_2 \ldots i_n} = f(i_1, i_2, \ldots, i_n)$$

Bevis. Låt oss först konstatera att

$$j_1^{i_1} j_2^{i_2} \dots j_n^{i_n} = \begin{cases} 1 & \text{om} \quad j_1 = i_1 \quad \text{och} \quad j_2 = i_2 \quad \text{och} \quad \dots \text{ och } j_n = i_n, \\ 0 & \text{om} \quad j_1 \neq i_1 \quad \text{eller} \quad j_2 \neq i_2 \quad \text{eller} \quad \dots \text{ eller } j_n \neq i_n. \end{cases}$$

Betrakta nu funktionen

$$g(x_1, x_2, \dots, x_n) = \sum_{i_1, i_2, \dots, i_n} a_{i_1 i_2 \dots i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}.$$

Om $j_1 j_2 \dots j_n$ är en binär vektor så är

$$g(j_1, j_2, \dots, j_n) = \sum_{i_1, i_2, \dots, i_n} a_{i_1 i_2 \dots i_n} j_1^{i_1} j_2^{i_2} \dots j_n^{i_n} = a_{j_1 j_2 \dots j_n}$$

Detta betyder att $f = g$ dvs för varje $j_1 j_2 \dots j_n$ är $f(j_1, j_2, \dots, j_n) = g(j_1, j_n, \dots, j_n)$ då och endast då $f(j_1, j_2, \dots, j_n) = a_{j_1 j_2 \dots j_n}$. Alltså implicerar likheten $f(i_1, i_2 \dots i_n) = a_{i_1 i_2 \dots i_n}$ att

$$g(x_1, x_2, \dots x_n) = \sum f(i_1, i_2, \dots, i_n) x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} = f(x_1, x_2, \dots, x_n)$$

och omvänt, om $f = g$ så måste $a_{i_1 i_2, \dots i_n} = f(i_1, i_2, \dots, i_n)$. □

Funktionerna $x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$ kallas ibland **mintermer** eller **minimala Booleska funktioner** – de antar värdet 1 endast om $(x_1, x_2, \dots, x_n) = (i_1, i_2, \dots, i_n)$.

Framställningen av f ur sats (15.7) kallar man för den **disjunktiva normalformen av** f (det finns också en konjunktiv normalform – se Övning (15.9)).

Nu skall vi svara på frågan hur man kan bestämma den disjunktiva normalformen av en given Boolesk funktion.

(15.8) Metod 1. Låt $f(x_1, x_2, \dots, x_n)$ vara en Boolesk funktion. Man räknar ut alla värden $f(i_1, i_2, \dots, i_n)$ för alla binära vektorer $i_1 i_2 \dots i_n$. Därefter får man

$$f(x_1, x_2, \dots, x_n) = \sum f(i_1, i_2, \dots, i_n) x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$$

i enlighet med sats (15.7). Den metoden kan vara mycket arbetsam – man måste beräkna q^n värden $f(i_1, i_2, \dots i_n)$. Låt t.ex. $f(x_1, x_2, x_3) = x_1 x_2 + x_2 \bar{x}_3$. Betrakta tabellen:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	0	
0	1	0	1	$\mapsto \bar{x}_1 x_2 \bar{x}_3$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$\mapsto x_1 x_2 \bar{x}_3$
1	1	1	1	$\mapsto x_1 x_2 x_3$

Alltså är

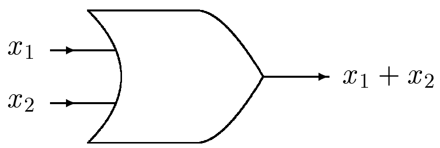
$$f(x_1, x_2, x_3) = \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$$

(15.9) Metod 2. Vi exemplifierar först metoden med hjälp av samma funktion f som i Metod 1. $f(x_1, x_2, x_3)$ är summan av två termer: $x_1 x_2$ och $x_2 \bar{x}_3$. Vi vill ha en summa av mintermer. För att få sådana termer multiplicerar vi $x_1 x_2$ med $x_3 + \bar{x}_3 = 1$ och $x_2 \bar{x}_3$ med $x_1 + \bar{x}_1 = 1$ eftersom $x_1 x_2$ saknar x_3 och $x_2 \bar{x}_3$ saknar x_1. Vi får:

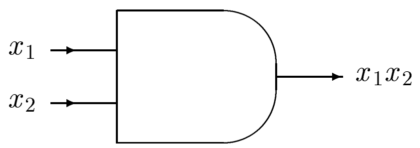
$$\begin{aligned} x_1 x_2 (x_3 + \bar{x}_3) &= x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 \\ (x_1 + \bar{x}_1) x_2 \bar{x}_3 &= x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 \end{aligned}$$

Nu är $f(x_1, x_2, x_3) = x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3$ (ty $x_1 x_2 \bar{x}_3 + x_1 x_2 \bar{x}_3 = x_1 x_2 \bar{x}_3$) och uppgiften är löst. Allmänt: om f innehåller en term som är produkt av en del av $x_1, x_2, \ldots, x_n, \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ så multiplicerar vi den termen med $x_i + \bar{x}_i = 1$ för varje index i som inte är representerat. Därefter eliminerar vi de mintermer $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ som vi redan har (ty $a + a = a$ i en Boolesk algebra). Proceduren upprepar vi för varje term i f som inte är en minterm.

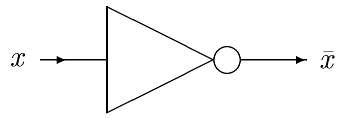
Som en viktig tillämpning av Sats (15.7) skall vi visa att varje Boolesk funktion $y = f(x_1, x_2, \ldots, x_n)$ kan realiseras med hjälp av en krets som är uppbyggd av ett antal grindar OR, AND och NOT. Standardsymbolerna för dessa gindar är:



OR-grind

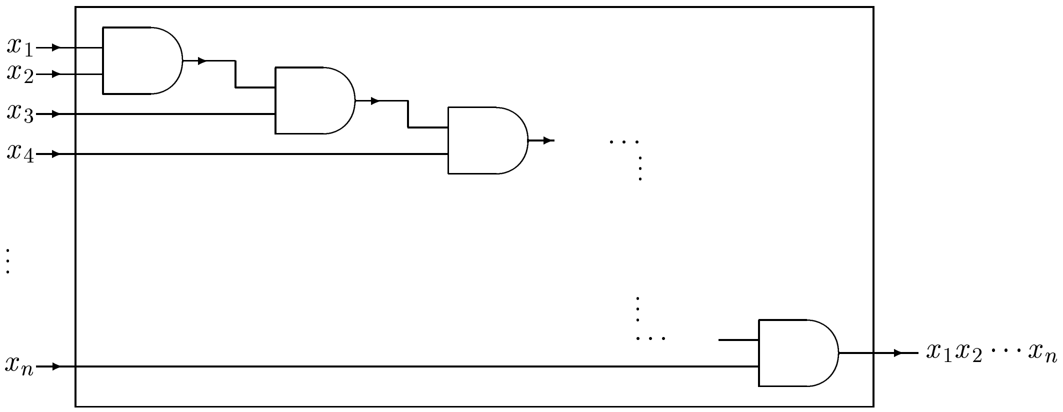


AND-grind

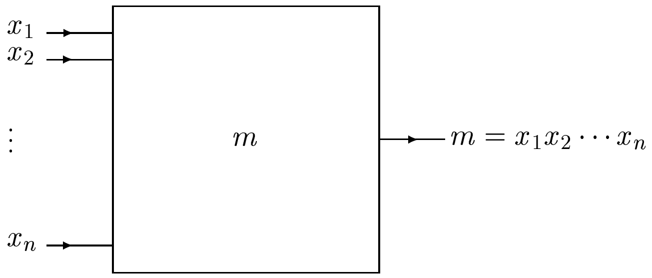


NOT-grind

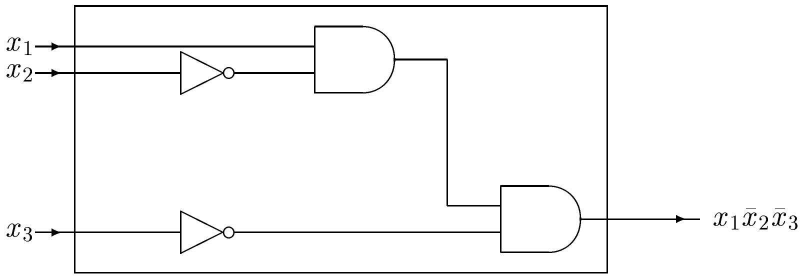
Först har vi:



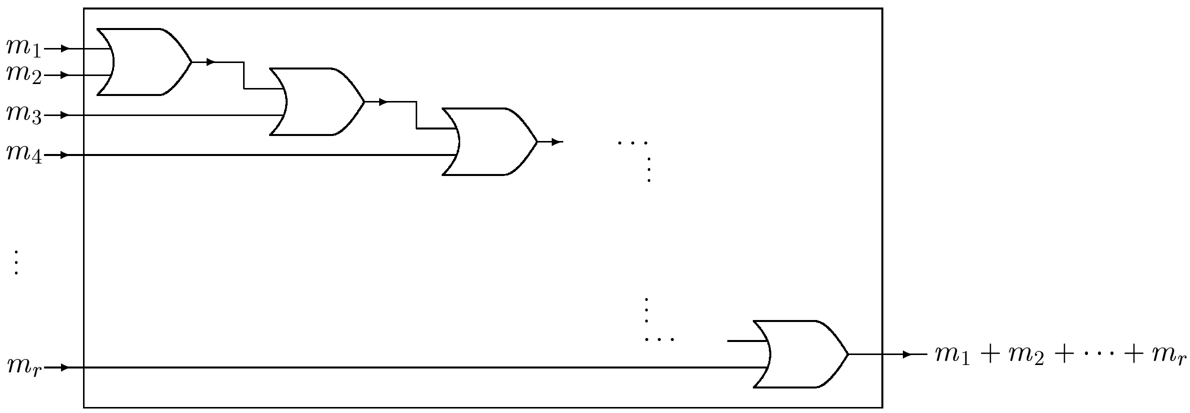
Vi skall beteckna den kretsen kortare:



Om vi vill realisera en godtycklig minterm $x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}$ och t.ex. $i_k = 0$, dvs mintermen innehåller faktorn $\bar{x}_k$, så måste vi placera NOT före ingången av x_k till OR t.ex. $x_1\bar{x}_2\bar{x}_3$ ges av



På det sättet får vi att varje minterm $x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}$ kan realiseras. Nu är $f(x_1, x_2, \ldots, x_n) = \sum f(i_1, i_2, \ldots, i_n)x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n} = m_1 + m_2 + \ldots + m_r$ en summa av olika mintermer m_1, m_2, $\ldots$, m_r. En sådan summa realiseras av



och vårt påstående är bevisat. Senare i övningar kommer vi att syssla med andra möjliga uppsättningar av enkla funktioner som är fullständiga i den meningen att varje Boolesk funktion kan realiseras med hjälp av dem. Här har vi visat att AND, OR och NOT ger en **fullständig uppsättning** – varje Boolesk funktion kan realiseras med hjälp av dessa tre.

Ett viktigt praktiskt problem är att för en given Boolesk funktion f bestämma en framställning som innehåller så få produkttermer som möjligt och så få faktorer i dessa termer som möjligt. Det finns olika algoritmer som man tillämpar i detta syfte. Vi skall diskutera en av dem: **Quine-McCluskey's minimeringsalgoritm**.

(15.10) Definition. Låt $m = x_1^{i_1}x_2^{i_2}\dots x_n^{i_n}$ vara en minterm. Man säger att en funktion p är en **delprodukt** till m om p är en produkt av vissa av potenserna $x_1^{i_1}, x_2^{i_2}, \dots, x_n^{i_n}$. □

(15.11) Exempel. $m = x_1\bar{x}_2\bar{x}_3x_4$. Då är t.ex. $x_1\bar{x}_2, x_1\bar{x}_3x_4, \bar{x}_3x_4$ osv delprodukter till m. □

(15.12) Definition. Låt f vara en Boolesk funktion. En **implikant** till f är en delprodukt p av en minterm för f sådan att om $p(\mathbb{x}) = 1$ så är $f(\mathbb{x}) = 1$, där $\mathbb{x}$ betecknar $x_1, x_2, \dots, x_n$. □

(15.13) Exempel. Låt $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3$. Då är $x_1x_2x_3$ en implikant (varje minterm är en implikant!) Även x_1x_3 är en implikant, ty om $x_1x_3 = 1$ så är $x_1 = 1$, $x_3 = 1$. Då är $f(x_1, x_2, x_3) = 1x_21 + 1\bar{x}_21 + 0\bar{x}_20 = x_2 + \bar{x}_2 = 1$, dvs $f(x_1, x_2, x_3) = 1$ om $x_1x_3 = 1$. □

(15.14) Definition. En implikant p till f kallas **prim** om ingen äkta delprodukt av p är en implikant till f. □

(15.15) Exempel. I exempel (15.13) är $x_1x_2x_3$ en implikant till f. Den är inte prim ty x_1x_3 är dess delprodukt som också är en implikant till f. Men x_1x_3 är en primimplikant ty varken x_1 eller x_3 är implikanter till f (t.ex. om $x_1 = 1$, så kan $f(1, x_2, x_3)$ vara 0; tag $x_2 = x_3 = 0$). □

(15.16) Sats. *Varje Boolesk funktion är en summa av sina primimplikanter.*

Bevis. Låt f vara en Boolesk funktion. Låt g vara summan av alla primimplikanter till f. Om $g(\mathbb{x}) = 1$, så antar någon primimplikant värdet 1, vilket ger att $f(\mathbb{x}) = 1$. Omvänt, om $f(\mathbb{x}) = 1$, så måste en minterm m anta värdet 1 dvs $m(\mathbb{x}) = 1$. En minterm är en implikant. Om den inte är en primimplikant så kan man stryka någon (eller några) av dess faktorer så att man får en primimplikant. Alltså finns det en primimplikant p till f sådan att $p(\mathbb{x}) = 1$. Då antar g värdet 1 ty p ingår i g som en term. □

(15.17) Hur hittar man alla primimplikanter enligt Quine-McCluskey's metod?

Vi skall exemplifiera metoden med hjälp av en Boolesk funktion f. Först skriver vi ut den disjunktiva normalformen för f. Låt

$$\begin{aligned} f(x_1,x_2,x_3,x_4) &= \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4+\bar{x}_1\bar{x}_2\bar{x}_3x_4+\bar{x}_1\bar{x}_2x_3\bar{x}_4+x_1\bar{x}_2\bar{x}_3\bar{x}_4+ \\ &\quad \bar{x}_1\bar{x}_2x_3x_4+x_1x_2\bar{x}_3\bar{x}_4+\bar{x}_1x_2x_3x_4 \end{aligned}$$

För att beteckna en implikant skriver vi kortare $\to i_1i_2\dots i_n$, där i_k är alla exponenter av variablerna i implikanten. Här har vi 7 implikanter av längden 4:

$$\begin{array}{l} \to 0000 \\ \to 0001 \\ \to 0010 \\ \to 1000 \\ \to 0011 \\ \to 1100 \\ \to 0111 \end{array}$$

Nu bestämmer vi alla implikanter av längden 3. p är en sådan implikant om man kan förlänga p med en variabel x_i som inte ingår i p så att både px_i och $p\bar{x}_i$ är implikanter av längden 4. Vi jämför 0000 med alla produkter (6 stycken) och bestämmer de som skiljer sig från 0000 på exakt ett ställe som betecknas med "-":

$$\begin{array}{ccc} 0000 & 0000 & 0000 \\ 0001 & 0010 & 1000 \\ \hline 000\text{-} & 00\text{-}0 & \text{-}000 \end{array}$$

Alltså är $\bar{x}_1\bar{x}_2\bar{x}_3, \bar{x}_1\bar{x}_2\bar{x}_4$ och $\bar{x}_2\bar{x}_3\bar{x}_4$ implikanter av längden 3.

Nu upprepar vi samma procedur med 0001 och sedan med alla andra implikanter av längden 4. Vi får då samtliga implikanter av längden 3:

$$\begin{array}{rl} \to & 000\text{-} \\ \to & 00\text{-}0 \\ & \text{-}000 \\ \to & 00\text{-}1 \\ \to & 001\text{-} \\ & 1\text{-}00 \\ & 0\text{-}11 \end{array}$$

dvs $\bar{x}_1\bar{x}_2\bar{x}_3$, $\bar{x}_1\bar{x}_2\bar{x}_4$, $\bar{x}_2\bar{x}_3\bar{x}_4$, $\bar{x}_1\bar{x}_2x_4$, $\bar{x}_1\bar{x}_2x_4$, $x_1\bar{x}_3\bar{x}_4$ och $\bar{x}_1x_3x_4$.

Nu bestämmer vi alla implikanter av längden 2 genom att undersöka alla par ur implikanter av längden 3 och välja ut sådana som skiljer sig på exakt ett ställe. Dessa är:

$$\begin{array}{c} 000- \\ 001- \\ \hline 00-- \end{array} \qquad \begin{array}{c} 00-0 \\ 00-1 \\ \hline 00-- \end{array}$$

Det finns alltså enbart en implikant av längden 2:

$$00--$$

dvs $\bar{x}_1\bar{x}_2$. Vilka implikanter är primimplikanter? Det är sådana som inte har andra implikanter som äkta delprodukter. Det är

$$00--$$

dvs $\bar{x}_1\bar{x}_2$ av längden 2 och

$$\begin{array}{lll} -000 & \text{dvs} & \bar{x}_2\bar{x}_3\bar{x}_4 \\ 1-00 & \text{dvs} & x_1\bar{x}_3\bar{x}_4 \\ 0-11 & \text{dvs} & \bar{x}_1x_3x_4 \end{array}$$

av längden 3.

Praktiskt kan man nedteckna de par som man använder för att bilda en kortare implikant. En implikant som ger upphov till en kortare implikant kan inte vara prim.

Alltså är proceduren avslutad och

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2 + \bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_3\bar{x}_4 + \bar{x}_1x_3x_4$$

Detta uttryck kan vanligen förenklas ytterligare. För att undersöka sådana möjligheter bildar vi en tabell som innehåller primimplikanter och de mintermer vilka innehåller dessa primimplikanter som delprodukter:

	0000	0001	0010	1000	0011	1100	0111
-000	×			×			
1-00				×		×	
0-11					×		×
00--	×	×	×		×		

(15.18) Definition. Man säger att en primimplikant är **väsentlig** om det finns en minterm som har bara denna implikant som delprodukt. □

Man kan inte utelämna en väsentlig implikant ur representationen av f som summa av primimplikanter (då förändrar man funktionen). I vårt fall är alla implikanter utom -000 väsentliga. Resultatet är

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2 + x_1\bar{x}_3\bar{x}_4 + \bar{x}_1x_3x_4$$

Vi tar ett exempel till. Låt oss tänka oss en sådan här tabell:

	m_1	m_2	m_3	m_4
p_1	$\times$			$\times$
p_2	$\times$	$\times$	$\times$	
p_3		$\times$	$\times$	$\times$

Här är p_i primimplikanter och m_i mintermer. Ingen av p_1, p_2, p_3 är väsentlig – p_1 är inte väsentlig för m_1 (p_2 duger också), p_1 är inte väsentlig för m_4 (p_3 duger), p_2 är inte väsentlig för m_3 (p_3 finns också) osv.

Hur kan man välja p_1, p_2, p_3 så att man får alla m_1, m_2, m_3, m_4?

Man resonerar så här:

$$\begin{array}{lllll} m_1 & \text{fås ur} & p_1 & \text{eller} & p_2 \\ m_2 & \text{fås ur} & p_2 & \text{eller} & p_3 \\ m_3 & \text{fås ur} & p_2 & \text{eller} & p_3 \\ m_4 & \text{fås ur} & p_1 & \text{eller} & p_3 \end{array}$$

Alltså

$$m_1 \text{ och } m_2 \text{ och } m_3 \text{ och } m_4$$

får man ur

$$(p_1 \text{ eller } p_2) \text{ och } (p_2 \text{ eller } p_3) \text{ och } (p_2 \text{ eller } p_3) \text{ och } (p_1 \text{ eller } p_3).$$

Vi kan skriva det genom att använda de logiska konnektiven:

$$m_1 \wedge m_2 \wedge m_3 \wedge m_4 \Leftrightarrow (p_1 \vee p_2) \wedge (p_2 \vee p_3) \wedge (p_2 \vee p_3) \wedge (p_1 \vee p_3)$$

dvs

$$m_1 \wedge m_2 \wedge m_3 \wedge m_4 \Leftrightarrow (p_1 \wedge p_2) \vee (p_1 \wedge p_3) \vee (p_2 \wedge p_3) \vee (p_1 \wedge p_2 \wedge p_3)$$

Med andra ord kan vi välja

$$\begin{aligned} f(\mathbb{x}) = m_1 + m_2 + m_3 + m_4 &= p_1 + p_2 \\ &= p_1 + p_3 \\ &= p_2 + p_3 \\ &= p_1 + p_2 + p_3 \end{aligned}$$

Bland dessa framställningar väljer vi den som innehåller minsta antalet "+" och minsta antalet bokstäver i de ingående termerna.

ÖVNINGAR

15.1. Visa att i varje Boolesk algebra gäller följande likheter:

(a) $x + \bar{x}y = x + y$,

(b) $xy + \bar{x}z + yz = xy + \bar{x}z$,

(c) $(x + \bar{y})(y + \bar{z})(z + \bar{x}) = (\bar{x} + y)(\bar{y} + z)(\bar{z} + x)$,

(d) $\bar{x}\bar{y} + \bar{x}y + x\bar{y} + xy = 1$.

Formulera duala likheter i varje fall.

15.2. Visa att i varje Boolesk algebra gäller följande påståenden:

(a) om $x, y \in \mathcal{B}$ är sådana att $x + y = 1$ och $xy = 0$, så är $y = \bar{x}$,

(b) om $x, y, z \in \mathcal{B}$ och $x + y + z = xyz$, så är $x = y = z$,

(c) om $x, y \in \mathcal{B}$ och $x + y = y$ och $\bar{x}y = 0$, så är $x = y$,

(d) om $x, y, z \in \mathcal{B}$ och $xz + y\bar{z} = 1$, så är $x + y = 1$.

15.3. Visa att om $x, y, z, t \in \mathcal{B}$ och $x = yz + \bar{y}t$, så är $\bar{x} = y\bar{z} + \bar{y}\bar{t}$

15.4. Är det sant att i varje Boolesk algebra gäller att $\overline{xy + yz + zx} = \bar{x}\bar{y} + \bar{y}\bar{z} + \bar{z}\bar{x}$?

15.5. Konstruera kretsar som består av OR, AND, NOT och realiserar följande Booleska funktioner:

(a) $x_1(x_2 + x_3)$, (b) $\overline{(x_1 + x_2)\bar{x}_3} + x_1x_2$,
(c) $x_1 + x_2\bar{x}_3 + x_3\bar{x}_4$, (d) $\overline{x_1(\bar{x}_2 + x_3)}$.

15.6. Visa att varje Boolesk funktion $f(x_1, x_2, \ldots, x_n)$ kan skrivas med hjälp av $x_1, x_2, \ldots, x_n$ och

(a) "+" och "$^-$"; (b) "$\cdot$" och "$^-$".

15.7. Bestäm alla värden på $x_1, x_2, x_3, x_4 \in \{0, 1\}$ som uppfyller ekvationerna $(x_1x_2 + \bar{x}_3)\bar{x}_4 = 1$ och $(\overline{x_2 + x_4} + x_2)x_3 = 0$.

15.8. Bestäm den disjunktiva normalformen för följande Booleska funktioner:

(a) $f(x, y, z) = x + \bar{y} + y(x + \bar{z})$, (b) $f(x_1, x_2, x_3, x_4) = (x_1 + \bar{x}_4)\overline{(\bar{x}_2 + x_3)}$,
(c) $f(x, y, z) = 1$, (d) $f(x, y, z) = x + y\bar{z}$,
(e) $f(x, y, z) = x$, (f) $f(x, y, z) = x\bar{y} + y(x + \bar{z})$,
(g) $f(x, y, z) = y\bar{z}$, (h) $f(x, y, z) = x + y + z$.

Bestäm den disjunktiva normalformen för $\bar{f}$ i alla dessa fall.

15.9. (Den konjunktiva normalformen och maxtermer). Med en maxterm menar man

$$x_1^{i_1} + x_2^{i_2} + \ldots + x_n^{i_n}, \quad \text{där } i_k = 0 \text{ eller } 1.$$

(a) Låt $m = x_1^{i_1}x_2^{i_2}\ldots x_2^{i_n}$ vara en minterm. Visa att $\bar{m}$ är en maxterm.

Sats (15.7) säger att varje Boolesk funktion är en summa av mintermer:

$$f(x_1, x_2, \ldots, x_n) = \sum f(i_1, i_2, \ldots, i_n)x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}.$$

Vi konjugerar den likheten och utnyttjar de Morgans lag samt (a):

$$\begin{aligned}\overline{\overline{f(x_1, x_2, \ldots, x_n)}} &= \overline{\sum \overline{f(i_1, i_2, \ldots, i_n)} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}} = \\ &= \prod (\overline{\overline{f(i_1, i_2, \ldots, i_n)}} + \bar{x}_1^{i_1} + \bar{x}_2^{i_2} + \ldots + \bar{x}_n^{i_n})\end{aligned}$$

dvs

$$(*) \qquad f(x_1, x_2, \ldots, x_n) = \prod (f(i_1, i_2, \ldots, i_n) + \bar{x}_1^{i_1} + \bar{x}_2^{i_2} + \ldots + \bar{x}_n^{i_n})$$

Lägg märke till att bland faktorerna finns det enbart sådana som svarar mot $f(i_1,\ i_2,\ \ldots,\ i_n\ ) = 0$ (dvs $\overline{f(i_1, i_2, \ldots, i_n)} = 1$). De övriga är $= 1$. Formeln $(*)$ kallar man för den **konjunktiva normalformen**. Bestäm den för funktioner ur Övning 8.

15.10. (a) Hur många olika Booleska funktioner $f : \mathbb{Z}_2^n \to \mathbb{B}$ finns det?

(b) En (Boolesk) funktion $f(x_1, x_2, \ldots, x_n)$ kallas symmetrisk om den antar samma värde oberoende av ordningsföljden av $x_1, x_2, \ldots, x_n$. Hur många olika symmetriska Booleska funktioner finns det?

15.11. Försök minimera antalet ”+” och ”·” i representationen av de givna funktionerna

(a) $f(x, y, z) = xy\bar{z} + y\bar{z} + xz$,

(b) $f(x, y, z) = (x + y)(x + \bar{y}) + y$,

(c) $f(x, y, z) = xz + xy + \bar{x}\bar{y}$.

15.12. (NAND grindar). Funktionen NAND (NAND=NOT AND och betecknas med ”↑”) är definierad genom tabellen:

x_1	x_2	$x_1 \uparrow x_2$
0	0	1
0	1	1
1	0	1
1	1	0

Visa att man kan realisera NOT, AND och OR grindar med hjälp av NAND.

15.13. (NOR grindar). Funktionen NOR (NOR=NOT OR och betecknas med ”↓”) är definierad genom tabellen:

x_1	x_2	$x_1 \downarrow x_2$
0	0	1
0	1	0
1	0	0
1	1	0

Visa att man kan realisera NOT, AND och OR grindar med hjälp av NOR.

15.14. (INHIBITOR grind). Funktionen INH är definierad genom tabellen och betecknas med "$*$".

x_1	x_2	$x_1 * x_2$
0	0	0
0	1	0
1	0	1
1	1	0

Visa att man kan realisera NOT, AND och OR med hjälp av INH och en konstant källa av signalen 1.

Ledning. T.ex. är $\bar{x} = 1 * x$.

15.15. (Generatorer för en Boolesk algebra). Om $\mathcal{B}$ är en Boolesk algebra och $x_1, x_2, \ldots, x_n \in \mathcal{B}$ så kan man med hjälp av dessa element och operationerna $+, \cdot, ^-$ bilda alla möjliga uttryck t.ex. $x_1+x_2, x_1+x_2\bar{x}_3$ osv. Om varje element i $\mathcal{B}$ kan skrivas som ett sådant uttryck så säger man att $x_1, x_2, \ldots, x_n$ **genererar** $\mathcal{B}$ och $x_1, x_2, \ldots, x_n$ kallas **generatorer för** $\mathcal{B}$.

(a) Motivera att funktionerna $f_i(x_1 x_2, \ldots, x_n) = x_i$ genererar den Booleska algebran $\mathcal{F}_n$.

(b) Visa att om $x_1, x_2, \ldots, x_n$ genererar den Booleska algebran $\mathcal{B}$ så kan varje element i $\mathcal{B}$ skrivas som summa av mintermer $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$, där $x^1 = x$ och $x^0 = \bar{x}$.

15.16. Vilka av följande delmängder till $U = \{a, b, c, d\}$ bildar en Boolesk algebra (med avseende på $\cap, \cup$ och $^-$):

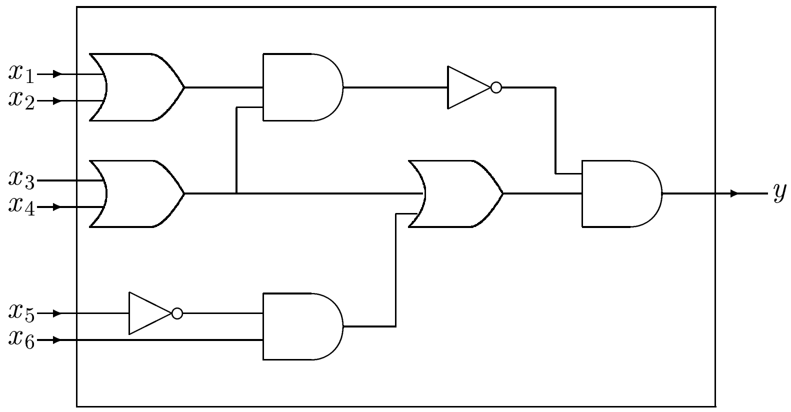
(a) $\emptyset, \{a\}, \{a, b\}, U$;

(b) $\emptyset, \{a, b\}, \{c, d\}, U$;

(c) $\emptyset, \{a, b\}, \{b, c\}, U$;

15.17. Låt $U = \{a, b\}$. Skriv ut tabellerna för addition ($\cup$), multiplikation ($\cap$) och konjugat ($^-$) i den Booleska algebra som består av alla delmängder till U (dvs. $\emptyset = 0, \{a\} = \alpha, \{b\} = \beta, U = 1$)

15.18. Bestäm den Booleska funktion som realiseras av:



15.19. Låt $\mathcal{B}, \mathcal{B}_2$ vara två Booleska algebror. Med produkten av $\mathcal{B}_1$ och $\mathcal{B}_2$ menar man mängden av alla par (b_1, b_2), där $b_1 \in \mathcal{B}_1$ och $b_2 \in \mathcal{B}_2$ och med addition $(b_1, b_2) + (b_1', b_2') = (b_1 + b_1', b_2 + b_2')$, multiplikation $(b_1, b_2)(b_1', b_2') = (b_1 b_1', b_2 b_2')$ och konjugat $\overline{(b_1, b_2)} = (\bar{b}_1, \bar{b}_2)$. Visa att produkten $\mathcal{B}_1$ och $\mathcal{B}_2$ är en Boolesk algebra. Den betecknas med $\mathcal{B}_1 \times \mathcal{B}_2$.

15.20. En (allmän) Boolesk funktion definieras som funktion $f : \mathbb{Z}_2^n \to \mathbb{B}^m$, där $\mathbb{B}^m = \mathbb{B} \times \mathbb{B} \times \ldots \times \mathbb{B}$ (m stycken faktorer). Vi vet (se (15.2)(c)) att sådana funktioner bildar en Boolesk algebra. Med **halvadderare** menar man funktionen $f(x, y) = (s, c)$, där $f : \mathbb{Z}_2^2 \to \mathbb{B}^2$ ges av tabellen:

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Man kan betrakta f som två funktioner $s = f_1(x, y)$ och $c = f_2(x, y)$. Konstruera en krets som realiserar f (dvs f_1 och f_2) och består av OR, AND och NOT-grindar.

15.21. Lös samma uppgift som i förra övningen för **fulladderare** $f(x, y, c) = (s, c')$:

c	x	y	s	c'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

15.22. Konstruera en fulladderare ur två halvadderare och en OR-grind.

15.23. Minimera:

(a) $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1\bar{x}_1x_3 + \bar{x}_1\bar{x}_2\bar{x}_3$,

(b) $f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2$,

(c) $f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4$,

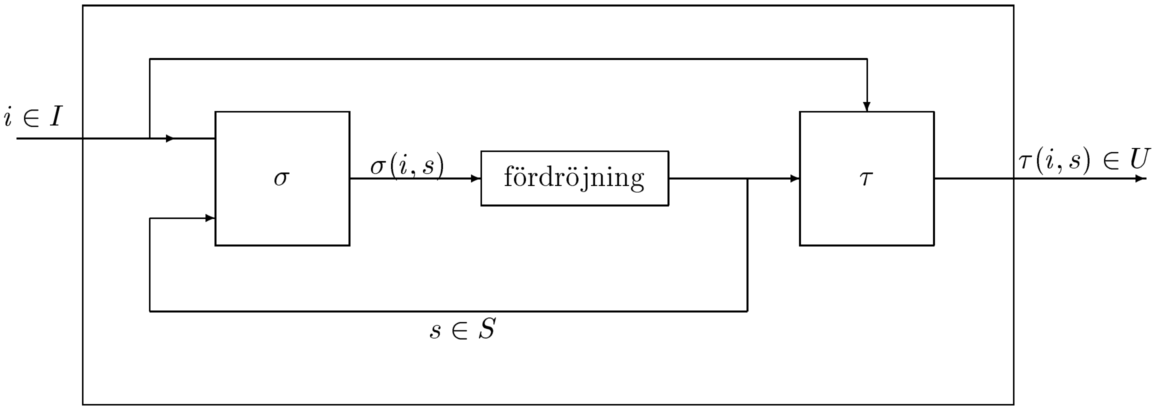
(d) $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_1x_2\bar{x}_3$,

(e) $f(x_1, x_2, x_3) = x_1 + \bar{x}_1x_3 + x_2x_3$.

Kapitel 16

LINJÄRA ÄNDLIGA AUTOMATER

Ändliga automater är abstrakta modeller av komponenter och moduler i datorer. De har en stor praktisk betydelse för analys och syntes av sekvensnät, dvs nätverk bestående av olika binära grindar (t.ex. NAND-grindar – se exempel (16.2) (c)) och fördröjningsenheter. Intuitivt kan en automat uppfattas som en låda med en ingång och en utgång (se fig. 16.1). Ingången påverkas av insignaler i vissa tidsintervall. De resulterar i att lådans tillstånd kan övergå i ett annat samt att man får en utsignal. Nästa tillstånd och utsignal kan bero på både insignalen och det närmast föregående tillståndet. En formell definition är följande:



Figur 16.1

(16.1) Definition. Med en **ändlig automat** $\mathcal{A}$ menar man en uppsättning av tre ändliga mängder:

S – tillståndsmängden,*
I – mängden av insignaler,
U – mängden av utsignaler,

samt två funktioner:

* S från "state".

$\sigma : I \times S \longrightarrow S$ (nästa tillståndsfunktionen),

$\tau : I \times S \longrightarrow U$ (utsignalfunktionen).

En ändlig automat kallas **linjär** om S, I, U är ändligt-dimensionella vektorrum över en ändlig kropp och σ, τ är linjära avbildningar. □

(16.2) Exempel. (a) En vippa (eller ett d-element[†]) $\mathcal{D}$ är en linjär automat för vilken:

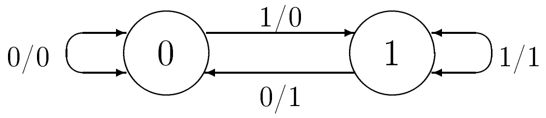
$$I = \{0,1\}, \quad S = \{0,1\}, \quad U = \{0,1\}$$

och funktionerna σ, τ är definierade med hjälp av följande tabell:

Insignal:	σ		τ	
Tillstånd:	0	1	0	1
0	0	1	0	0
1	0	1	1	1

$\mathcal{D}$ fungerar som fördröjning med 1 tidsenhet, dvs utsignalen är lika med insignalen ett tidsintervall[‡] tidigare. Vi har $\sigma(i,s) = i$ (insignalen bestämmer nästa tillstånd) och $\tau(i,s) = s$ (tillståndet bestämmer utsignalen). Det är klart att $\sigma : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ och $\tau : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ är linjära avbildningar.

Man kan beskriva $\mathcal{D}$ med dess graf. Varje vertex svarar mot ett tillstånd (se fig. 16.2). Pilarna $s \overset{a/b}{\to} s'$ säger att vid insignalen a är utsignalen b och nästa tillstånd s'.



Figur 16.2

Varje ändlig automat har en motsvarande graf som kallas **tillståndsgrafen** för automaten.

(b) Låt $\mathcal{A} = (I, S, U, \sigma, \tau)$ vara **adderaren** dvs

$$I = \mathbb{Z}_2^2 = \{00, 01, 10, 11\},\ S = \mathbb{Z}_2 = \{0,1\},\ U = \mathbb{Z}_2 = \{0,1\}.$$

och funktionerna σ, τ är definierade på följande sätt:

[†]"d" från "delay"

[‡]Ett tidsintervall är tiden mellan två på varandra följande insignaler.

Insignal:	σ				τ			
Tillstånd:	00	01	10	11	00	01	10	11
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Adderaren fungerar så att insignalen är två motsvarande siffror a_t och b_t av två tal (i binära systemet):

$$\begin{aligned} a &= a_n a_{n-1} \ldots a_0, \\ b &= b_n b_{n-1} \ldots b_0. \end{aligned}$$

Som resultat får man utsignalen $a_t + b_t + c_t = t$ där c_t är minnessiffran som svarar mot tillståndet av automaten ($c_0 = 0$). Detta betyder att utsignalerna ger siffrorna i $a + b$. Vi lämnar som övning en motivering att $\mathcal{A}$ inte är en linjär automat.

(c) Låt NAND $= (I, S, U, \sigma, \tau)$ där

$$I = \mathbb{Z}_2^2 = \{00, 01, 10, 11\},\ S = \{0\},\ U = \{0, 1\}.$$

Automaten har hela tiden samma tillstånd (ty $S = \{0\}$) så att funktionen σ är ointressant ($\sigma(i, 0) = 0$ för varje i) och τ avbildar I i U. Funktionen τ är definierad så att

x_1	x_2	$\tau(x_1, x_2)$
0	0	1
0	1	1
1	0	1
1	1	0

τ är inte linjär, ty en linjär avbildning måste avbilda 00 på 0. □

(16.3) Anmärkning. (a) Om σ eller τ är oberoende av I (t.ex. $I = \{0\}$), dvs $\sigma(i, s)$ eller $\tau(i, s)$ antar samma värde då s är fixerad och i godtycklig, så skriver vi $\sigma : S \to S$ eller $\tau : S \to U$. Om τ är oberoende av I kallar man automaten för en **Moore-automat**. En automat i definitionens (16.1) mening kallas vanligen för **Mealy-automat**. Om $S = \{0\}$, dvs automaten har enbart ett tillstånd, kallas den **kombinatorisk**.

(b) Man kan visa ganska lätt (det gör man i kurser i digitalteknik) att varje ändlig automat kan realiseras med hjälp av ett sekvensnät, dvs ett nät bestående av linjära logiska grindar (NAND-grindarna är tillräckliga) och fördröjningsenheter (t.ex. av typen (16.2) (a)). Även omvänt, kan varje sekvensnät tolkas som en ändlig automat. □

Vi skall ägna resten av detta avsnitt åt en mycket viktig klass av ändliga automater som kallas linjärt återkopplade skiftregister. Automater av den typen dyker upp mycket naturligt när man studerar rekurrenssekvenser. Sådana sekvenser har en stor betydelse i samband med radarkommunikation, kryptering och slumptalsgenerering. Linjärt återkopplade skiftregister används också i samband med kodning och avkodning. Vi börjar med några exempel på rekurrenssekvenser.

(16.4) Exempel. (a) Låt $s_0 = 1, s_1 = 2$ och $s_{n+2} = s_{n+1} + s_n$ då $n \geq 0$. Dessa villkor definierar en oändlig följd av heltal: $1, 2, 3, 5, 8, \ldots$. Den kallas **Fibonacciföljden** och är en av de mest berömda talsekvenserna i vetenskap och teknik.

(b) Låt $s_0 = 1$ och $s_{n+1} = 2s_n$ då $n \geq 0$. Nu får vi följden $1, 2, 4, 8, \ldots$ dvs $s_n = 2^n$, $n \geq 0$.

(c) Låt $s_0 = 1, s_1 = 0$ och $s_{n+2} = s_{n+1} + s_n$ då $n \geq 0$ men låt $s_i \in \mathbb{Z}_2$. Vi får Fibonacciföljden över $\mathbb{Z}_2 : 1, 0, 1, 1, 0, 1, 1, \ldots$. Följden är tydligen periodisk med perioden 1,0,1 av längden 3. □

(16.5) Definition. Låt $c_1, c_2, \ldots, c_l$ och $s_0, s_1, \ldots, s_{l-1}$ vara givna element i en kommutativ ring R. Låt

$$s_{n+l} = c_1 s_{n+l-1} + c_2 s_{n+l-2} + \ldots + c_l s_n \quad \text{då} \quad n \geq 0.$$

Sekvensen $s_0, s_1, s_2, \ldots$ kallas då för en linjär **rekurrensföljd** och ekvationen som definierar den för en **linjär rekurrensekvation** (eller **differensekvation**). Polynomet

$$p(X) = X^l - c_1 X^{l-1} - \ldots - c_l$$

kallar man för **karakteristiska polynomet** av följden (eller **kopplingspolynomet** av följden). Man säger att följden är **periodisk** om det finns $p > 0$ så att $s_{n+p} = s_n$ för varje $n \geq 0$. p kallas då **perioden** av följden. □

(16.6) Matrisrepresentation av rekurrensföljder. Låt $s = (s_0, s_1, s_2, \ldots)$ vara en rekurrensföljd där

$$s_{n+l} = c_1 s_{n+l-1} + \ldots + c_l s_n$$

och låt $\bar{s}_n = (s_n, s_{n+1}, \ldots, s_{n+l-1})$. Då har vi:

$$\bar{s}_{n+1} = \begin{bmatrix} s_{n+1} \\ s_{n+2} \\ \vdots \\ s_{n+l} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \\ c_l & c_{l-1} & c_{l-2} & \ldots & c_1 \end{bmatrix} \begin{bmatrix} s_n \\ s_{n+1} \\ \vdots \\ s_{n+l-1} \end{bmatrix} = M_s \bar{s}_n$$

dvs $\bar{s}_{n+1} = M_s \bar{s}_n$. Matrisen M_s eller kortare M, där

(16.7)
$$M = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ c_l & c_{l-1} & c_{l-2} & \dots & c_1 \end{bmatrix},$$

kommer att kallas **följdens matris**. Vi har $\bar{s}_{n+1} = M\bar{s}_n = M^2\bar{s}_{n-1} = \ldots = M^{n+1}\bar{s}_0$.

Vidare förutsätter vi att $R = \mathbb{F}$ är en ändlig kropp.

(16.8) Sats. *Låt $s_0, s_1, s_2, \ldots$ vara en rekurrensföljd som i definitionen* (16.1) *med $c_l \neq 0$. Då är sekvensen periodisk och dess kortaste period är en delare till varje annan period.*

Bevis. Vi har $\det(M_s) = \pm c_l \neq 0$ så att $M_s = M$ tillhör gruppen av alla inverterbara $l \times l$-matriser med element ur $\mathbb{F}$. Men den gruppen är ändligt (ty $\mathbb{F}$ är ändlig) så att M har en ändlig ordning T dvs $M^T = E$, där E betecknar enhetsmatrisen. Då är $\bar{s}_{n+T} = M^{n+T}\bar{s}_0 = M^T M^n \bar{s}_0 = \bar{s}_n$ för varje $n \geq 0$. Detta visar att T är en period av rekurrenssekvensen. Bevis av det faktum att den kortaste perioden av en helt godtycklig periodisk följd är en delare till varje annan period lämnar vi som övning (se övn. 16.6). □

(16.9) Linjära automater och rekurrensföljder. Det finns ett mycket nära samband mellan linjära rekurrensföljder och linjära automater. Låt $c_1, c_2, \ldots, c_l \in \mathbb{F}$ och låt

$$I = \{0\}, \quad S = \mathbb{F}^l, \quad U = \mathbb{F}.$$

Låt vidare

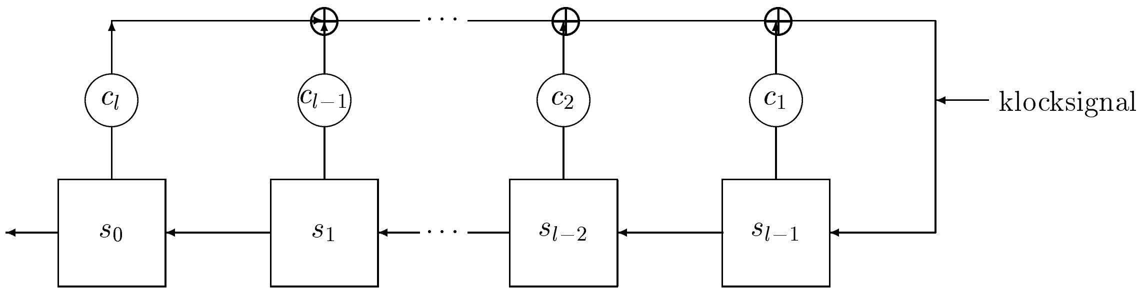
$$\sigma : \mathbb{F}^l \to \mathbb{F}^l,$$

där

$$\sigma(\bar{x}) = M\bar{x}, \qquad \bar{x} = (x_1, \ldots, x_l)^t \ \in \mathbb{F}^l$$

och M är matrisen ovan.

Definiera $\tau : \mathbb{F}^l \to \mathbb{F}$. Vi skall beteckna den automaten med $\mathcal{A}(c_1, c_2, \ldots, c_l)$. Om starttillståndet av den är $\bar{s}_0 = (s_0, \ldots, s_{l-1}) \in \mathbb{F}^l$ så är $\bar{s}_n = M^n\bar{s}_0 = (s_n, s_{n+1}, \ldots, s_{n+l-1})$ dvs



Figur 16.3

$\tau(\bar{s}_n) = s_n$ (se (16.6)). Detta betyder att utsignalerna bildar sekvensen $s_0, s_1, s_2, \ldots$. Automaten representeras grafiskt (och realiseras) som i fig. 16.3.[§]

Automaten kallas **linjärt återkopplat skiftregister**. Vårt syfte är att undersöka för vilka val av $c_1, c_2, \ldots, c_l$ man får långa icke-periodiska sekvenser av utsignaler. Därför är vi intresserade av den kortaste perioden av $s_0, s_1, s_2, \ldots$ (den skall vara lång!).

Nu kan vi visa huvudsatsen i detta avsnitt. Men vi börjar först med ett hjälpresultat som vi utnyttjar i beviset.

(16.10) Lemma. *Låt $s_0, s_1, s_2, \ldots$ vara en rekurrenssekvens som i definitionen* (16.5). *Då är karakteristiska polynomet av matrisen M (se* (16.6)) *lika med karakteristiska polynomet av rekurrenssekvensen.*

Bevis. Vi måste visa att $p(X) = \det(XE - M)$, där E är enhetsmatrisen med l rader och l kolonner. Likheten kan visas med induktion med avseende på l. Om $l = 1$ har man $p(X) = X - c_1$ och $M = [c_1]$ så att likheten gäller. Låt $D_l = \det(XE - M)$. Observera nu att en utveckling av determinanten D_l efter första kolonnen ger

$$D_l = \det(XE - M) = \begin{vmatrix} X & -1 & 0 & \ldots & 0 \\ 0 & X & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & -1 \\ -c_l & -c_{l-1} & -c_{l-2} & \ldots & X - c_1 \end{vmatrix} = XD_{l-1} - c_l.$$

Om $D_{l-1} = X^{l-1} - c_1X^{l-2} - \cdots c_{l-1}$, så är $D_l = X^l - c_1X^{l-1} - \cdots c_{l-1}X - c_l$. □

Vi behöver också ett viktigt begrepp.

(16.11) Definition. Med **exponenten** av ett polynom $p(X) \in \mathbb{F}[X]$ menas det minsta naturliga talet e sådant att $p(X) | X^e - 1$ (se vidare övn. 16.7). □

[§] □ betecknar ett d-element, $\odot$ betecknar en grind som multiplicerar med c om c ersätter punkten, och $\oplus$ är en grind som adderar (alla operationer i $\mathbb{F}$).

(16.12) Sats. *Låt $s_0, s_1, s_2, \ldots$ vara en rekurrensföljd över $\mathbb{F}$ sådan att*

$$s_{n+l} = c_1 s_{n+l-1} + c_2 s_{n+l-2} + \ldots + c_l s_n,\ s_i \in \mathbb{F},\ c_l \neq 0.$$

Om karakteristiska polynomet $X^l - c_1 X^{l-1} - \cdots - c_l$ är irreducibelt så är den kortaste perioden av $s_0, s_1, s_2, \ldots$ lika med exponenten av detta polynom.

Bevis. Låt e beteckna exponenten av karakteristiska polynomet $p(X)$ och T den kortaste perioden för den givna sekvensen. Alltså är e den minsta naturliga exponenten sådan att $p(X)|X^e - 1$ och T är den minsta positiva exponenten sådan att $M^T \bar{s}_0 = s_T = \bar{s}_0$, där M är sekvensens matris.

Låt $K = \mathbb{F}[X]/(p(X)) = \mathbb{F}[\alpha]$, där $\alpha^l - c_1\alpha^{l-1} - \cdots - c_l = 0$. Som vi vet kan varje element i K skrivas som en entydig linjär kombination av $1, \alpha, \ldots, \alpha^{l-1}$ dvs K är ett linjärt rum över $\mathbb{F}$ med dessa element som bas.

Multiplikation av elementen i K med α är en linjär avbildning över $\mathbb{F}$ vars matris är den transponerade matrisen till följdens matris M:

$$M^t = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & c_l \\ 1 & 0 & 0 & \ldots & 0 & c_{l-1} \\ 0 & 1 & 0 & \ldots & 0 & c_{l-2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & c_1 \end{bmatrix}$$

ty $\alpha \cdot \alpha^i = \alpha^{i+1}$ då $i < l-1$ och $\alpha \cdot \alpha^{l-1} = c_l + \cdots + c_1\alpha^{l-1}$.

Observera nu att det karakteristiska polynomet av M^t och M är identiska (det gäller alltid för en godtycklig kvadratisk matris M). Detta visar att $p(X)$ är karakteristiska polynomet för M^t.

Först visar vi att $T \geq e$. Låt T vara den kortaste perioden. Då är $\bar{s}_T = M^T \bar{s}_0 = \bar{s}_0$ så att $\bar{s}_0$ är en egenvektor till M^T hörande till egenvärdet 1. Alltså är 1 ett egenvärde till matrisen $(M^t)^T$. Denna matris svarar mot multiplikation av elementen i K med α^T. Låt $v \in K$ vara en egenvektor hörande till egenvärdet 1 för multiplikation med α^T dvs $\alpha^T v = v$ och $v \neq 0$. Alltså är $(\alpha^T - 1)v = 0$ och eftersom K är en kropp och $v \neq 0$ får man $\alpha^T - 1 = 0$. Detta visar att $X^T - 1$ är delbart med $p(X)$ därför att α är ett nollställe till $p(X)$ och $p(X)$ är ett irreducibelt polynom. Definitionen av e ger att $T \geq e$.

Nu visar vi olikheten $e \geq T$. Eftersom $p(X)$ delar $X^e - 1$ och $p(\alpha) = 0$ så är $\alpha^e = 1$. Alltså är multiplikation av elementen i K med α^e den identiska avbildningen. Detta säger att $(M^t)^e = E$ och således $M^e = E$. Vi får $M^e \bar{s}_0 = \bar{s}_e = \bar{s}_0$, vilket visar att e är en period av den givna sekvensen. Definitionen av T ger nu att $e \geq T$. □

Vi avslutar med en sats som sammanfattar våra tidigare resultat:

(16.13) Sats. *Låt $s_0, s_1, s_2, \ldots$ vara en rekurrensföljd över $\mathbb{F}$ med matrisen M och karakteristiska polynomet $p(X)$. Låt α vara ett nollställe till $p(X)$ i en kropp K som innehåller $\mathbb{F}$. Om $p(X)$ är irreducibelt över $\mathbb{F}$ så är följande tal lika:*

(a) Den kortaste perioden av $s_0, s_1, s_2, \ldots$.

(b) Exponenten av karakteristiska polynomet $p(X)$.

(c) Ordningen av matrisen M i gruppen $GL_l(\mathbb{F})$ av alla $l \times l$ matriser med nollskild determinant och element i kroppen $\mathbb{F}$, dvs den minsta positiva exponent T sådan att $M^T = E$, E enhetsmatrisen.

(d) Ordningen av α i den multiplikativa gruppen av kroppen K dvs den minsta exponent $T > 0$ sådan att $\alpha^T = 1$.

Bevis. Likheten mellan talen i (a) och (b) visades i förra satsen. Likheten mellan talen i (b) och (c) följer direkt från beviset av olikheten $e \geq T$ i samma sats. Vi visade där att $M^e = E$ så att exponenten e är större än eller lika med ordningen $o(M)$ av M. Men $M^{o(M)} = E$ så att $M^{o(M)}\bar{s}_0 = \bar{s}_{o(M)} = \bar{s}_0$, dvs $o(M)$ är perioden av sekvensen. Enligt definitionen av e får man att $o(M) \geq e$. Alltså får man $e = o(M)$. Likheten mellan (a) och (d) följer ur ett mera allmänt resultat nedan. □

(16.14) Lemma. *Låt $f(X)$ vara ett irreducibelt polynom med koefficienter i kroppen $\mathbb{F}$ och låt K vara en kropp som innehåller $\mathbb{F}$ och ett nollställe α till $f(X)$. Då är exponenten av $f(X)$ lika med ordningen av α i den multiplikativa gruppen av K.*

Bevis. Låt e vara exponenten av $f(X)$ och låt T vara ordningen av α i K. Då är $\alpha^T = 1$, så att $X^T - 1$ har α som sitt nollställe. $f(X)$ är ett irreducibelt polynom med detta nollställe så att $f(X)$ delar $X^T - 1$. Detta visar att $T \geq e$. Men $f(X)$ delar $X^e - 1$ så att även $\alpha^e = 1$. Detta visar att $e \geq T$. Tillsammans får vi likheten $e = T$. □

(16.15) Anmärkning. För vilka polynom får man en maximallängdsekvens? Låt oss repetera att ett polynom $p(X)$ kallas primitivt då potenserna av dess nollställe α i $K = \mathbb{F}[X]/(p(X)) = \mathbb{F}[\alpha]$ ger alla nollskilda element. Detta betyder att exponenten av $p(X)$ är lika med antalet nollskilda element i K. När graden av $p(X)$ är fixerad är detta det största möjliga värdet av polynomets exponent. Notera att om graden av $p(X)$ är lika med n och kroppen $\mathbb{F}$ har q element så är antalet element i K lika med q^n. Alltså är största möjliga exponenten av $p(X)$ lika med $q^n - 1$. Detta inträffar precis då $p(X)$ är primitivt. □

(16.16) Exempel. Låt $s_{n+4} = s_{n+3} + s_n$, $n \geq 0$ och $s_i \in \mathbb{Z}_2$. Då är karakteristiska polynomet $X^4 - X^3 - 1 = X^4 + X^3 + 1 \in \mathbb{Z}_2[X]$. Vi vet redan att detta polynom är primitivt. Alltså har följden $s_0, s_1, s_2, \ldots$ kortaste perioden $2^4 - 1 = 15$ om $(s_0, s_1, s_2, s_3) \neq \bar{0}$. Det finns 15 olika val av starttillståndet $\bar{s}_0$ för $\mathcal{A}(1, 0, 0, 1)$. □

ÖVNINGAR

16.1. Bestäm kortaste perioden av följande rekurrensföljder:

(a) $s_{n+3} = s_{n+2} + s_n$, $\bar{s}_0 = (0,1,1)$ över $\mathbb{Z}_2$,

(b) $s_{n+5} = s_{n+2} + s_n$, $\bar{s}_0 = (0,1,0,1,0)$ över $\mathbb{Z}_2$,

(c) $s_{n+4} = s_{n+3} + s_{n+2} + s_{n+1} + s_n$, $\bar{s}_0 = (1,1,1,1)$ över $\mathbb{Z}_2$,

(d) $s_{n+2} = 2s_n$, $\bar{s}_0 = (1,2)$ över $\mathbb{Z}_3$.

16.2. Låt $s = (s_0, s_1, s_2, \ldots)$ vara en periodisk sekvens över $\mathbb{F}$ (dvs det finns p så att $s_{n+p} = s_n$ då $n \geq 0$). Låt $I(s)$ vara mängden av alla polynom $p(X) \in \mathbb{F}[X]$, $p(X) = c_0X^l + c_1X^{l-1} + \ldots + c_l$ sådana att

$$c_0 s_{n+l} + c_1 s_{n+l-1} + \ldots + c_l s_n = 0 \quad \text{för varje } n \geq 0.$$

Visa att $I(s)$ är ett ideal i $\mathbb{F}[X]$ (notera att $X^p - 1$ är ett sådant polynom i $I(s)$).

Anmärkning. Idealet $I(s)$ är principalt (som alla ideal i $\mathbb{F}[X]$). Låt $I(s) = (m(X))$, där $m(X)$ har högsta koefficienten 1. Då kallas $m(X)$ **minimipolynomet** för s. Det är en delare till varje polynom $p(X) \in I(s)$ och bland annat till $X^p - 1$. Detta ger en möjlighet att beräkna $m(X)$. Notera att polynomen i $I(s)$ kan beskrivas som karakteristiska polynomen av s.

16.3. Bestäm alla rekurrensföljder som definieras av den givna rekurrensekvationen över $\mathbb{Z}_2$. Bestäm deras kortaste perioder:

(a) $s_{n+2} = s_n$, $n \geq 0$,

(b) $s_{n+3} = s_{n+2}$, $n \geq 0$.

16.4. Bestäm det kortaste linjärt återkopplade skiftregister som genererar den periodiska sekvensen $s = (s_0, s_1, s_2, \ldots)$ med perioden p då:

(a) $s = (1,1,0,1,1,\ldots)$, $p = 5$ över $\mathbb{Z}_2$,

(b) $s = (2,1,1,2,1,1,\ldots)$, $p = 6$ över $\mathbb{Z}_3$,

(c) $s = (1,2,3,1,1,2,\ldots)$, $p = 6$ över $\mathbb{Z}_5$.

16.5. Låt $\mathcal{A}$ vara en ändlig automat som ges av följande:

$$I = \{0,1\},\ S = \{z_0, z_1\},\ U = \{0,1\}$$

Insignal:	σ		τ	
Tillstånd:	0	1	0	1
z_0	z_0	z_1	0	1
z_1	z_1	z_0	0	1

(a) Rita grafen av $\mathcal{A}$;

(b) Försök beskriva hur $\mathcal{A}$ fungerar (studera utsignalsekvensen);

(c) Är $\mathcal{A}$ en linjär automat?

16.6. Låt $s_0, s_1, s_2, \ldots$ vara en periodisk följd med kortaste perioden p^*. Visa att $p^*|p$ om p är en period.

Ledning: $p = p^*g + r, \quad 0 \leq r < p^*$. Visa att även r är en period.

16.7. Låt $p(X)$ vara ett irreducibelt polynom av grad n över en ändlig kropp $\mathbb{F}$. Visa att $p(X)$ är en delare till polynomet $X^N - 1$ för ett naturligt tal N.

Ledning: Betrakta den ändliga kroppen $\mathbb{F}[X]/(p(X)) = \mathbb{F}[\alpha]$, där $p(\alpha) = 0$, och motivera att $\alpha^N = 1$ för ett lämpligt N.

Chapter 17

FAST FOURIER TRANSFORM

(17.1) The Fourier transform of a periodic function. The development of a function in its harmonics has many applications in physics and engineering. The Fourier series of a function f with period 1 looks like

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos 2n\pi x + b_n \sin 2n\pi x) \ .$$

The formulas simplify if we use complex numbers:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{2n\pi i x} \ ,$$

where the Fourier coefficients c_n can be computed as

$$c_n = \int_0^1 f(x) e^{-2n\pi i x} \, dx \ .$$

When computing numerically one approximates an integral like $\int_0^1 \varphi(x)\, dx$ by a finite sum. A simple method would be to divide the interval $[0,1]$ in N equally long subintervals and to compute

$$\frac{1}{N} \sum_{j=0}^{N-1} \varphi(\tfrac{j}{N}) \ .$$

In particular we have approximations

$$c_n \sim \widetilde{c}_n := \frac{1}{N}\sum_{j=0}^{N-1} f(\tfrac{j}{N}) e^{\frac{-2nj\pi i}{N}} \,. \tag{17.2}$$

Note that the coefficients $\widetilde{c}_n$ depend only on the value of f in the N points $0, \frac{1}{N}, \ldots, \frac{N-1}{N}$. Note also that there are only N different numbers $\widetilde{c}_n$, as $\widetilde{c}_{N+n} = \widetilde{c}_n$. So our formula (17.2) can be seen as a linear transformation which associates a vector* $(\widetilde{c}_0, \ldots, \widetilde{c}_{N-1})$ to the vector $(f(0), \ldots, f(\frac{N-1}{N}))$. We can apply the transformation to any vector in $\mathbb{C}^N$. This leads us to the definition of the discrete Fourier transform.

(17.3) Definition. Let K be a field. An element $\alpha \in K$ with $\alpha^N = 1$ is called an **Nth root of unity**. An Nth root of unity ω is **primitive** if $\omega^N = 1$ but $\omega^j \neq 1$ for $0 < j < N$. $\square$

Example. Let $K = \mathbb{C}$. A primitive Nth root of unity is $e^{\frac{2\pi i}{N}}$. In particular, for $N = 8$ we have $\omega = e^{\frac{\pi i}{4}} = \frac{1}{2}\sqrt{2}(1+i)$. The element $\omega^2 = i$ is not a primitive 8th root of unity (but it is a primitive 4th root of unity). The primitive 8th roots of unity are ω, ω^3, ω^5 and ω^7. $\square$

Example. Let $K = \mathbb{Z}_{101}$. Then 10 is a primitive 4th root of unity: as $10^2 \equiv -1 \pmod{101}$, we have that $10^4 \equiv 1 \pmod{101}$. In particular, 10 is a square root of -1 in $\mathbb{Z}_{101}$. $\square$

(17.4) Definition. Let $\omega \in K$ be a primitive Nth root of unity. Let Ω be the linear transformation $\Omega\colon K^N \to K^N$ with matrix $\Omega_{mj} = \omega^{mj}$ for $0 \leq m, j < N$. This matrix depends on the chosen root of unity ω and to emphasise this dependence we sometimes write Ω_ω. The **discrete Fourier transform** $F(f)$ of the vector $f = (f_0, , \ldots, f_{N-1})$ is the vector Ωf. Its mth component is $\sum_{j=0}^{N-1} \Omega_{mj} f_j = \sum_{j=0}^{N-1} f_j \omega^{mj}$. $\square$

Example. Let $K = \mathbb{C}$, $N = 4$ and $\omega = -i$, $f = (2, 1+i, 2i, 0)$. Then

$$\Omega f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 2 \\ 1+i \\ 2i \\ 0 \end{pmatrix} = \begin{pmatrix} 3+3i \\ 3-3i \\ 1+i \\ 1-i \end{pmatrix} .$$

The matrix Ω is invertible, as $\det \Omega \neq 0$: it is a Vandermonde determinant. One easily computes that $\det \Omega = 16i$. The columns of the matrix are hermitian orthogonal and we have $\Omega\overline{\Omega} = 4I$, so the inverse of Ω is $\frac{1}{4}\overline{\Omega}$. The matrix $\overline{\Omega}$ is of the same type as Ω, but formed with the primitive root of unity $\omega^{-1} = i$, so $\overline{\Omega}_{-i} = \Omega_i$. $\square$

This holds in general: $\Omega_\omega^{-1} = \frac{1}{N}\Omega_{\omega^{-1}}$ (see Exercise 17.4). A more symmetric formula would be obtained by taking $\frac{1}{\sqrt{N}}\Omega_\omega$ as matrix (with inverse $\frac{1}{\sqrt{N}}\Omega_{\omega^{-1}}$) but that choice complicates the computations.

---

*We consider vectors always as column vectors, but in running text it is easier to write them as rows. It would be more correct to write here $(\widetilde{c}_0, \ldots, \widetilde{c}_{N-1})^t$.

Example. Let $K = \mathbb{Z}_{101}$, $N = 4$ and $\omega = -10$, $f = (2, 11, 20, 0)$ Then

$$\Omega f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -10 & -1 & 10 \\ 1 & -1 & 1 & -1 \\ 1 & 10 & -1 & -10 \end{pmatrix} \begin{pmatrix} 2 \\ 11 \\ 20 \\ 0 \end{pmatrix} = \begin{pmatrix} 33 \\ -27 \\ 11 \\ -9 \end{pmatrix} .$$

□

(17.5) The Fast Fourier Transform (FFT) algorithm. To compute the discrete Fourier transform using matrix multiplication as above uses $O(N^2)$ multiplications. The FFT algorithm reduces this to $O(N \log N)$ operations. It first appeared in a paper by C. Runge from 1903, but it was completely forgotten as the difference becomes only important for large N. The modern history begins with a paper of J.W. Cooley and J.W. Tukey [Math. Comput. **19** (1965), 297–301]. Actually there are several related algorithms which all go under the name FFT.

The strategy used is 'divide and conquer': split a problem into two subproblems of approximately half size, recursively solve each subproblem and combine the solutions of the subproblems to form the solution of the given problem. The FFT works best if we take $N = 2^n$. Variants exist for other values of N, but we concentrate on this case.

The algorithm is based on the observation that the mth component $\alpha_m := \sum_{j=0}^{N-1} f_j \omega^{mj}$ of $F(f)$ is the value at $X = \omega^m$ of the polynomial $f(X) = f_0 + f_1 X + \cdots + f_{N-1} X^{N-1}$. As N is even we can write $N = 2M$ and collect terms whose exponent differ by M. So

$$f(X) = (f_0 + f_M X^M) + \cdots + (f_{M-1} + f_{2M-1} X^M) X^{M-1} .$$

Now we note that $\omega^M = -1$ and that ω^2 is a primitive Mth root of unity. For even $m = 2h$ we find

$$\alpha_{2h} = \sum_{k=0}^{M-1} (f_k + f_{k+M}) X^k |_{X=\omega^{2h}} = \sum_{k=0}^{M-1} f'_k X^k |_{X=(\omega^2)^h} ,$$

which amounts to computing a discrete Fourier transform for a different vector f', with $M = N/2$ and ω^2 as root of unity. For odd $m = 2h+1$ we have a similar, slightly more complicated formula:

$$\alpha_{2h+1} = \sum_{k=0}^{M-1} (f_k - f_{k+M}) X^k |_{X=\omega^{2h+1}} = \sum_{k=0}^{M-1} ((f_k - f_{k+M})\omega^k) X^k |_{X=\omega^{2h}} = \sum_{k=0}^{M-1} f''_k X^k |_{X=(\omega^2)^h} .$$

Again we have the problem of computing a discrete Fourier transform with $M = N/2$ and ω^2 as root of unity, but now starting from the vector f''. So indeed we have divided our problem into two subproblems of approximately half the size. Each of these subproblems can be solved in the same way: we compute the Fourier transform recursively. We formulate an explicit computation scheme.

(17.6) Theorem. *Set $f^{(n)}_{0k} := f_k$ for $k = 0, 1, , \ldots, N-1$ and define recursively*

$$\begin{aligned} f^{(r-1)}_{mk} &= f^{(r)}_{mk} + f^{(r)}_{m,k+2^{r-1}} \\ f^{(r-1)}_{m+2^{n-r},k} &= (f^{(r)}_{mk} - f^{(r)}_{m,k+2^{r-1}})(\omega^{2^{n-r}})^k \end{aligned}$$

for $r = n, n-1, \ldots, 1$, $m = 0, \ldots, 2^{n-r}$, $k = 0, \ldots, 2^{r-1}$. The coefficients of the discrete Fourier transform satisfy $\alpha_{j2^{n-r}+m} = \sum_{k=0}^{2^r-1} f^{(r)}_{mk}(\omega^{2^{n-r}})^{jk}$ for all r and in particular $\alpha_m = f^{(0)}_{m0}$.

Proof. By induction on n; the induction start is the computation preceeding this theorem. For $j = 2h+1$ we compute $\alpha_{h2^{n-r+1}+2^{n-r}+m} = \alpha_{j2^{n-r}+m}$ to be

$$\sum_{k=0}^{2^r-1} f^{(r)}_{mk}(\omega^{2^{n-r}})^{jk} = \sum_{k=0}^{2^{r-1}-1} (f^{(r)}_{mk} - f^{(r)}_{m,k+2^{r-1}})(\omega^{2^{n-r}})^k(\omega^{2^{n-r+1}})^{hk} \, ,$$

which indeed equals $\sum_{k=0}^{2^{r-1}-1} f^{(r)}_{2^{n-r}+m,k}(\omega^{2^{n-r+1}})^{hk}$. For $j = 2h$ the computation is similar. $\square$

As to the practical realisation of this algorithm, one needs only an array $\widetilde{f}\,[0{:}\, N-1]$ of length N to store the N quantities $f^{(r)}_{mk}$ if one replaces the pair $f^{(r)}_{mk}$ and $f^{(r)}_{m,k+2^{r-1}}$ by the pair $f^{(r-1)}_{mk}$ and $f^{(r-1)}_{m+2^{n-r},k}$ after computation. So one stores $f^{(r)}_{mk}$ as the sth component of the array $\widetilde{f}$ with $s(r, m, k)$ recursively defined by

$$\begin{aligned} s(n, 0, k) &= k \\ s(r-1, m, k) &= s(r, m, k) \\ s(r-1, m+2^{n-r}, k) &= s(r, m, k+2^{r-1}) \, . \end{aligned}$$

Using the binary representations $m = \beta_{n-1} + \beta_{n-2}2 + \ldots \beta_r 2^{n-r-1}$ and $k = \beta_0 + \cdots + \beta_{r-1}2^{r-1}$ one has

$$s(r, m, k) = \beta_0 + \cdots + \beta_{r-1}2^{r-1} + \beta_r 2^r + \cdots + \beta_{n-1}2^{n-1} \, .$$

In particular, $s(0,m,0) = \rho(m)$ with ρ the so-called **bit-reversal map**

$$\rho(\beta_0 + \ldots + \beta_{n-1}2^{n-1}) = \beta_{n-1} + \ldots + \beta_0 2^{n-1}\ .$$

We write a pseudo program:

for $r := n$ **step** -1 **until** 1
  do for $k = 0$ **step** 1 **until** $2^{r-1}-1$
    do $w := (\omega^{2^{n-r}})^k$**;** **for** $m = 0$ **step** 2^r **until** $2^n - 1$
      do $u := \widetilde{f}[m+k]$**;** $v := \widetilde{f}[m+k+2^{r-1}]$**;**
        $\widetilde{f}[m+k] := u+v$**;** $\widetilde{f}[m+k+2^{r-1}] := (u-v)w$
      od
    od
  od;

Example. $N = 4$, $\omega = -i$, $f = (2, 1+i, 2i, 0)$. We get

$$\begin{array}{lclcccccl}
\widetilde{f}[00] & = & 2 & \longrightarrow & 2+2i & \longrightarrow & 3+3i & = & \alpha_0 \\
\widetilde{f}[01] & = & 1+i & \longrightarrow & 1+i & \longrightarrow & 1+i & = & \alpha_2 \\
\widetilde{f}[10] & = & 2i & \longrightarrow & 2-2i & \longrightarrow & 3-3i & = & \alpha_1 \\
\widetilde{f}[11] & = & 0 & \longrightarrow & 1-i & \longrightarrow & 1-i & = & \alpha_3
\end{array}$$

□

In the second column of arrows one has to use the powers of ω^2, not of ω. In the example above one might fail to notice this, because only the zeroth power occurs.

To compute the inverse transformation one can either use the same algorithm, but now applied to ω^{-1}, or go backwards, i.e., compute the inverse of each step. This leads to the original algorithm of Cooley and Tucker. To formulate it we observe that α_m may be computed as the remainder of $f(X)$ upon division by $X - \omega^m$. The linear term is a factor of $(X-\omega^m)(X+\omega^m) = X^2 - \omega^{2m}$. We can first divide by $X^2 - \omega^{2m}$ and then divide the remainder by $X - \omega^m$, which again leads to a recursion. We use the following easily proved result.

(17.7) Lemma. *Let R be a commutative ring with unit. Let $p = p_1 p_2$ be the product of two monic polynomials in $R[X]$. Let $f \in R[X]$ give remainder r upon division by p: $f = qp + r$ with $\deg r < \deg p$. Let $r = q_1 p_1 + r_1$ with $\deg r_1 < \deg p_1$. Then r_1 is the remainder of f upon division by p_1.*

The numbers $1 = \omega^0, \ldots, \omega^{N-1}$ are the roots of the polynomial $X^N - 1$. We can factorise $X^N - 1 = X^N - \omega^N$ in steps, again using that $N = 2^n$. Given $X^s - \omega^t$ with $s = 2s'$ and $t = 2t'$ even we factorise $X^s - \omega^t = (X^{s'} - \omega^{t'})(X^{s'} + \omega^{t'}) = (X^{s'} - \omega^{t'})(X^{s'} - \omega^{t'+M})$, where $N = 2M$.

Example. $N = 4$, $\omega = -i$.

$$\begin{array}{ccccccc}
 & & & X^4 - 1 & & & \\
 & & / & & \backslash & & \\
 & X^2 - 1 & & & & X^2 - (-1) & \\
 / & & \backslash & & / & & \backslash \\
X - 1 & & X - (-1) & & X - i & & X - (-i)
\end{array}$$

Consider as before $f = 2 + (1+i)X + 2iX^2$. We write the remainders in the same tree.

$$\begin{array}{ccccccc}
 & & & 2 + (1+i)X + 2iX^2 & & & \\
 & & / & & \backslash & & \\
 & (1+i)X + (2+2i) & & & & (1+i)X + (2-2i) & \\
 / & & \backslash & & / & & \backslash \\
3 + 3i & & 1 + i & & 1 - i & & 3 - 3i
\end{array}$$

□

In the steps of the division we always divide a polynomial of degree at most $2s-1$ by one of the form $X^s - c$. This is particularly simple: replace every occurrence of X^s by c. If $f = a_{2s-1}X^{2s-1} + \ldots + a_1X + a_0$ then $r = (a_{s-1} + ca_{2s-1})X^{s-1} + \ldots + (a_0 + ca_s)$. We can therefore compute the coefficients with a similar recursion as in the first algorithm.

(17.8) Theorem. *Set $f^{(0)}_{m0} := f_m$ for $m = 0,, \ldots, N-1$ and define recursively*

$$\begin{aligned}
f^{(r-1)}_{mk} &= f^{(r)}_{mk} + f^{(r)}_{m+2^{n-r}}(\omega^{2^{n-r}})^k \\
f^{(r-1)}_{m,k+2^{r-1}} &= (f^{(r)}_{mk} - f^{(r)}_{m+2^{n-r},k})(\omega^{2^{n-r}})^k
\end{aligned}$$

for $r = 1, \ldots, n$, $m = 0, \ldots, 2^{n-r}$, $k = 0, \ldots, 2^{r-1}$. Then $\alpha_k = f^{(n)}_{0k}$.

Again we can use one array of length N to store the numbers involved. As the recursion goes in the other direction we have to place the f_m in bit reversed order in the array. Our pseudo program becomes:

for $r := 1$ **step** 1 **until** n
  do for $k = 0$ **step** 1 **until** $2^{r-1} - 1$
    do $w := (\omega^{2^{n-r}})^k$; **for** $m = 0$ **step** 2^r **until** $2^n - 1$
      do $u := \widetilde{f}[m+k]$; $v := \widetilde{f}[m+k+2^{r-1}]w$;
        $\widetilde{f}[m+k] := u+v$; $\widetilde{f}[m+k+2^{r-1}] := u-v$
      od
    od
  od;

We count the number of operations: for each step in the innermost loop we have two additions and one multiplication, and for each r there are $N/2$ such steps, with r running till $n = \log N$. The values of ω^k can be computed at the start, so the number of operations needed is $O(N \log N)$. The transform deserves its name.

EXERCISES

17.1. Show that the Nth roots of unity in a field K form an Abelian group under multiplication.

17.2. Let μ_N be the group of Nth roots of unity in $\mathbb{C}$. Show that it is isomorphic to $(\mathbb{Z}_N, +)$. Show that under this isomorphism the primitive roots of unity are mapped to $\mathbb{Z}_N^*$.

17.3. Verify that 3 is a primitive 6th root of unity in $\mathbb{Z}_7$. With $N = 6$, $\omega = 3$ write out the matrix Ω from Def. (17.4).

17.4. a) Let ω be a primitive Nth root of unity in a field K. Show that $\sum_{j=0}^{N-1} \omega^{jm} = N$ if $N|m$ and 0 otherwise.

b) Prove that $\Omega_\omega \cdot \Omega_{\omega^{-1}} = N\,I$.

17.5. Prove Lemma (17.7).

17.6. Explain the following `MAPLE` session. (Hint: use the online help.)

```
    |\^/|     Maple V Release 4 (Chalmers University of Tech)
._|\|   |/|_. Copyright (c) 1981-1996 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> readlib(FFT);
                              proc(m, x, y)  ...  end

> x:=array([2,1,0,0]);y:=array([0,1,2,0]);
                                 x := [2, 1, 0, 0]

                                 y := [0, 1, 2, 0]
```

```
> FFT(2,x,y);
                                        4

> print(x);print(y);
                                [3, 3., 1, 1.]

                               [3, -3., 1, -1.]

> iFFT(2,x,y);
                                        4

> print(x);print(y);
                     [2.000000000, 1.000000000, 0, 0]

                     [0, 1.000000000, 2.000000000, 0]
```

Chapter 18

FAST MULTIPLICATION

In this chapter we describe algorithms to perform exact multiplication of large integers. The asymptotically fastest algorithm is due to Schönhage–Strassen [*Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292], but it is difficult to implement. Several algorithms were tested by D.H. Bailey in his computations of digits of the number π, see [J.M Borwein, P.B. Borwein and D.H. Bailey, *Ramanujan, modular equations, and approximations to pi, or How to compute one billion digits of pi*. Amer. Math. Monthly **96** (1989), 201–219]. All have in common that they are based on the Fast Fourier Transform.

(18.1) An integer a written in base b can be considered as a special polynomial $a = \sum a_i b^i$ where all coefficients a_i lie between 0 and $b-1$. In the decimal system $27 = 2\cdot 10^1 + 7\cdot 10^0$ while in binary notation the same number is represented as $11011 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$. The difference between numbers $\sum a_i b^i$ and polynomials in $\mathbb{Z}[b]$ becomes only apparent after adding two such: for numbers we can have carries, whereas for polynomials the coefficients just will become larger than $b-1$. To come to a number we have to release the carry, which is done by evaluating the polynomial at the point b.

Given an integer we divide its binary or decimal expansion in equally long chunks, e.g., of word length L. This determines b (e.g., $b = 2^L$). With b now fixed we consider two integers $f = \sum f_i b^i$ and $g = \sum g_i b^i$ and the corresponding polynomials $f(X) = \sum f_i X^i$, $g(X) = \sum g_i X^i$. Let $\sum h_i X^i = h(X) := f(X)g(X)$. The coefficients satisfy

$$h_m = \sum_{i=0}^{m} f_{m-i} g_i \ .$$

To multiply in this way the product of two polynomials of degree N requires $O(N^2)$ multiplications and additions, but here the FFT will come in.

(18.2) Definition. Let K be a field and $f = (f_0, \ldots, f_l, 0, \ldots, 0)$, $g = (g_0, \ldots, g_l, 0, \ldots, 0)$ be two vectors in K^N with $f_m = g_m = 0$ for $m \geq N/2$. The **convolution** $f * g$ of f and g is the vector h with coefficients $h_m = \sum f_{m-i} g_i$. $\square$

(18.3) Definition. Let $f = (f_0, \ldots, f_l, 0, \ldots, 0)$, $g = (g_0, \ldots, g_l, 0, \ldots, 0)$ be two vectors in K^N. The **component-wise multiplication** $f \odot g$ of f and g is the vector h with coefficients $h_m = f_m g_m$. □

(18.4) Theorem. *Let F be a discrete Fourier transform on K^N. Let f and g be vectors for which the convolution is defined. Then $F(f * g) = F(f) \odot F(g)$.*

Proof. Denote by $f(X) = \sum f_i X^i \in K[X]$ the polynomial corresponding to the vector f. Then the mth coefficient of $F(f)$ equals $f(\omega^m)$, that of $F(g)$ equals $g(\omega^m)$ and that of $F(f*g)$ is $(fg)(\omega^m) = f(\omega^m)g(\omega^m)$. □

Example. $N = 4$, $K = \mathbb{C}$, $\omega = -i$ (as in the example in (17.4)), $b = 10$. We multiply $f = 97$ with $g = 99$. We have

$$\Omega \begin{pmatrix} 7 \\ 9 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 16 \\ 7-9i \\ -2 \\ 7+9i \end{pmatrix}, \qquad \Omega \begin{pmatrix} 9 \\ 9 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 18 \\ 9-9i \\ 0 \\ 9+9i \end{pmatrix}.$$

So $F(f * g) = F(f) \odot F(g)$ is the column vector $(288, -18 - 144i, 0, -18 + 144i)$ and the inverse Fourier transform is obtained by multiplying with $\frac{1}{4}\overline{\Omega}$, yielding $\frac{1}{4}(252, 576, 324, 0) = (63, 144, 81, 0)$ or $fg = 63+1440+8100 = 9603$, which is indeed the correct answer $(9700-97)$. □

Example. Use $K = \mathbb{Z}_{101}$, $N = 4$ and $\omega = -10$ on the same numbers, now using multiplication in $\mathbb{Z}_{101}$, so the answer is $(-2) \cdot (-4) = 8$. We get

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -10 & -1 & 10 \\ 1 & -1 & 1 & -1 \\ 1 & 10 & -1 & -10 \end{pmatrix} \begin{pmatrix} 7 & 9 \\ 9 & 9 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 16 & 18 \\ 18 & 20 \\ -2 & 0 \\ -4 & -2 \end{pmatrix}.$$

Then $F(f * g) = (-15, -44, 0, 8)$; multiplication with $\overline{\Omega}$ gives $(50, -30, 21, 0)$. To divide by 4 we multiply with -25, yielding $(63, 43, 81, 0)$. Now $10^2 \equiv -1 \pmod{101}$, so we get $63 + 430 - 81 \equiv 8$. To obtain the product of the polynomials f and g in $\mathbb{Z}[X]$ instead of $\mathbb{Z}_{101}[X]$ we note that only the second coefficient can be bigger than 101, but is at most 162. So therefore it is either 43 or 144, which can be decided by computing $f(X)g(X) \bmod 2$: we get $f(X)g(X) \equiv (1+X)(1+X) \equiv 1 + X^2 \pmod 2$, and the coefficient is 144, in agreement with the computation over $\mathbb{C}$. □

In these examples the release of the carry leads to three additions, one more than in the classical algorithm, and the multiplications are also not easier. But for large numbers there will be considerable savings, especially when the discrete Fourier transform is computed with the FFT algorithm. The disadvantage of working over $\mathbb{C}$ is that the Nth roots of unity for large $N = 2^n$ are irrational complex numbers, which requires floating point arithmetic and a careful control of rounding errors. But as the end result consists of integers, rounding to the nearest integer will give the correct result. Computation over a finite field has the advantage that the computation always stays exact. In his computation of π the current record holder* Yasumasa Kanada (University of Tokyo) actually used a $\mathbb{C}$-based algorithm. Supercomputer architecture is geared to floating point computations.

(18.5) To use the FFT algorithm over a finite field we need a $\mathbb{Z}_p$ with an $N = 2^n$th root of unity.

Lemma. A field $\mathbb{Z}_p$ contains a primitive Nth root of unity if and only if N divides $p-1$.

Proof. The order of an element of $\mathbb{Z}_p^*$ divides the order of the group, which is $p-1$. Conversely, $\mathbb{Z}_p^*$ is cyclic so has an element α of order $p-1$. Then $\alpha^{\frac{p-1}{N}}$ has order N. $\square$

We list some primes smaller than $2^{31}-1$, which is suitable for use in 32-bit machines (1 bit used for sign):

$$\begin{array}{rcll}
2013265921 & = & 2^{27}\cdot 15+1\,, & 31 \text{ is a primitive } 2^{27}\text{th root of unity}\\
2113929217 & = & 2^{25}\cdot 63+1\,, & 5 \text{ is a primitive } 2^{25}\text{th root of unity}\\
2130706433 & = & 2^{24}\cdot 127+1\,, & 3 \text{ is a primitive } 2^{24}\text{th root of unity}
\end{array}$$

We now describe an algorithm due to Pollard and analysed by Lipson (see [Lipson, *Elements of algebra and algebraic computing*]). Write again f and g as polynomials of degree $N-1$ in base b with $b < 2^{31}-1$; with word size 32 this allows storage of the b-ary digits as one machine word, which is important for the speed of the algorithm. If we suppose that we have input and output given as string of decimal digits, a good choice for b is 10^9. By grouping those in equally long chunks, i.e., in working in a base which is a power of ten, we avoid having first to convert the number to binary. Choose k primes $p_i > b$ of the form $p = 2^e l + 1$, each smaller than word size. We shall shortly determine which value of k is needed. Now compute $h_i(X) := f(X)g(X) \pmod{p_i}$ in $\mathbb{Z}_{p_i}[X]$ using FFT. Determine $h(X) := f(X)g(X) \in \mathbb{Z}[X]$ with the Chinese Remainder Theorem.

To correctly use the FFT for the convolution of two vectors we need that $2N-1 \le 2^E$ where E is the smallest exponent e_i of our primes p_i. To choose k we note that the coefficients h_m of $h(X)$ satisfy $h_m < Nb^2$, so we can recover h with the Chinese Remainder Theorem if $\prod p_i > Nb^2$. As $N \le 2^{E-1} < b$ and $p_i > b$ we see that three primes suffice, for which we can take the primes listed above. Then $E = 24$ and we can only multiply numbers of maximally $31 \cdot 2^{23}$ binary digits, or approximately 800 million decimal digits. The number of operations needed is clearly dominated by the FFT and its inverse, giving a bound $O(N \log N)$, but only for finitely many N.

*206,158,430,000 decimal digits, computed in 1999.

(18.6) We come to the Schönhage-Strasssen algorithm, which gives an asymptotic bound. Consider two binary N-digit numbers. The first idea is the following. Write $N = KL$ with K and L approximately equal ($\sim \sqrt{N}$). We use base $b = 2^L$, and apply a FFT polynomial multiplication as above. Now the digit in base b are still large integers, so for arithmetic in base b we have to use a fast multiplication, but with much smaller base. This yields a nested system of fast multiplications, which ends with multiplication of small numbers with the school method. If the product of two large natural numbers f and g is less than p, then it makes no difference if we consider the multiplication in $\mathbb{Z}$ or in $\mathbb{Z}_p$. Therefore the goal is to describe multiplication in $\mathbb{Z}_p$ in terms of arithmetic over $\mathbb{Z}_q$ with q a smaller prime.

The second idea is to use only Fermat primes $F_n = 2^{2^n}+1$ for optimal use of the binary number system. Not all F_n are prime (see exercise 3 in the chapter Restgrupper). The observation is that this is not needed for the FFT algorithm to work. For a field $k = \mathbb{Z}_{F_n}$ we worked with vector spaces k^n; if F_n is not prime then $\mathbb{Z}_{F_n}$ is only a ring, which we now write R. One computes with matrices over R in the same way as usual, except that one has to be careful with division. The vector space k^N is now replaced by R^N, which is called a free R-module of rank N; we write as before elements as column vectors.

(18.7) Proposition. *The number 2 is a primitive 2^{n+1}th root of unity in $\mathbb{Z}_{F_n}$. In particular 2 is invertible. The matrix Ω_2 with entries $2^{i\cdot j}$ has inverse $2^{-(n+1)}\Omega_{1/2}$.*

The first statement follows directly from the congruence $2^{2^n} \equiv -1 \pmod{F_n}$, once we have extended the definition of roots of unity from fields to commutative rings with unit. For the remaining statements we refer to the exercises.

(18.8) Example. As an example of the reduction step in the algorithm we consider multiplication in $\mathbb{Z}_{F_3} = \mathbb{Z}_{257}$, which will be reduced to arithmetic in $\mathbb{Z}_{17}$. We also need reduction modulo $2^2 = 4$. We use the hexadecimal system to write numbers, with digits `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `A`, `B`, `C`, `D`, `E` and `F`. For the residue classes in $\mathbb{Z}_{17}$ we use also the numbers $-8, \ldots, 8$.

We multiply $f = \texttt{BE}$ with $g = \texttt{6F}$ (or $190 \cdot 111$ in decimal notation). Write the numbers in base 4:

$$\begin{aligned}\texttt{BE} &= \texttt{2}\cdot 4^3 + \texttt{3}\cdot 4^2 + \texttt{3}\cdot 4 + \texttt{2}\\ \texttt{6F} &= \texttt{1}\cdot 4^3 + \texttt{2}\cdot 4^2 + \texttt{3}\cdot 4 + \texttt{3}\end{aligned}$$

The corresponding polynomials in $\mathbb{Z}[X]$ are

$$\begin{aligned}f(X) &= \texttt{2}X^3 + \texttt{3}X^2 + \texttt{3}X + \texttt{2}\\ g(X) &= \texttt{1}X^3 + \texttt{2}X^2 + \texttt{3}X + \texttt{3}\end{aligned}$$

To find the product $h(X) = f(X)g(X)$ it suffices to compute it modulo 17 and 4, provided the coefficients of the product are less than 68. But each term $h_m = \sum f_i g_{m-i}$ consist of at most four non-zero terms, each at most $3 \cdot 3$. Therefore the coefficients are at most 36, which shows that reduction modulo two is not sufficient, but modulo four does the trick.

We first compute over $\mathbb{Z}_{17}$. The powers of $\omega = \mathtt{2}$ are $\mathtt{1}$, $\mathtt{2}$, $\mathtt{4}$, $\mathtt{8}$, $\mathtt{10} = -\mathtt{1}$, $-\mathtt{2} = \mathtt{F}$, $-\mathtt{4} = \mathtt{D}$, $-\mathtt{8} = \mathtt{9}$ and $\mathtt{1}$. We write the tree for division by factors of $X^8 - \mathtt{1}$:

$$
\begin{array}{cccccccc}
\multicolumn{8}{c}{X^8 - \mathtt{1}} \\
\multicolumn{4}{c}{/} & \multicolumn{4}{c}{\backslash} \\
\multicolumn{4}{c}{X^4 - \mathtt{1}} & \multicolumn{4}{c}{X^4 - \mathtt{10}} \\
\multicolumn{2}{c}{/} & \multicolumn{2}{c}{\backslash} & \multicolumn{2}{c}{/} & \multicolumn{2}{c}{\backslash} \\
\multicolumn{2}{c}{X^2 - \mathtt{1}} & \multicolumn{2}{c}{X^2 - \mathtt{10}} & \multicolumn{2}{c}{X^2 - \mathtt{4}} & \multicolumn{2}{c}{X^2 - \mathtt{D}} \\
/ & \backslash & / & \backslash & / & \backslash & / & \backslash \\
X - \mathtt{1} & X - \mathtt{10} & X - \mathtt{4} & X - \mathtt{D} & X - \mathtt{2} & X - \mathtt{F} & X - \mathtt{8} & X - \mathtt{9}
\end{array}
$$

We apply the FFT algorithm to $f(X)$ and $g(X)$. We write the coefficients of the polynomials sequentially for each step. The transform stands in the last row in bit reversed order. For $f(X)$ we obtain:

$$
\begin{array}{cccccccc}
2 & 3 & 3 & 2 & 0 & 0 & 0 & 0 \\
2 & 3 & 3 & 2 & 2 & 3 & 3 & 2 \\
5 & 5 & -1 & 1 & -3 & -6 & 7 & -5 \\
-7 & 0 & 3 & -5 & 2 & -8 & 1 & -4
\end{array}
$$

and for $g(X)$:

$$
\begin{array}{cccccccc}
3 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \\
3 & 3 & 2 & 1 & 3 & 3 & 2 & 1 \\
5 & 4 & 1 & 2 & -6 & 7 & -5 & -1 \\
-8 & 1 & -8 & -7 & 8 & -3 & 4 & 3
\end{array}
$$

We multiply the elements of the last row and apply the inverse transform, with $\omega^{-1} = \mathtt{9}$, using the first algorithm of the previous section, see (17.6). This means going up in the tree. We find:

$$
\begin{array}{cccccccc}
5 & 0 & -7 & 1 & -1 & 7 & 4 & 5 \\
5 & 5 & -6 & -2 & 6 & -4 & -8 & 2 \\
-1 & 3 & -6 & 7 & -2 & -2 & -5 & 7 \\
-3 & 1 & 6 & -3 & 1 & 5 & -1 & 0
\end{array}
$$

Finally we have to divide by $N = 8$, or multiply with -2, yielding $(6, -2, 5, 6, -2, 7, 2, 0)$ as convolution. The corresponding polynomial in $\mathbb{Z}_{17}[X]$ is

$$\texttt{6} + \texttt{F}X + \texttt{5}X^2 + \texttt{6}X^3 + \texttt{F}X^4 + \texttt{7}X^5 + \texttt{2}X^6 \ .$$

As four is small compared to 17 we easily compute the product $f(X)g(X) \bmod 4$ directly (without FFT):

$$\texttt{2} + \texttt{3}X + \texttt{2}X^2 + \texttt{3}X^3 + \texttt{3}X^4 + \texttt{3}X^5 + \texttt{2}X^6 \ .$$

The coefficient h_3 of X^3 satisfies $h_3 \equiv \texttt{6} \pmod{17}$ and $h_3 \equiv \texttt{3} \pmod{4}$, so it is equal to $\texttt{17}$ ($= 23$ decimally), and $h_2 = \texttt{16}$. We conclude that in $\mathbb{Z}[X]$

$$h(X) = \texttt{6} + \texttt{F}X + \texttt{16}X^2 + \texttt{17}X^3 + \texttt{F}X^4 + \texttt{7}X^5 + \texttt{2}X^6 \ .$$

We set $X = \texttt{4}$ to obtain $fg \in \mathbb{Z}$:

$$\texttt{6} + \texttt{3C} + \texttt{160} + \texttt{5C0} + \texttt{F00} + \texttt{1C00} + \texttt{2000} = \texttt{5262} \qquad (= 21090 \text{ decimally}).$$

Our goal was to compute $fg \bmod 257$. Reduction is easy, because $\texttt{100} \equiv -1 \pmod{257}$, so $fg \equiv \texttt{62} - \texttt{52} = \texttt{10} \pmod{257}$. In fact, to compute $fg \bmod 257$ it would have sufficed to compute the remainder of $h(X)$ upon division by $X^4 + 1$, which is the polynomial

$$r(X) = -\texttt{9} + \texttt{8}X + \texttt{14}X^2 + \texttt{17}X^3 \ .$$

Its reduction modulo 17 can be computed using only the right half of our division tree. In the third line of the inverse Fourier transform we find the sequence $(-2, -2, -5, 7)$, which we have to divide by 4, i.e., multiply by -4: the sequence $(8, 8, 3, 6)$ consists of the coefficients of $r(X) \bmod 17$. In the expression $h_m - h_{m+4} = \sum(f_i g_{m-i} - f_i g_{m+4-i})$ still only four terms appear, so the range of possible values is less than 68, and the Chinese Remainder Theorem still applies. So we can save half of work in the FFTs. □

We now describe the multiplication algorithm. Given two large integers f and g we choose an m with $F_m > fg$. The product $h = fg$ will be computed in $\mathbb{Z}_{F_m}$. We suppose that $m = 2n-1$ is odd (the case $m = 2n-2$ can be handled similarly; alternatively we consider only m of the form $2^e + 1$). We suppose that a fast algorithm for multiplication in $\mathbb{Z}_{F_n}$ is available. We represent the numbers $0, \ldots, F_m - 2$ (multiplication by $-1 \equiv 2^{2^m} \pmod{F_m}$ is easy) as numbers with $N = 2^n$ places in base $2^{2^{n-1}}$. Let $f(X)$ and $g(X)$ be the corresponding polynomials. Let $h(X) = f(X)g(X)$ and set $r(X)$ the remainder of $h(X)$ upon division by $X^N + 1$, i.e., $h(X) = q(X)(X^N + 1) + r(X)$ for some $q(X)$ and $\deg r(X) < N$. The steps in the algorithm are now:

1. Compute $r(X) \bmod F_n$ using the fast Fourier transform for $\mathbb{Z}_{F_n}$ with $\omega = 2$ the primitive $2N$th root of unity and its inverse.

2. Compute $r(X) \bmod 2^n$.

3. Find $r(X)$ with the Chinese remainder theorem and compute $h = r(2^{2^{n-1}})$.

The Chinese remainder theorem can be applied because all coefficients of $r(X)$ lie in a range which is bounded by $2^n \cdot 2^{2^{n-1}} \cdot 2^{2^{n-1}} < 2^n \cdot F_n$, and F_n is odd, so relatively prime to 2^n. As 2^n is much smaller than F_n the computation of $r(X) \bmod 2^n$ can be done with standard algorithms.

Finally we analyse the cost of the algorithm. The main work is in the first step, so we concentrate on that. There are three FFTs, each needing $O(N \log N)$ operations in $\mathbb{Z}_{F_n}$. Some of them involve multiplications, but only multiplication by powers of 2, which can be realised by shifts and subtractions (see exercise 18.6). We operate on sequences of 2^{n-1} bits, so the transforms cost $O(N^2 \log N)$. Let $M(n)$ be the cost in bit operations of multiplication in $\mathbb{Z}_{F_n}$. The component-wise multiplication of two sequences of $N = 2^n$ elements has therefore a cost of $NM(n)$. All together we find

$$M(2n-1) = O(2^{2n}n) + 2^n M(n) \ .$$

We write $c(n) = 2^{-n}M(n)$ and try to solve the equation $c(2n-1) = 2An + 2c(n)$ for some constant A. We first note that $c(2n) = 2An + 2c(n)$ is easy to solve; indeed write $n = 2^k$ and $c(n) = nd(k)$, then $d(k+1) = A + d(k)$ so $d(k) = Ak + B$ and $c(n) = O(n \log n)$. Note that also our original $c(n)$ grows less than quadratic, so in replacing $c(2n-1)$ by $c(2n)$ we make an error which is less than a linear function of n. We conclude that $c(n) = O(n \log n)$ and $M(n) = O(2^{2n} n \log n)$. The bit length of the binary numbers we multiply is at most $N := 2^{2n-2}$, so our multiplication costs $M(N) = O(N \log N \log\log N)$.

EXERCISES

18.1. Compute $189 \cdot 124$ in $\mathbb{Z}_{257}$ using the methods of Example (18.8).

18.2. Show that $\omega = 3$ is a primitive 4th root of unity in the ring $\mathbb{Z}_{16}$. Compute $\sum_{j=0}^{3} \omega^{ij}$ for $i = 0, \ldots, 3$.

18.3. Let ω be an Nth root of unity in the ring R. Show that ω has a multiplicative inverse. Show that $(\omega^M)^{-1} = \omega^{N-m}$ for all $m \in \mathbb{Z}$.

18.4. Let $\omega = 2 \in \mathbb{Z}_{F_n}$, and let $N = 2^{n+1}$. Show that $\omega^0, \omega^1, \ldots, \omega^{\frac{N}{2}-1}$ are represented in $\mathbb{Z}_{F_n}$ as powers of 2. Show that ω is a primitive Nth root of unity.

18.5. Let $N = 2^{n+1}$. Show that in any ring

$$\sum_{j=0}^{N-1} \alpha^j = (1+\alpha)(1+\alpha^2)\cdots(1+\alpha^{\frac{N}{2}})$$

Hint: use the binary representation of j.
In the situation of Prop. 18.7, with $\omega = 2$, consider ω^s. Write $s = 2^u t$ with t odd and show that $(\omega^s)^{2^{n-u}} = -1$ in $\mathbb{Z}_{F_n}$. Conclude that $\sum_{j=0}^{N-1} \omega^{sj} = 0$.
Show that Ω_2^{-1} is as stated.

18.6. Represent elements of $\mathbb{Z}_{F_n}$ in binary notation. Let a be an arbitrary element of $\mathbb{Z}_{F_n}$. Show how to compute the product $a \cdot 2^c$.

APPENDIX: LOGISKA KONNEKTIV OCH KVANTORER

Varje vetenskap, liksom varje yrke, har sitt eget språk som ofta är en blandning av vardagliga ord och speciella termer. En instruktionshandbok för ett kylskåp eller för en dator är full av olika termer som man måste förstå för att kunna ha användning av apparaten. Ibland kan yrkestermer översättas till vardagliga uttryck då man vill förklara något för den oinvigda. Men ofta är en sådan översättning omöjlig. Det är tänkbart att nya vetenskapliga rön i biologi om t ex växternas liv, eller nya forskningsresultat om läkemedel, kan förklaras utan komplicerade facktermer. När det gäller matematik är situationen annorlunda. Det är mycket svårt och egentligen omöjligt att förklara matematiska problem utan det matematiska språket även på en mycket låg nivå. Precis som man lär sig främmande språk för att t ex kunna kommunicera på engelska, måste man lära sig det matematiska språket för kunna använda matematik och diskutera matematik med andra. Precis som med främmande språk lär man sig det matematematiska språket successivt. Samtidigt måste man hela tiden vara medveten om att det är oerhört viktigt att förstå vad orden betyder för att undvika missförstånd och kunna uttrycka sig korrekt. Det matematiska språket består av olika termer och beteckningar. Dessa termer påminner ibland om vardagliga uttryck. Men man måste vara mycket försiktig därför att vardagliga termer kan leda våra associationer i fel riktning. Vi får se i detta avsnitt att t ex sådana ord som "*eller*" eller "*och*" används i matematiska sammanhang i en mycket bestämd mening som ibland avviker från vår vardagliga användning av dessa ord. Samma situation förekommer med främmande språk – vi tror ibland att ett engelskt ord betyder något annat än vad det verkligen gör därför att ordet påminner om ett svenskt ord. I matematiska sammanhang introduceras nya termer och begrepp i form av **definitioner**. Ofta i sådana sammanhang skriver man uttryckligen ordet "definition". Men ibland definieras nya matematiska begrepp i den löpande texten. Vi skall försöka använda fet stil då en ny term introduceras. Detta avsnitt ägnas åt de logiska konnektiven som t ex "*eller*", "*och*", "*om ..., så ...*" samt "*då och endast då*" som mycket ofta används i det matematiska språket. Vi diskuterar också uttrycken "*för alla*" och "*det finns*". Samtidigt introducerar vi några vanliga matematiska beteckningar.

Låt oss börja med ett exempel som visar att betydelsen av ordet "*eller*" i vardagliga situationer kan variera.

(A.1) Exempel. Låt oss betrakta två meningar:

"I kväll läser jag eller går på bio"

Detta påstående består egentligen av två meningar: $p =$ "*I kväll läser jag*" och $q =$ "*I kväll går jag på bio*". I matematiska sammanhang brukar man använda symbolen $\vee$ i stället för "*eller*". Vi kan skriva vårt påstående på formen

$$p \vee q.$$

När är detta påstående sant? Det är klart att det är sant om jag läser i kväll. Det är också sant om jag går på bio i kväll. Men det är också sant då jag både läser och går på bio under kvällen.

Betrakta nu ett annat påstående:

"*I kväll flyger jag till New York eller till Kairo*"

Här har vi också två beståndsdelar $p =$ "*I kväll flyger jag till New York*" och $q =$ "*I kväll flyger jag till Kairo*". Men bindeordet "*eller*" betyder här snarare "*antingen p eller q*" – enligt våra kunskaper om världen endast en av möjligheterna kan inträffa, dvs meningen är sann om exakt en av utsagorna visar sig vara sann. I matematiska sammanhang tolkas betydelsen av "*eller*" alltid i enlighet med det första exemplet. Vi formulerar en exakt definition om en liten stund. □

I fortsättningen betecknar vi meningar eller vad man kallar i matematiska sammanhang **utsagor** med olika bokstäver $a, b, c, ..., p, q, r$. Nu ger vi följande definition:

(A.2) Definition. Om p och q är två utsagor så kallas utsagan "*p eller q*" för **disjunktion**. Den betecknas med $p \vee q$. Disjunktionen $p \vee q$ är sann då minst en av utsagorna p eller q är sann. □

Detta visar att i matematiska sammanhang kommer man överens att sanningen av en utsaga "*p eller q*" tolkas i enlighet med den första möjligheten i exempel (A.1).

Vårt intresse är snarare inriktat på matematiska utsagor som t ex $2+2=4$ eller $2+2=5$. Vi sysslar endast med **utsagor som antingen är sanna eller falska**. Den förutsättningen gäller inte alla utsagor i vardagliga situationer. T ex kan vi inte säga om följande utsaga är sann eller falsk: "Kanske har jag lust att gå på bio". Om en matematisk utsaga p är sann så säger vi att p har **logiska värdet** eller kortare **(sannings)värdet** S (eller ibland 1). Om p är falsk så säger vi att p har logiska värdet F (eller ibland 0). Utsagan $2+2=4$ har sanningsvärdet S, däremot har $2+2=5$ sanningsvärdet F. Ibland kommer vi att skriva $v(p) = S$ om utsagan p är sann, och $v(p) = F$ om den är falsk.

Nu kan vi beskriva värdet av disjunktionen $p \vee q$ beroende på värdena av p och q med hjälp av följande tabell:

p	q	$p \vee q$
F	F	F
F	S	S
S	F	S
S	S	S

Nu övergår vi till ordet "*och*". Här finns det inte någon skillnad mellan den vardagliga betydelsen och den matematiska. Om vi säger

I kväll läser jag och går på bio

så är den utsagan sann precis då både utsagan $p =$ "*I kväll läser jag*" och utsagan $q =$ "*I kväll går jag på bio*" är sanna. En formell definition är följande.

(A.3) Definition. Om p och q är två utsagor så kallas utsagan "p *och* q" för **konjunktion**. Den betecknas med $p \wedge q$. Konjunktionen $p \wedge q$ är sann exakt då både p och q är sanna. □

En tabell som visar sanningsvärdet av $p \wedge q$ beroende på sanningsvärdena av p och q är följande:

p	q	$p \wedge q$
F	F	F
F	S	F
S	F	F
S	S	S

Det är något svårare att hantera en annan mycket vanlig konstruktion: "*om ... så ...*". T ex

Om vädret är bra i kväll, så tar vi en lång promenad"

Här har vi två utsagor $p =$ "*Vädret är bra i kväll*" och $q =$ "*Vi tar en lång promenad*". Med hjälp av p och q konstruerar vi den nya utsagan "*Om p så q*" som kallas **implikation** och betecknas med $p \Rightarrow q$. I stället för "*Om p så q*" använder man ofta andra uttryck som t ex

p *medför* (*att*) q

eller

p *implicerar* (*att*) q.

Innan vi formulerar den exakta definitionen låt oss betrakta följande exempel:

(A.4) Exempel. Låt n beteckna ett heltal, $p(n)$ utsagan "6 *delar* n" och $q(n)$ utsagan "3 *delar* n". Varje utsaga

6 *delar* n *implicerar att* 3 *delar* n

dvs $p(n) \Rightarrow q(n)$ är onekligen sann. Men låt oss testa den utsagan för olika värden på n. Om $n = 12$ så säger den:

6 *delar* 12 *implicerar att* 3 *delar* 12,

om $n = 13$ får vi

$$6 \textit{ delar } 13 \textit{ implicerar att } 3 \textit{ delar } 13,$$

och för $n = 15$:

$$6 \textit{ delar } 15 \textit{ implicerar att } 3 \textit{ delar } 15.$$

Observera att alla dessa utsagor är sanna. Men i första fallet är både $p(12)$ och $q(12)$ sanna, i det andra är både $p(13)$ och $q(13)$ falska, däremot i det tredje är $p(15)$ falsk, men $q(15)$ sann. $\square$

Observera att i det sista exemplet saknas endast fallet då en sann utsaga implicerar en falsk. Detta är också grunden för den exakta definitionen av sanningsvärdet av en implikation nedan – en implikation är falsk endast då en sanning implicerar en osanning. Däremot kan en osanning implicera vad som helst – både sanning och osanning.

(A.5) Definition. Om p och q är två utsagor så kallas utsagan "*om p, så q*" för **implikation**. Den betecknas med $p \Rightarrow q$. Implikationen $p \Rightarrow q$ är falsk enbart då p är sann och q är falsk. $\square$

Tabellen för sanningsvärdet av implikationen $p \Rightarrow q$ är således följande:

p	q	$p \Rightarrow q$
F	F	S
F	S	S
S	F	F
S	S	S

(A.6) Anmärkning. Observera att implikationen $p \Rightarrow q$ alltid är sann då p är falsk. Således är t ex implikationen:

$$(1 = 2) \Rightarrow (2 = 3)$$

sann. Men om en implikation $p \Rightarrow q$ är sann och p är sann så måste även q vara sann. Den observationen spelar en mycket viktig roll i logiska resonemang både i vardagliga situationer och i matematiska sammanhang. Ofta kallar man p för **förutsättning** eller **antagande**. Man kallar q för **slutsats** eller **påstående**. Alltså om förutsättningen är sann och implikationen

$$\textit{förutsättning} \Rightarrow \textit{slutsats}$$

är sann, så är slutsatsen sann. $\square$

(A.7) Anmärkning. Vi har redan noterat att man uttrycker implikationen $p \Rightarrow q$ på flera olika sätt

$$\textit{om } p \textit{ så } q,$$
$$p \textit{ medför (att) } q,$$
$$p \textit{ implicerar (att) } q.$$

Men det finns två andra sätt. Man säger också att

$$p \textit{ är tillräckligt för (att) } q$$

eller

$$q \textit{ är nödvändigt för (att) } p.$$

Försök formulera dessa utsagor med p och q i exempel (A.4) och tänk igenom de så konstruerade meningarna för att inse att även i det vardagliga språket överensstämmer detta uttryckssätt med uttrycken "*medför att*" eller "*implicerar*". □

En annan viktig konstruktion är "*p är ekvivalent med q*", vilket betecknas med $p \Leftrightarrow q$. Uttrycket "*ekvivalent med*" betyder i vardagliga termer att p och q säger samma sak (fast för det mesta på olika sätt). Låt oss även den här gången börja med ett exempel:

(A.8) Exempel. Låt n vara ett godtyckligt heltal, $p(n)$ utsagan "3 *delar* n", och $q(n)$ utsagan "3 *delar summan av siffrorna i* n".

$$3 \textit{ delar } n \textit{ är ekvivalent med att } 3 \textit{ delar summan av siffrorna i } n$$

är en mycket välkänd egenskap. Låt oss testa den då $n = 12$ och $n = 13$. I första fallet har vi

$$3 \textit{ delar } 12 \textit{ är ekvivalent med att } 3 \textit{ delar summan av siffrorna i } 12,$$

medan i det andra

$$3 \textit{ delar } 13 \textit{ är ekvivalent med att } 3 \textit{ delar summan av siffrorna i } 13.$$

Bägge utsagorna är sanna, men i det första fallet är både $p(12)$ och $q(12)$ sanna, medan i det andra är både $p(13)$ och $q(13)$ falska. Detta svarar mot en riktig föreställning om en ekvivalens: sanning är ekvivalent med sanning, och osanning är ekvivalent med osanning. Däremot sanning och osanning är inte ekvivalenta. Detta exempel är grunden för vår nästa definition. □

(A.9) Definition. Om p och q är två utsagor så kallas utsagan "*p är ekvivalent med q*" för **ekvivalens**. Den betecknas med $p \Leftrightarrow q$. Ekvivalensen $p \Leftrightarrow q$ är sann enbart då p och q har samma sanningsvärde. □

Detta betyder att sanningstabellen för ekvivalens är följande:

p	q	$p \Leftrightarrow q$
F	F	S
F	S	F
S	F	F
S	S	S

(A.10) Anmärkning. Ekvivalens $p \Leftrightarrow q$ utläses också på flera olika sätt. I stället för "*p är ekvivalent med q*" säger man t ex

p då och endast då q

eller

p om och endast om q

eller

p är tillräckligt och nödvändigt för q.

□

Vi avslutar med en mycket enkel konstruktion – man tar en utsaga och man formulerar en ny "*ej p*" eller "*det är inte sant att p (gäller)*". Den kallas för negation.

(A.11) Definition. Om p är en utsaga så kallas utsagan "*ej p*" för **negationen** av p. Den betecknas med $\neg p$. Utsagorna p och $\neg p$ har motsatta sanningsvärden. □

Den sista meningen betyder att sanningstabellen för negation är följande:

p	$\neg p$
F	S
S	F

De fem symboler som vi har introducerat: $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\neg$ kallas för **de logiska konnektiven**. Vanligen har man att göra med mera sammansatta utsagor i vilka fler än ett av dessa konnektiv ingår. T ex

$$[(p \wedge q) \vee r] \Rightarrow [(\neg p \vee \neg q) \wedge r]$$

Uttryck av den här typen kallas allmänt för **satsformer**. Precis som tidigare kan man undersöka det logiska värdet av en satsform beroende på sanningsvärdena av de ingående satserna. Låt oss betrakta några exempel:

(A.12) Exempel. (a) Satsformen

$$(p \Rightarrow q) \Rightarrow (q \Rightarrow p)$$

har olika sanningsvärden beroende på sanningsvärdena av p och q. Vi kan studera dessa sanningsvärden med hjälp av följande tabell:

p	q	$p \Rightarrow q$	$q \Rightarrow p$	$(p \Rightarrow q) \Rightarrow (q \Rightarrow p)$
F	F	S	S	S
F	S	S	F	F
S	F	F	S	S
S	S	S	S	S

Vi ser att satsformen är falsk enbart om p är falsk och q är sann.

Man kunde komma fram till den slutsatsen mycket snabbare. Man kan nämligen fråga sig när implikationen ovan är falsk. Vi vet att detta inträffar exakt då $v(p \Rightarrow q) = S$ och $v(q \Rightarrow p) = F$. Men den sista likheten gäller exakt då $v(p) = F$ och $v(q) = S$. Detta är just resultatet av vår studie med hjälp av tabellen ovan.

(b) Nu skall vi undersöka sanningsvärdena av satsformen

$$\neg(p \Rightarrow q) \Leftrightarrow (p \wedge \neg q).$$

Vi gör det med hjälp av en tabell. Du kan försöka göra det utan tabellen genom att ställa frågan när satsformen är falsk (eller sann, men det går snabbare med den första frågan).

p	q	$p \Rightarrow q$	$\neg(p \Rightarrow q)$	$\neg q$	$p \wedge \neg q$	$\neg(p \Rightarrow q) \Leftrightarrow (p \wedge \neg q)$
F	F	S	F	S	F	S
F	S	F	S	F	F	S
S	F	S	F	S	S	S
S	S	S	F	F	F	S

I detta exempel har vi en ekvivalens av två uttryck: $\neg(p \Rightarrow q)$ och $p \wedge \neg q$. Vi kan uppfatta den ekvivalensen så att implikationen $p \Rightarrow q$ är falsk precis då p är sann och q är falsk. Detta visste vi redan tidigare i samband med vår definition av sanningsvärdet hos en implikation.

$\square$

Som vi ser är satsformen i exempel (A.4) (b) alltid sann helt oberoende av vilka sanningsvärden tillskriver man p och q. Sådana satsformer är mycket viktiga därför att de representerar tankemönster som alltid är sanna. Vi antar följande definition:

(A.13) Definition. En satsformel som är sann för alla möjliga uppsättningar av sanningsvärdena av de ingående variablerna kallas en **tautologi** (ibland en **logisk sanning**). En satsformel som är falsk för alla möjliga uppsättningar av sanningsvärdena av de ingående variablerna kallas en **kontradiktion**.

□

Ett exempel på en kontradiktion är

$$p \Leftrightarrow \neg p$$

— en sanning kan inte vara ekvivalent med en osanning.

Möjligheten att kontrollera tautologierna som i exempel (A.4) kan användas för att kontrollera om vissa utsagor är korrekta, t ex då man vill bilda negationen av ett påstående. Betrakta följande exempel.

(A.14) Exempel. Olikheten $1 < x < 5$ kan betraktas som en konjunktion $p \wedge q$ om

$$p = \text{“}x > 1\text{”}$$

och

$$q = \text{“}x < 5\text{”}$$

Vad betyder att $1 < x < 5$ inte gäller? Försök formulera ett svar på denna fråga! Formellt vill vi omformulera utsagan $\neg(p \wedge q)$. En stunds eftertanke säger att om x inte befinner sig mellan 1 och 5 så måste x vara mindre än eller lika med 1, eller också större än eller lika med 5 dvs

$$\neg[(1 < x) \wedge (x < 5)] \Longleftrightarrow [\neg(1 < x) \vee \neg(x < 5)] \Longleftrightarrow (x \leq 1 \vee x \geq 5).$$

Vårt resonemang följer följande tautologi: $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$ som är en av de så kallade **de Morgans lagar** (se nedan). □

Vi lämnar som övningar bevisen av några enkla och viktiga tautologier som används i liknande situationer då man vill bilda negationen av en satsformel. Fler tautologier finns i övningar och i avsnittet om deduktion och induktion.

Den dubbla negationens lag:

$$\neg\neg p \Leftrightarrow p,$$

De Morgans lagar (negationen av en disjunktion och negationen av en konjunktion):

$$\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q),$$
$$\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q).$$

Negationen av en implikation (se Exempel (A.4)):

$$\neg(p \Rightarrow q) \Leftrightarrow (p \wedge \neg q).$$

Negationen av en ekvivalens:

$$\neg(p \Leftrightarrow q) \Leftrightarrow [(p \wedge \neg q) \vee (q \wedge \neg p)].$$

Tautologier används ofta i samband med logiska resonemang såväl i matematiska sammanhang som i vardagliga situationer. Vi skall studera flera exempel i avsnittet om deduktion och induktion, men redan nu kan vi betrakta följande (kriminal-)fall:

(A.15) Exempel. Tre misstänkta personer A, B och C berättar var sin version av en händelse. Om A talar sanning så gör det B också det. Om C ljuger så ljuger även A. Minst en av A, B, C ljuger. Slutsatsen är att A ljuger. Varför?

Låt $p =$ "*A talar sanning*", $q =$ "*B talar sanning*", $r =$ "*C talar sanning*". Vårt påstående säger att implikationen

$$[(p \Rightarrow q) \wedge (\neg r \Rightarrow \neg p) \wedge (\neg(p \wedge q \wedge r))] \Rightarrow (\neg p)$$

är sann samtidigt som vi vet att våra förutsättningar gäller. Om vi lyckas visa att implikationen är en tautologi så visar vi att $\neg p$ måste vara sant dvs A ljuger (se (A.6)).

Är implikationen ovan en tautologi? Vi skall inte studera satsformen med hjälp av sanningstabeller som i exempel (A.4). Låt oss i stället anta att implikationen ovan är falsk. Detta inträffar precis då $v(\neg p) = F$ och $v((p \Rightarrow q) \wedge (\neg r \Rightarrow \neg p) \wedge (\neg(p \wedge q \wedge r))) = S$. Alltså $v(p) = S$ och $v(p \Rightarrow q) = S$, $v(\neg r \Rightarrow \neg p) = S$ samt $v(\neg(p \wedge q \wedge r)) = S$ dvs $v(p \wedge q \wedge r) = F$.

Likheten $v(p \Rightarrow q) = S$ säger att $v(q) = S$ ty $v(p) = S$. Likheten $v(\neg r \Rightarrow \neg p) = S$ säger att $v(\neg r) = F$ ty $v(\neg p) = F$. Alltså är $v(r) = S$. Nu vet vi att $v(p) = v(q) = v(r) = S$ dvs $v(p \wedge q \wedge r) = S$. Vi har fått en motsägelse – om implikationen ovan inte är en tautologi så är $v(p \wedge q \wedge r) = S$ och $v(p \wedge q \wedge r) = F$. Alltså är implikationen en tautologi.

Om Du tycker att vårt resonemang är svårt kan du försöka kontrollera tautologin med hjälp av en tabell (det blir 8 rader i tabellen!). $\square$

Vi avslutar detta avsnitt med några kommentarer om två mycket vanliga uttryck som används i matematiska sammanhang – "*det finns*" och "*för alla*".

(A.16) Exempel. Betrakta två påståenden:

det finns en reell lösning till ekvationen $x^2 - 3 = 0$

och

det finns ett heltal som ligger mellan $1/2$ *och* $3/2$.

Dessa påståenden nedtecknas på följande sätt:

$$\exists_{x \in \mathbb{R}} \quad x^2 - 3 = 0$$

och

$$\exists_{x \in \mathbb{Z}} \quad \tfrac{1}{2} < x < \tfrac{3}{2}.$$

Symbolen $\exists$ betyder just "**det finns**". Observera att vi skriver något nersänkt, liksom index, var vi befinner oss — i första fallet säger vi att det finns ett reellt tal x, och i det andra, att det finns ett heltal x. Användningen av bokstaven x har ingen betydelse. Vi kunde lika gärna byta x mot en annan bokstav. När man utläser symbolen $\exists$ med efterföljande text så använder man vanligen frasen "*det finns ... sådant att ...*". T ex säger första påståendet att

"*det finns ett reellt tal* x *sådant att* $x^2 - 3 = 0$"

och det andra att

"*det finns ett heltal* x *sådant att* $\frac{1}{2} < x < \frac{3}{2}$"

Som Du säkert märker avviker det formella språket från våra ursprungliga uttryck som dock säger exakt samma sak. Symbolen $\exists$ kallas **existenskvantor**. □

(A.17) Exempel. Betrakta nu två påståenden som använder frasen "*för alla*" (ibland "*för varje*" eller "*varje*"):

för varje reellt tal x *gäller det att* $x^2 + 1 > 0$

och

alla heltal är delbara med 2

(det andra påståendet är helt enkelt inte sant, men det har inte någon betydelse för våra exempel). Nu använder vi en annan symbol: $\forall$ som utläses "**för alla"** (ibland "**för varje**) och kallas **allkvantor**. Med hjälp av denna kvantor skriver vi:

$$\forall_{x \in \mathbb{R}} \quad x^2 + 1 > 0$$

och

$$\forall_{n\in\mathbb{N}} \quad 2|n$$

Rent formellt utläser vi dessa symboler så här:

för alla reella tal x gäller det att $x^2 + 1 > 0$

och

för alla heltal n gäller det att 2 *dividerar n*

Det är bara ett annat sätt att säga samma sak som tidigare, att alla heltal är jämna. Vi har ändrat formuleringen för att skriva det hela kortare med hjälp av en matematisk symbol. □

Hur bygger man negationer av uttryck som innehåller allkvantorn eller existenskvantorn? Betrakta ett exempel.

(A.18) Exempel. (a) Låt X vara mängden av alla elever i en skola. Om vi säger att det finns en elev i skolan, säg x, som pratar franska, vad är negationen av detta påstående? Man kan säga att ingen av eleverna i skolan pratar franska. Om vi vill använda det matematiska uttrycket "*för varje*" (eller "*för alla*") så kan vi formulera oss så att "*för varje elev x i skolan X, x pratar inte franska*". Vi kan gå ett steg längre i våra formaliseringssträvanden. Låt $f(x)$ betyda just att "*x pratar franska*". Då hade vi

$$\exists_{x\in X} \quad f(x)$$

och

$$\neg\exists_{x\in X} \quad f(x)$$

betyder att

$$\forall_{x\in X} \quad \neg f(x).$$

Detta är just den allmänna metoden att bilda negationen av uttrycket $\exists_{x\in X} f(x)$ dvs vi har tautologin:

$$\neg\exists_{x\in X} \quad f(x) \qquad \Longleftrightarrow \qquad \forall_{x\in X} \quad \neg f(x).$$

(b) Vi behåller samma beteckningar och påstår att alla elever i skolan X pratar franska. Med samma beteckningar som ovan nedtecknar vi vårt påstående som

$$\forall_{x\in X} \quad f(x).$$

Vad betyder negationen av detta påstående? Helt klart betyder det att det finns (minst) en elev i skolan som inte pratar franska. Alltså betyder:

$$\neg\forall_{x\in X} \quad f(x)$$

att

$$\exists_{x\in X}\neg f(x).$$

Vi antecknar den allmänna tautologin:

$$\neg\forall_{x\in X} \quad f(x) \qquad \Longleftrightarrow \qquad \exists_{x\in X} \quad \neg f(x).$$

□

Vi skall avsluta detta avsnitt med exempel som visar att man måste vara mycket försiktig då man kastar om uttrycken "*det finns*" och "*för alla*".

(A.19) Exempel. Låt X vara mängden av alla gifta kvinnor i Göteborg och Y mängden av alla gifta män i samma stad. Låt $x \in X$ och $y \in Y$. Beteckna med $f(x, y)$ utsagan "*x är gift med y*". Vad säger påståendet

$$\forall_{x\in X}\exists_{y\in Y} \quad f(x, y) \text{ ?}$$

Det säger att för varje gift kvinna i Göteborg finns en gift man i Göteborg så att de är gifta – ett rimligt påstående som dock inte behöver vara sant. Vad säger

$$\exists_{y\in Y}\forall_{x\in X} \quad f(x, y) \text{ ?}$$

Den här gången får vi att det finns en man i Göteborg som är gift med alla kvinnor i staden. Alltså var försiktig då kvantorerna skall placeras! □

ÖVNINGAR

A.1. Visa att följande satsformer är tautologier:

(a) $\neg(\neg p \wedge p)$ (motsägelselagen),

(b) $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$ (transpositionslag),

(c) $\neg p \Rightarrow (p \Rightarrow q)$ (Duns–Scotus lag),

(d) $(p \wedge q) \Rightarrow p$,

(e) $(p \wedge q \Rightarrow r) \Leftrightarrow [p \Rightarrow (q \Rightarrow r)]$,

(f) $[p \wedge (q \vee r)] \Leftrightarrow [(p \wedge q) \vee (p \wedge r)]$.

A.2. Vilka av följande satsformer är tautologier?

(a) $[(p \vee q) \wedge \neg p] \Rightarrow q$;

(b) $[(p \Rightarrow q) \wedge (q \Rightarrow p)] \Rightarrow (p \vee q)$,

(c) $[(p \Rightarrow q) \wedge (q \Rightarrow r)] \Rightarrow (p \Rightarrow r)$.

A.3. Man definierar **Sheffers streck** "|" med hjälp av sanningstabellen:

p	q	$p \vert q$
F	F	S
F	S	S
S	F	S
S	S	F

(a) Motivera att $p|q \Leftrightarrow \neg(p \wedge q)$.

(b) Uttryck $\neg p$, $p \vee q$ och $p \wedge q$ med hjälp av Sheffers streck.

A.4. Bilda negationen av följande meningar och använd kvantorer för att nedteckna både dessa meningar och deras negationer:

(a) Äpplen är röda;

(b) Varje pojke tycker om en flicka;

(c) Varje hund har en svans;

A.5. Är följande resonemang riktiga?

(a) Låt l, m, p vara tre räta linjer i planet. Om det inte är sant att l är parallell med m eller p inte är parallell med m, så är l parallell med m eller p är parallell med m.

(b) Kajsa kan logik då och endast då det är inte sant att det är inte sant att Kajsa kan logik.