Matematiska Institutionen, KTH
VT 1996
Anders Björner

KINESISKA RESTSATSEN och STRUKTURSATSER

I vissa fall kan algebraiska uträkningar delas upp på flera mindre uträkningar som kan utföras "parallellt" och sedan sättas samman för att ge svaret. Vi skall här studera några viktiga fall av detta.

1. Kinesiska Restsatsen. Följande resultat lär ha varit känt i Kina på 1200-talet (och i specialfall redan 1000 år tidigare):

Sats 1. Låt $m_1, m_2, \ldots, m_k$ vara positiva tal så att $SGD(m_i, m_j) = 1$ för alla $1 \leq i < j \leq k$. Då har systemet av kongruenser (där alla $a_i \in \mathbb{Z}$)

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \cdot \\ \cdot \\ \cdot \\ x \equiv a_k \pmod{m_k} \end{cases} \tag{1}$$

en lösning, och denna är unik modulo $m = m_1\, m_2 \ldots m_k$.

Bevis. Låt $M_i = \prod\limits_{j \neq i} m_j = \frac{m}{m_i}$. Av $SGD(M_i,\, m_i) = 1$ följer existens av tal b_i sådana att $b_i\, M_i \equiv 1 \pmod{m_i}$, $1 \leq i \leq k$. Låt

$$x = a_1\, b_1\, M_1 + a_2\, b_2\, M_2 + \cdots + a_k\, b_k\, M_k. \tag{2}$$

Eftersom

$$b_j\, M_j \equiv \begin{cases} 1 \pmod{m_j} \\ 0 \pmod{m_i}\ , i \neq j\ , \end{cases}$$

följer

$$x = a_1 \underbrace{b_1 M_1}_{\equiv 0} + \cdots + a_i \underbrace{b_i M_i}_{\equiv 1} + \cdots + a_k \underbrace{b_k M_k}_{\equiv 0} \equiv a_i \pmod{m_i},$$

så x löser systemet (1).

Om även x' är en lösning, så $x - x' \equiv a_i - a_i = 0 \pmod{m_i} \Rightarrow m_i \big| (x - x')$, för $\forall\ 1 \leq i \leq k$. Men eftersom m_i är relativt prima måste då $\underbrace{m_1\, m_2\, \ldots m_k}_{= m} \big| (x - x')$, d.v.s. $x \equiv x' \pmod{m}$. $\square$

Algoritmisk aspekt. Ovanstående bevis ger samtidigt en effektiv algoritm att praktiskt lösa system av typ (1). Arbetet ligger framför allt i att bestämma talen b_i, de multiplikativa inverserna till M_i modulo m_i. Detta kan göras effektivt med Euklides algoritm.

Märk att om flera system av typ (1) med olika högerled $a = (a_1, \ldots, a_k)$ skall lösas, så behöver talen b_i bara bestämmas en gång. "Vikterna" $y_i = b_i\,M_i$ kan sedan lagras i datorns minne och varje individuellt system (1) har enl. (2) lösningen

$$x = a_1\,y_1 + a_2\,y_2 + \cdots + a_k\,y_k \pmod{m}.$$

Exempel. Vi löser systemet:

$$\begin{cases} x \equiv 1 \pmod 4 \\ x \equiv 2 \pmod 3 \\ x \equiv 4 \pmod 5 \end{cases} \tag{3}$$

Vi har $M_1 = 3 \cdot 5 = 15$, $M_2 = 4 \cdot 5 = 20$, $M_3 = 4 \cdot 3 = 12$, och vill lösa

$$\begin{cases} y_1 = 15 \cdot b_1 \equiv 1 \pmod 4 \\ y_2 = 20 \cdot b_2 \equiv 1 \pmod 3 \\ y_3 = 12 \cdot b_3 \equiv 1 \pmod 5 \end{cases} \tag{4}$$

Talen b_1, b_2, b_3 kan som nämnts alltid bestämmas med Euklides algoritm, men i sådana här enkla fall går det i regel snabbare att direkt hitta lösningen genom "trial and error". Genom att söka igenom små multipler av 15, 20 resp. 12, hittar vi följande lösning

$$\begin{cases} y_1 = -15 \\ y_2 = -20 \\ y_3 = 36 \end{cases}$$

varav: $x = 1 \cdot (-15) + 2 \cdot (-20) + 4 \cdot 36 = 89 \equiv 29 \pmod{60}$.

Svar: $x \equiv 29 \pmod{60}$.

Kommentarer: 1. Talen b_1, b_2, b_3 och därmed y_1, y_2, y_3 är inte entydigt bestämda. Exempelvis skulle $y_1 = 45$, $y_2 = 40$ och $y_3 = -24$ lika gärna kunnat användas. Om b_1, b_2, b_3 och b_1', b_2', b_3' båda löser systemet (4) gäller att $b_1 \equiv b_1' \pmod 4$, $b_2 \equiv b_2' \pmod 3$ och $b_3 \equiv b_3' \pmod 5$

2. Märk att lösningen till systemet (3) bara kan bestämmas modulo $60 = 4 \cdot 3 \cdot 5$. Vi kan alltså inte utan ytterligare information veta om $x = -31$, $x = 29$, $x = 89$, eller något av oändligt många andra möjliga x-värden är en i något särskilt fall avsedd lösning.

2. Struktursats för $\mathbb{Z}_n$.

Vi börjar med att definiera begreppen isomorfi och direkt produkt för ringar. Motsvarande begrepp för grupper diskuteras i Biggs 13.5 och 13.6.

En avbildning $f : R_1 \to R_2$ mellan två ringar kallas en *ringisomorfi* om

(i) f är en bijektion (d.v.s., injektion + surjektion)
(ii) $f(a+b) = f(a) + f(b), \quad \forall a, b \in R_1$,
(iii) $f(ab) = f(a)f(b), \quad \forall a,\, b \in R_1$.

Ringarna R_1 och R_2 är då ur matematisk synpunkt helt identiska, och kallas *isomorfa.* (Övning: Kontrollera att $f(0) = 0\,, f(1) = 1\,, f(-a) = -f(a)$, och om a^{-1} existerar för $a \in R_1$, så $f(a^{-1}) = f(a)^{-1}$.)

Låt R_1, $R_2, \ldots, R_k$ vara ringar. Det enklaste sättet att sätta ihop dem till en större ring är att ta den direkta (Cartesiska) produkten som mängder

$$R_1 \times R_2 \times \cdots \times R_k = \{(a_1,\ a_2, \ldots, a_k)\,|a_i \in R_i\,,\, 1 \le i \le k\}$$

och på den definiera addition och multiplikation koordinatvis

$$\big(a_1, \ldots, a_k\big) + \big(b_1, \ldots, b_k\big) = \big(a_1 + b_1, \ldots, a_k + b_k\big)$$
$$\big(a_1, \ldots, a_k\big) \cdot \big(b_1, \ldots, b_k\big) = \big(a_1 \cdot b_1, \ldots, a_k \cdot b_k\big).$$

Detta gör $R_1 \times R_2 \times \cdots \times R_k$ till en ring (Övning: verifiera ringaxiomen) som kallas den *direkta produkten* av ringarna R_i.

Exempel. $R = \mathbb{Z}_8 \times \mathbb{Z}_9 \times \mathbb{Z}_5$ är en ring med $8 \cdot 9 \cdot 5 = 360$ element. Här är två exempel på algebraiska operationer i R :

$$(4, 6, 3) + (6, 6, 2) = (2, 3, 0)$$
$$(4, 6, 3) \cdot (6, 6, 2) = (0, 0, 1)$$

(Obs! I första koordinaten räknar vi modulo 8, i andra modulo 9, etc.)

Sats 2. Antag att $n = p_1^{e_1}\, p_2^{e_2} \ldots p_k^{e_k}$ är primfaktorsuppdelningen av talet n. Då är avbildningen

$$f : \mathbb{Z}_n \longrightarrow \mathbb{Z}_{p_1^{e_1}} \times \mathbb{Z}_{p_2^{e_2}} \times \cdots \times \mathbb{Z}_{p_k^{e_k}}$$

definierad av $\big[a\big]_n \mapsto \Big(\big[a\big]_{p_1^{e_1}}\,, \big[a\big]_{p_2^{e_2}}, \ldots, \big[a\big]_{p_k^{e_k}}\Big)$ en ringisomorfi.

Bevis. $\big[a\big]_n = \big[b\big]_n \Longleftrightarrow a \equiv b \pmod{n}$
$\Longleftrightarrow a \equiv b \ \ \big(\text{mod } \ p_i^{e_i}\big) \quad \forall\, 1 \le i \le k$
$\Longleftrightarrow \big[a\big]_{p_i^{e_i}} = \big[b\big]_{p_i^{e_i}} \quad , \forall\, 1 \le i \le k$
$\Longleftrightarrow f\big(\big[a\big]_n\big) = f\big(\big[b\big]_n\big).$

De framåtriktade implikationerna $\Rightarrow$ visar att avbildningen f är väldefinierad. De bakåtriktade implikationerna $\Leftarrow$ visar att f är injektiv.

För att visa att f är en ringisomorfi måste vi verifiera:
(i) *f är en bijektion:* Eftersom f är injektiv och de två ringarna har samma antal element måste f vara en bijektion.
(Kommentar: Detta resonemang är snabbt men dåligt ur den synpunkten att det inte ger någon algoritm för beräkning av den inversa funktionen f^{-1}. Men en sådan alogritm har vi redan: Kinesiska restsatsen! Mer om detta strax.)

(ii) *f respekterar addition:*

$$\begin{aligned}
f\big(\big[a\big]_n + \big[b\big]_n\big) &= f\big(\big[a+b\big]_n\big) \\
&= \Big(\big[a+b\big]_{p_1^{e_1}}, \ldots, \ \big[a+b\big]_{p_k^{e_k}}\Big) \\
&= \Big(\big[a\big]_{p_1^{e_1}} + \big[b\big]_{p_1^{e_1}}, \ldots, \ \big[a\big]_{p_k^{e_k}} + \big[b\big]_{p_k^{e_k}}\Big) \\
&= \Big(\big[a\big]_{p_1^{e_1}}, \ldots, \big[a\big]_{p_k^{e_k}}\Big) + \Big(\big[b\big]_{p_1^{e_1}}, \ldots, \big[b\big]_{p_k^{e_k}}\Big) \\
&= f\big(\big[a\big]_n\big) + f\big(\big[b\big]_n\big).
\end{aligned}$$

(iii) *f respekterar multiplikation:* Helt analogt. ☐

Tillämpad aspekt. Om $p_i^{e_i}$ är mycket mindre än n så sker beräkningar i ringen $\mathbb{Z}_{p_i^{e_i}}$ snabbare än i $\mathbb{Z}_n$. Eftersom aritmetiken i en dator sker i $\mathbb{Z}_n$ för något mycket stort tal n (och ej i $\mathbb{Z}$, av det enkla skälet att den oändliga mängden $\mathbb{Z}$ ej kan representeras i en dator), så visar Sats 2 på en möjlighet att utföra aritmetiken i många små ringar $\mathbb{Z}_{p_i^{e_i}}$. T.ex. kan $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k} > 5 \cdot 10^{42}$ väljas så att $p_i^{e_i} < 100$ för $\forall\, 1 \leq i \leq k$.

Till exempel, om $a \cdot b$ skall beräknas (modulo n) så kan vi använda följande algoritm:

$$\begin{array}{lcl}
[a]_n & \xrightarrow{f} & \left([a]_{p_1^{e_1}}, \dots, [a]_{p_k^{e_k}}\right) \\
[b]_n & \xrightarrow{f} & \left([b]_{p_1^{e_1}}, \dots, [b]_{p_k^{e_k}}\right) \\
 & \times & \\
\hline
[c]_n & \xleftarrow{f^{-1}} & \left([c_1]_{p_1^{e_1}}, \dots, [c_k]_{p_k^{e_k}}\right) \quad \text{med koordinatvis multiplikation}
\end{array}$$

vilket ger $a \cdot b \equiv c \,(\mathrm{mod}\ n)$. Beräkning av f kan ske effektivt med divisionsalgoritmen (division av a med respektive $p_i^{e_i}$ ger som rest $i-te$ koordinaten av $f\left([a]_n\right)$), och beräkning av f^{-1} sker effektivt med kinesiska restalgoritmen.

Om stora mängder additioner, subtraktioner och multiplikationer av stora tal skall utföras i $\mathbb{Z}_n$ kan avsevärda tidsvinster göras med denna metod. Alla ingående tal överförs via f till $\mathbb{Z}_{p_1^{e_1}} \times \dots \times \mathbb{Z}_{p_k^{e_k}}$, där sedan alla operationer görs. Slutligen överförs svaret via f^{-1} tillbaks till $\mathbb{Z}_n$. Denna metod är snabb av två skäl:

1) beräkningar i $\mathbb{Z}_{p_i^{e_i}}$ är snabbare än i $\mathbb{Z}_n$

2) beräkningarna i de olika "små" ringarna $\mathbb{Z}_{p_i^{e_i}}$ kan eventuellt utföras *parallellt.*

Lägg märke till att för fixt $n = p_1^{e_1} \dots p_k^{e_k}$ så är moduli $p_i^{e_i}$, $1 \leq i \leq k$, också fixa, och beräkning av f^{-1} med kinesiska restalgoritmen kan göras ytterst snabbt med förlagrade vikter y_i (se kommentaren efter Sats 1).

Det bör påpekas att den skisserade metoden för aritmetik i $\mathbb{Z}_n$ inte är bra om annat än ringoperationer ingår, t.ex. vid division och storleksjämförelser mellan heltal. För en detaljerad diskussion av denna och andra metoder för så kallad *snabb aritmetik*, se Knuth (1981).

Tyvärr är det svårt att exemplifiera ovanstående med något realistiskt exempel. Principen framgår ändå av följande (något "orealistiska") problem.

Exempel. Beräkna determinanten

$$D = \begin{vmatrix} 27 & 8 & -10 \\ 5 & 19 & 13 \\ -26 & 25 & 14 \end{vmatrix} \quad (\mathrm{mod}\ 30)$$

Determinanten beror bara av ringoperationer (multiplikationer och additioner, tänk på definitionen) så vi kan utnyttja ringisomorfin i Sats 2 med $30 = 2 \cdot 3 \cdot 5$:

$$f\big([D]_{30}\big) = \left(\begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix}_2 , \begin{vmatrix} 0 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & 1 & -1 \end{vmatrix}_3 , \begin{vmatrix} 2 & 3 & 0 \\ 0 & -1 & 3 \\ -1 & 0 & -1 \end{vmatrix}_5 \right)$$

$$= \big([1]_2 \;, [2]_3 \;, [3]_5\big).$$

Här har determinanterna beräknats med gängse regler från linjär algebra men modulo 2, 3 resp. 5.

Exempelvis:

$$\begin{vmatrix} 2 & 3 & 0 \\ 0 & -1 & 3 \\ -1 & 0 & -1 \end{vmatrix}_5 = \begin{vmatrix} 2 & 0 & 0 \\ 0 & -1 & 3 \\ -1 & -1 & -1 \end{vmatrix}_5 = 2 \cdot \begin{vmatrix} -1 & 3 \\ -1 & -1 \end{vmatrix}_5 = 2(1+3) \equiv_5 3,$$

där vi i första steget adderat kolonnerna 1 och 2.

Att beräkna $f^{-1}\big([1]_2 \,, [2]_3 \,, [3]_5\big)$ är ekvivalent med att lösa systemet av kongruenser

$$\begin{cases} x \equiv 1 \pmod 2 \\ x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \end{cases} \tag{5}$$

vilket vi kan göra med kinesiska restalgoritmen:

$$\text{Systemet} \begin{cases} y_1 &= 15 \cdot b_1 \equiv 1 \pmod 2 \\ y_2 &= 10 \cdot b_2 \equiv 1 \pmod 3 \\ y_3 &= \;\;6 \cdot b_3 \equiv 1 \pmod 5 \end{cases}$$

har en lösning $y_1 = 15,\ y_2 = 10,\ y_3 = 6$,
så $x = 1 \cdot 15 + 2 \cdot 10 + 3 \cdot 6 = 53 \equiv 23 \pmod{30}$ löser systemet (5).

Alltså: $f^{-1}\big([1]_2 \,, [2]_3, \ [3]_5\big) = [23]_{30}$, vilket ger svaret: $D \equiv 23 \pmod{30}$.

Avslutningsvis vill vi påpeka att snabb aritmetik av detta slag (Sats 2 + Kinesiska Restsatsen) är mycket användbar för exakt lösning av stora linjära ekvationssystem med heltalskoefficienter. Detta finns beskrivet i Kapitel 1 av Mackiw (1985).

3. Struktursats för ändliga Abelska grupper. Om vi bara betraktar den additiva strukturen så säger Sats 2 att

$$\mathbb{Z}_n \cong \mathbb{Z}_{p_1^{e_1}} \times \cdots \times \mathbb{Z}_{p_k^{e_k}}$$

som additiva grupper. Därav sluter vi att varje ändlig cyklisk grupp är isomorf med en direkt produkt av cykliska grupper av primtalspotens-ordning. Detsamma gäller i själva verket för *varje* ändlig Abelsk grupp, enligt följande struktursats som varit känd sedan slutet av 1800-talet.

Sats 3. Låt G vara en ändlig Abelsk grupp. Då existerar en unik multimängd primtalspotenser $\{p_1^{e_1}, p_2^{e_2}, \ldots, p_k^{e_k}\}$ så att

$$G \cong \mathbb{Z}_{p_1^{e_1}} \times \mathbb{Z}_{p_2^{e_2}} \times \cdots \times \mathbb{Z}_{p_k^{e_k}}. \tag{6}$$

Beviset för Sats 3 ligger utanför ramen för denna kurs. Observera att det här är tillåtet att $p_i = p_j$ och t.o.m. $p_i^{e_i} = p_j^{e_j}$ för $i \neq j$, därför använder vi ordet ”multimängd” i stället för mängd.

Av (6) följer att $|G| = p_1^{e_1}\, p_2^{e_2} \ldots p_k^{e_k}$. Av Sats 3 följer därför att man lätt kan ange alla tänkbara isomorfityper för Abelska grupper av ordning n (det finns en sådan typ för varje sätt att skriva n som produkt av primtalspotenser).

Exempel. (1) Varje Abelsk grupp av ordning 4 är isomorf med $\mathbb{Z}_4$ eller med $\mathbb{Z}_2 \times \mathbb{Z}_2$. (I detta fall kan ordet ”Abelsk” tas bort, se övning 13.5.2 och 13.6.4 i Biggs.)

(2) Varje Abelsk grupp av ordning 100 är isomorf med en av följande fyra grupper:

$$\begin{array}{lllllll} \mathbb{Z}_4 & \times & \mathbb{Z}_{25} & & & & \\ \mathbb{Z}_2 & \times & \mathbb{Z}_2 & \times & \mathbb{Z}_{25} & & \\ \mathbb{Z}_4 & \times & \mathbb{Z}_5 & \times & \mathbb{Z}_5 & & \\ \mathbb{Z}_2 & \times & \mathbb{Z}_2 & \times & \mathbb{Z}_5 & \times & \mathbb{Z}_5. \end{array}$$

(Vilken av dessa är isomorf med $\mathbb{Z}_{100}$? Svar (enligt Sats 2): $\mathbb{Z}_4 \times \mathbb{Z}_{25}$.)

Avslutande kommentar. Satserna 2 och 3 är exempel på en typ av satser som är mycket viktiga i matematiken. De säger att vissa mer komplicerade strukturer på ett konkret sätt är uppbyggda av enklare strukturer (som molekyler är uppbyggda av atomer). För att förstå de komplicerade strukturerna kan man då inrikta sig på att dels förstå de enklare strukturerna (för dessa kan det krävas en *klassifikation*) och dels på hur dessa enkla strukturer sätts samman till de ”komplicerade” strukturerna (beskrivet av en *struktursats*). Sats 3 säger exempelvis att de ”enkla” Abelska grupperna är de cykliska grupperna av primtalspotensordning. Andra välkända exempel är:

(1) Aritmetikens Fundamentalsats säger att varje positivt heltal n på ett unikt sätt kan skrivas som produkt av primtal. Denna välkända struktursats säger speciellt att de ”enkla” heltalen är primtalen.

(2) Algebrans Fundamentalsats säger att varje komplext moniskt polynom $p(z)$ på ett unikt sätt kan skrivas som produkt av moniska förstagradspolynom $p(z) = (z - a_1) \ldots (z - a_n)$, $a_i \in \mathbb{C}$.

(3) Betrakta mängden av alla reella funktioner $f : \mathbb{R} \to \mathbb{R}$ med period 2π (d.v.s. $f(x + 2\pi) = f(x)\,, \forall x \in \mathbb{R}$). Om en sådan funktion är tillräckligt välartad (exempelvis om dess derivata existerar och är kontinuerlig överallt) så gäller enligt en sats av Dirichlet-Fourier

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx),$$

för alla $x \in \mathbb{R}$. Detta kan uppfattas som en struktursats som säger att komplicerade periodiska funktioner f (t.ex. vibrationer) alltid är uppbyggda av enkla harmoniska svängningar $\cos kx$ och $\sin kx$ genom överlagring.

Om nu de Abelska gruppernas struktur är känd, åtminstone i det ändliga fallet, så är det naturligt att fråga om något motsvarande resultat finns i det icke-kommutativa fallet. Svaret är i stort sett negativt, inga allmänna struktursatser finns. Men i två fall är situationen ändå mycket positiv.

(1) För vissa viktiga klasser av geometriska transformationsgrupper (Lie-grupper, Coxeter-grupper) finns detaljerade struktursatser och klassifikation av de ”enkla” grupperna inom klasserna. Studiet av sådana grupper har varit en mycket viktig gren av 1900-tals-matematiken.

(2) I början av 1980-talet fullbordades en klassifikation av alla ändliga ”enkla” grupper, som sysselsatt många matematiker i årtionden. En fullständig utskrift av alla delar av denna klassifikation (med bevis) sägs kräva 10.000-20.000 sidor. Begreppet ”enkel” har här en något svagare innebörd än i Sats 3, och godtyckliga ändliga grupper är sammansatta av dessa enkla grupper på ett mer komplicerat sätt (inte som direkt produkt).

Referenser

D.E. Knuth: *The art of computer programming, Vol. 2 (Seminumerical algorithms)*, Addison-Wesley, 1969, 1981.

G. Mackiw: *Applications of abstract algebra*, Wiley & Sons, New York, 1985.

Övningar:

(1) Bestäm alla tal x sådana att $300 \leq x \leq 600$ och

$$\begin{cases} x & \equiv 1 \pmod{5} \\ x & \equiv 3 \pmod{6} \\ x & \equiv 5 \pmod{7} \end{cases}$$

(2) Bestäm alla tal x sådana att $0 \leq x \leq 2000$ och

$$\begin{cases} x & \equiv 1 \pmod{7} \\ x & \equiv 6 \pmod{11} \\ x & \equiv 5 \pmod{13} \end{cases}$$

(3) Låt $f : \mathbb{Z}_{60} \to \mathbb{Z}_3 \times \mathbb{Z}_4 \times \mathbb{Z}_5$ vara avbildningen i Sats 2.
   (a) Bestäm $f(1)$ och $f(10)$.
   (b) Bestäm $f^{-1}\big((1,0,0)\big)$, $f^{-1}\big((0,1,0)\big)$, $f^{-1}\big((0,0,1)\big)$.
   (c) Använd den metod för ”snabb aritmetik” som skisserades att med hjälp av f och f^{-1} beräkna

$$2(11+4)+19+7 \quad \text{i } \mathbb{Z}_{60}.$$

(4) Hur många isomorfityper av Abelska grupper av ordning 24 finns det? Ge ett exempel av varje typ.

(5) Den multiplikativa gruppen $\mathbb{Z}_{12}^*$ av invertibla element i $\mathbb{Z}_{12}$ är Abelsk. Bestäm den produkt av cykliska grupper (som i Sats 3) med vilken den är isomorf.

(6) Visa att varje Abelsk grupp av ordning 30 är cyklisk.

Svar:

(1) 411

(2) 512 och 1513

(3) (a) $(1,1,1)\,,(1,2,0)$

    (b) $40, 45, 36.$

(4) 3 isomorfityper: $\mathbb{Z}_8 \times \mathbb{Z}_3\,, \mathbb{Z}_4 \times \mathbb{Z}_2 \times \mathbb{Z}_3\,, \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3.$

(5) $\mathbb{Z}^*_{12} = \{[1], [5]\,, [7]\,, [11]\}$ och alla element har ordning 2, så $\mathbb{Z}^*_{12} \cong \mathbb{Z}_2 \times \mathbb{Z}_2.$

(6) $30 = 2 \cdot 3 \cdot 5$ så enligt Sats 3 existerar bara en isomorfiklass, nämligen för den cykliska gruppen $\mathbb{Z}_{30} \cong \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_5.$