Notater for forelesning 9/9

I dag skal vi ta for oss *Euklids algoritme*, som er en svært gammel metode for å beregne største felles divisor av to tall. Denne algoritmen vil være sentral i dette kurset, og derfor vil jeg formulere resultatet som et *teorem*, i motsetning til hvordan det gjøres i boken.

Teorem 0.1 (Euklids algoritme)**.** *La a og b være to positive tall. Da kan vi beregne den største felles divisoren $d = gcd(a, b)$ på følgende måte: Definer r_1 og q_1 ved å bruke divisjonsalgoritmen på a og b: $a = q_1 b + r_1$. Definer r_2 og q_2 ved å bruke divisjonsalgoritmen på b og r_1: $b = q_2 r_1 + r_2$. Induktivt definerer vi så r_k og q_k ved å bruke divisjonsalgoritmen på r_{k-2} og r_{k-1}: $r_{k-2} = q_k r_{k-1} + r_k$. Før eller siden vil denne prosessen stoppe, ved at en $r_{n+1} = 0$, altså ved $r_{n-1} = q_n r_n + 0$. Da er $d = gcd(a, b) = r_n$, eller om allerede $r_1 = 0$, så er $gcd(a, b) = b$.*

Utsagnet her er kanskje ikke så lett å forstå, så vi starter med et skikkelig eksempel. La oss bruke algoritmen for å beregne $gcd(1326, 231)$. Utregningen ser ut som følger (å utføre divisjonene er mellomregning og tas ikke med når slike oppgaver føres):

$$
\begin{aligned}
1326 &= 5.231 + 171 \\
231 &= 1.171 + 60 \\
171 &= 2.60 + 51 \\
60 &= 1.51 + 9 \\
51 &= 5.9 + 6 \\
9 &= 1.6 + 3 \\
6 &= 2.3 + 0
\end{aligned}
$$

Konklusjonen er altså at $gcd(1326, 231) = 3$. Vi stopper når divisjonen går opp, og den siste resten som er forskjellig fra null, altså 3 i nest siste linje, er tallet vi er ute etter.

La oss så gå over til beviset. Det er flere ting som må sjekkes her, la oss først se på hvorfor algoritmen stopper. Ved divisjon vil vi alltid kreve at resten er mindre enn tallet det deles på. For eksempel er $r_1 < b$ og $r_2 < r_1$. Generelt vil $r_k < r_{k-1}$ for alle k inntil det stopper, og disse tallene er også alle positive. Men når vi velger et stadig mindre positivt heltall, vil det til slutt stoppe. Hvor lang tid det tar før det stopper, avhenger selvsagt av a og b.

Induksjonen i beviset går på antall skritt i algoritmen, altså på tallet n i teoremet. Starten på induksjonen følger fra dette lemmaet:

Lemma 0.2. *Om $a = qb + r$ er $gcd(a, b) = gcd(b, r)$.*

Bevis: Om $c|a$ og $c|b$, vil også $c|r$ (og $c|b$). Derfor vil $gcd(a, b)|gcd(b, r)$. Og om $c|b$ og $c|r$ vil også $c|a$ (og $c|b$), så $gcd(b, r)|gcd(a, b)$. Siden begge er positive tall, må de

være like.

Hvis Euklids algoritme stopper etter ett skritt, altså hvis $r_1|b$ (sjekk at det er dette det betyr!), så er $gcd(a,b) = gcd(r_1,b) = r_1$, og starten på induksjonen er vist.

Anta så at Euklids algoritme beregner $gcd(x,y)$ for alle tall x og y slik at algoritmen stopper etter $n = k$ skritt. Vi vil vise at det samme gjelder dersom algoritmen først stopper etter $n = k+1$ skritt; anta a og b krever så mange skritt. Se på Euklids algoritme brukt på tallene b og r_1. Denne består av de siste k skrittene av algoritmen brukt på a og b. Ved induksjonsantagelsen er derfor $r_{k+1} = gcd(b,r_1)$ (dette er den k-te resten for b og r_1, men rest nummer $k+1$ for a og b). Men ved lemmaet er $gcd(a,b) = gcd(b,r_1)$, og vi konkluderer til slutt med at $gcd(a,b) = r_n$, som ønsket.

Det beviset jeg har presentert her, er essensielt det samme som det som står i boken, side 27. Les begge!

La oss nå bevise det som i boken heter Theorem 2.3, som ikke ble bevist på forrige forelesning.

Teorem 0.3. *La a og b være heltall, ikke begge null (slik at $gcd(a,b)$ er definert). Da finnes hele tall x og y slik at*

$$gcd(a,b) = ax + by$$

Bevis: Skriv opp Euklids algoritme for a og b, slik at $r_n = gcd(a,b)$. Da er $r_n = r_{n-2} - q_n r_{n-1}$ fra den siste ligningen. Vi setter inn for r_{n-1} fra den nest siste ligningen, og får

$$r_n = r_{n-2} - q_n r_{n-1} = r_{n-2} - q_n(r_{n-3} - q_{n-1}r_{n-2}) = r_{n-2}(1 + q_n q_{n-1}) - q_n r_{n-3}$$

For hver ligning oppover, får vi eliminert den r_k med høyest k fra høyre side av ligningen, og til slutt står vi igjen med er uttrykk på formen a ganger noe pluss b ganger noe på høyre side. Dette er utsagnet i teoremet!

Prøv selv, hvis du vil teste deg i induksjon, å formalisere dette resonnementet som en matematisk induksjon!

Som eksempel, la oss finne tall x og y slik at $3 = 1326x + 231y$, ved å gå baklengs gjennom algoritmen for disse to tallene:

$$\begin{aligned} 3 &= 9 - 1.6 = 9 - (51 - 5.9) = 6.9 - 51 \\ &= 6(60 - 1.51) - 51 = 6.60 - 7.51 \\ &= 6.60 - 7(171 - 2.60) = 20.60 - 7.171 \\ &= 20(231 - 1.171) - 7.171 = 20.231 - 27.171 \\ &= 20.231 - 27(1326 - 5.231) = 155.231 - 27.1326 \end{aligned}$$

Altså finner vi en løsning ved $x = -27$ og $y = 155$.

La oss merke oss en interessant konsekvens, som står godt forklart i boken, side 29:

$$gcd(ka, kb) = k\, gcd(a, b) \text{ for } k > 0$$

Beviset består i å observere at Euklids algoritme for ka, kb fås fra Euklids algoritme for a, b ved å gange hver ligning med k.

En annen interessant konsekvens dreier seg om forholdet mellom største felles divisor og minste felles multiplum. Se først på definisjonen:

Definisjon 0.4. La a og b være hele tall, begge ulik null. Da er det minste felles multiplum $m = lcm(a, b)$ definert som tallet som oppfyller

i) $a|m$ og $b|m$.
ii) Om $a|c$ og $b|c$ så er $m \leq c$.

Her er lcm forkortelse for det engelske least common multiple. Det er nok å se på dette for positive tall, og vi antar dette stilltiende fra nå av. Siden produktet av ab er et tall som både a og b deler, er mengden av felles multipler ikke tom. Derfor gir velordningsprinsippet at lcm er veldefinert.

Så resultatet:

$$lcm(a, b)gcd(a, b) = ab$$

Produktet av to positive tall er det samme som produktet av deres største felles divisor og minste felles multiplum. Vi skal se på beviset i boken (side 30-31), senere skal vi gi et enklere bevis basert på faktorisering av heltall.

På grunn av dette resultatet kan vi også beregne lcm fra Euklids algoritme; finn gcd, og ta produktet delt på gcd. Siden det er så lett å gå fram og tilbake mellom lcm og gcd, er det vanlig å fokusere bare på den ene. Men husk at vi bruker lcm når vi setter brøk på felles brøkstrek!

Jon Eivind Vatne