Matematisk Modellering
Projekt B:
ID-koder

Anders "Bongo" Bjerg Pedersen
Lars Roholm
Martin Hvolby
Jesper Frank Christensen

10. januar 2006

Indhold

1 Indledning

"... tyk, mellem-tyk, tynd, tyk, tyk, tynd, mellemtyk, ..."

Nogenlunde sådan lyder det i en velkendt dansk TV-reklame. Ordene kommer fra en kassemedarbejder, der registrerer kundens varer. Kunderne er forbløffede, og seerne får et smil på læben.
I dag støder vi på identifikationskoder alle vegne. Madvarer, tøj og andre detailvarer, ja vi er faktisk selv et "produkt" med en CPR-kode. Stregkoder er oftest ledsaget af en række cifre, en såkaldt ID-kode. Derfor virker ovenstående citat komisk, da alle idag (i den vestlige verden i hvet fald) ved, at man i stedet for scanning af stregkoden kan indtaste cifrene.
Stregkoder og ID-koder blev opfundet kort efter 2. verdenskrig. Formålet var at mindske det administrative arbejde i butikkerne. Ved at lade alle varer have deres egen kode kunne man lettere holde styr på mængden og salget af varer. Først i 1970'erne blev systemet for alvor taget i brug. Idag findes der mange forskellige former for ID-koder. Varer og produkter har ét system, bøger et andet, pas og personlige identifikationskort et tredje. De forskellige systemer skyldes naturligvis geografiske og branchemæssige præferencer, men lige så meget ønsket om at øge kodesystemets sikkerhed. Med sikkerhed menes der graden af muligt "snyd". I registreringen af koderne er det elementært, at kun de "korrekte"koder godtages. Denne opgave vil fokusere på ID-koders fejlkilder, og hvordan forskellige ID-systemer er konstrueret for at undgå disse.

1.1 Fejlkilder

Ved registrering af identifikationsnumre er det således væsentligt, at de "ugyldige" ID-koder opdages. Årsagerne til en kodes ugyldighed kan være mange og både være forsætlige eller uforsætlige. De forsætlige fejl er dem, hvor en given person bevidst har forsøgt at ændre eller konstruere en stregkode. Dette kan eksempelvis være ændring af personlige oplysninger (CPR-numre eller pas-numre).
De uforsætlige fejl er dem, der opstår, når en korrekt og gyldig kode registreres eller indlæses som ugyldig kode. Fejlkilderne hertil kan findes flere steder steder. Den første kan være den maskinelle registrering. Disse fejl opstår gerne pga. beskadigelse af stregkoden. Fejlen kan også skyldes den manuelle viderelevering eller registrering af koden. I den manuelle indstastning af et id-nummer er den hyppigst forekommende fejl, at et nummer ufrivilligt erstattes af et andet. Denne fejl er årsagen bag ca. 80% af de registrerede tilfælde af ukorrekte id-koder. Andre lignende fejl er, at to cifre ombyttes. Sjældnere resulterer misforståeler i, at kodens cifre indtastes i den omvendte rækkefølge/læseretning.
Fejlen kan ligeledes opstå i den mundtlige kommunikation mellem to personer. Disse fejl kaldes "fonetiske". Typiske fejl i den forbindelse er, at modtageren hører et andet tal end det formidlede. På dansk vil sådanne fejl typisk være ombytning af tallene 4 og 9. "Fireogtyve" kan let høres som "niogtyve".

På engelsk skyldes de fonetiske fejl typisk ligheden mellem f.eks. "sixteen" og "sixty".

1.2 Checkciffer

De oventående eksempler på fejlkilder giver anledning til en væsentlig pointe. Hvordan opdages disse fejl? Til dette har man opfundet en metode, der kontrollerer, om et identifikationsnummer er blevet registreret korrekt. Denne metode indeholder det såkaldte *checkciffer* (fra det engelske: *check digit*). Dette ciffer er oftest placeret til sidst i id-koden og angiver kodens korrekthed. Dette gøres ved en simpel algoritme. De øvrige ciffre indsættes i denne algoritme. Den producerer et outputciffer mellem 0 og 9. Dette ciffer anvendes derefter som *checkciffer*, og er således afhængigt af de øvrige cifre samt deres indbyrdes placering. Algoritmen skal da være konstrueret på en måde, som sikrer at ovenstående fejltyper (især ciffer-ombytning) opdages. Kort fortalt skal en ID-kode, dens algortime og det tilhørende *checkciffer* opfylde en række krav, nemlig at de beskrevne fejltyper opdages, da de resulterer i en ukorrekt ID-kode. Vi vil nu kigge lidt nærmere på hvordan disse krav defineres matematisk.

1.3 Generelle krav til ID-koder

Vi har valgt at fokusere på følgende fejltyper:

Fejltype	Korrekt kode	Ukorrekt kode
"Cifferændring"	**1**234567890	**2**234567890
"Cifferombytning"	**12**34567890	**21**34567890

Det er disse fejl vi gerne vil undgå. Derfor skal vores ID-koder efterleve en række krav. Disse kan vi definere matematisk:

Definition 1 *Vi betragter en gyldig ID-kode, X, med cifrene $a_1, a_2, ..., a_n$ og den ændrede kode X' med cifrene $a'_1, ..., a'_n$, hvor $a_i, a'_i \in \{0, 1, ..., 9\}$ for $i \in \{1, ..., n\}$. Hvis koden X' er gyldig, skriver vi $X' = 1$, og hvis den er ugyldig, skriver vi $X = 0$. Vi siger, at en kode tager højde for de pågældende fejltyper, hvis den ændrede (ugyldige) kode X' opdages. Da kan vi definere fejlkilderne og kravene således:*

- *En "cifferændring" i X defineres som:*

$$\exists! i \in \{1, ..., n\} : a'_i \neq a_i$$

  En kode, X tager højde for "cifferændring" hvis:

$$\forall\ a_i, i \in \{1, ..., n\} \nexists\ a'_i \neq a_i : X' = 1$$

- *En "cifferombytning" i X defineres som:*

$$\forall i, j \in \{1, ..., n\}, i \neq j\ \exists!(i, j) : a'_i = a_j \wedge\ a'_j = a_i$$

  En kode, X tager højde for "cifferombytning" hvis:

$$\forall\ a_i, a_j\ ;\ i, j \in \{1, ..., n\}, i \neq j \nexists\ a'_i = a_j \wedge a'_j = a_i : X' = 1$$

2 Stregkoder

Stregkoder er en af de teknologier, vi tager for givet næsten hver dag. Denne teknologi har gjort salg og køb af varer nemt og bekvemt for såvel forbruger som forhandler.
De første typer af stregkoder dukkede op i 1940'erne, efter at der i mange år havde været en række forsøg på at automatisere prischeck og betaling i supermarkeder. Stregkoden lignede dengang en skydeskive. Den bestod af et antal ringe inden i hinanden, som blev aflæst ved hjælp af noget optik. Gennembrudet kom dog først i 70'erne, hvor den amerikanske organisation, National Association of Food Chains, samarbejdede med elektronikfabrikanter om at udvikle den såkaldte Universal Product Code (UPC). Stregkoden blev hurtigt udbredt og benyttes idag inden for områder lige fra post til fødevarer.

Der findes idag over 50 forskellige[1] standarder for stregkoder. For hurtigt at nævne et par stykker:

UPC	–	En rent numerisk stregkode, der benyttes på de fleste masseproducerede fødevareprodukter idag.
EAN	–	(European Article Numbering) Europæisk version af UPC.
2/5 Kode	–	Også en rent numerisk kode.
Kode39	–	Indeholder både cifre og store bogstaver.
Kode128	–	Kan indeholde stort set alle typer skrifttegn.
POSTNET	–	En speciel kode som postvæsenet i USA benytter.

Bland de ovenstående er det UPC- og EAN-koderne, der er mest udbredt. Over 90% af de varer, der idag bliver solgt i supermarkeder, er mærket med stregkoder af denne type. Vi vil se nærmere på mekanikken bag disse.

2.1 Universal Product Code (UPC)

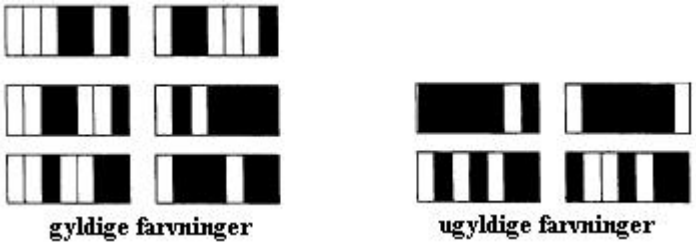
Herunder ses et eksempel på en UPC stregkode:

Stregerne repræsenterer den 12-cifrede kode som står nederst. Hvert af de 12 cifre i koden repræsenteres af en "blok"med 7 felter, dvs. 7 muligheder for enten et hul eller en farvning i stregkoden. Disse huller og farvninger skal arrangeres på en helt bestemt måde:

[1] Sikkert langt over 50...

- Det første felt skal altid være ufarvet, og det sidste skal altid være farvet. Herved fås nu kun 32 mulige kombinationer.
- De 7 felter skal være farvet således, at de som helhed danner 2 ufarvede og 2 farvede barer. En bar kan således højst være 4 felter bred. Med denne restriktion får vi 20 mulige kombinationer.
- Der skal være et ulige antal af mørke farvninger, herved har vi 10 mulige måder at foretage farvningen tilbage.

De 10 mulige måder at foretage farvningen fordeler vi netop som repræsentanter for tallene 0,1,...,9. Herunder ses eksempler på gyldige og ugyldige farvninger.



Vi vender tilbage til selve talkoden. I den tidligere afbildede stregkode havde vi koden 0-26229-77426-3.
Tallet til venstre er et kategoriseringsnummer. Hermed en liste over disse:

0	Alle varemærker fra USA, pånær følgende:
2	Varer, hvis pris er baseret på vægt.
3	Medicin.
4	Ikke-fødevarer som er mærket i forretningen.
5	Gavekort.
6,7	Industrielle varer og byggematerialer, f.eks. skruer, ledning etc.
1,8,9	Reserveret til endnu ukendte formål.

Den venstre blok af 5 cifre er producentens identifikations nummer. Dette er blevet tildelt firmaet af en UPC-kodekomité.
Den højre blok af 5 cifre er en produktkode, som producenten selv bestemmer. Fremstiller producenter mere end 100000 forskellige produkter, får de tildelt endnu et producentidentifikationsnummer.
Det sidste tal i koden, tallet til højre, er et checkciffer. Checkcifferets funktion er at afsløre, om der er fejl i skanningen/indtastningen af stregkoden. Med henblik på checkcifferets funktion kan vi nu opskrive et kriterie for, hvad der er en gyldig kode.

Definition 2 *For en UPC-stregkode* $a_1 - a_2a_3a_4a_5a_6 - a_7a_8a_9a_{10}a_{11} - c$ *skal der gælde:*

$$3a_1 + a_2 + 3a_3 + a_4 + 3a_5 + a_6 + 3a_7 + a_8 + 3a_9 + a_{10} + 3a_{11} + c \equiv 0 \pmod{10}$$

Eksempel 1 *Betragt førviste UPC-stregkode 0-26229-77426-3. Kategoriseringsnummeret er 0, producentens identifikationsnummer er 26229, produktkoden er 77426 og checkcifferet er 3. Vi checker efter, og finder at denne er gyldig:*

$$3\cdot 0+2+3\cdot 6+2+3\cdot 2+9+3\cdot 7+7+3\cdot 4+2+3\cdot 6+3 = 100 \equiv 0 \pmod{10}$$

2.2 European Article Numbering (EAN)

Herunder ses en EAN-stregkode.

Stregerne repræsenterer også her koden, efter samme princip som i UPC-koden.
EAN koden består af 13 cifre. De tre første cifre er en landekode som er blevet tildelt landene af International Article Numbering Association EAN i Bruxelles, Belgien. Danmark har f.eks. fået tildelt numrene 570-579. De næste ni cifre er en produktkode. Det er normalt, men ikke påkrævet, at de fire første cifre repræsenterer producenten, og de sidste fem repræsenterer det pågældende produkt. Det sidste ciffer er igen et checkciffer. Dettes funktion er igen at checke, om koden er blevet scannet korrekt. Som ved UPC-koden kan vi nu opskrive et lignende kriterie for EAN-koden:

Definition 3 *For en EAN-stregkode* $a_1 - a_2a_3a_4a_5a_6a_7 - a_8a_9a_{10}a_{11}a_{12}c$ *skal der gælde:*

$$a_1+3a_2+a_3+3a_4+a_5+3a_6+a_7+3a_8+a_9+3a_{10}+a_{11}+3a_{12}+c \equiv 0 \pmod{10}$$

Eksempel 2 *Vi betragter den viste EAN-stregkode 5-714561-124154. Landekoden er 571, og produktkoden er 456112415. Checkcifferet er 4. Vi checker efter og finder, at denne er gyldig:*

$$5+3\cdot 7+1+3\cdot 4+5+3\cdot 6+1+3\cdot 1+2+3\cdot 4+1+3\cdot 5+4 = 100 \equiv 0 \pmod{10}$$

2.3 Fejlsikkerhed

Vi kan spørge os selv, om checkciffersystemet afslører alle typer af fejl. Afslører det cifferfejl? Opdages en ugyldig kode, hvis præcist ét ciffer ændres i en gyldig kode? Afslører checket ombytning af nabocifre? Afslører checket cifferombytninger generelt? Vi vil betragte de to typer stregkoder, vi har fokuseret på tidligere. Da UPC- og EAN-koden er meget ens i deres checkciffersystem, vil vi kun gennemgå fejlsikkerheden af UPC-koden. Dette kan ultrolig nemt overføres til EAN-koden.
Vi vil starte med at finde ud af, om UPC-kodens ciffercheck afslører desiderede cifferfejl. Dvs. at ét ciffer i koden er erstattet med et vilkårligt andet ciffer. Vi skal altså vise, at for en gyldig UPC-kode $a_1a_2a_3a_4a_5a_6a_7 - a_8a_9a_{10}a_{11}a_{12}$ gælder det, at en udskiftning af et a_i med et a_i', hvor $a_i \neq a_i'$, medfører:

$$3a_1+a_2+3a_3+a_4+3a_5+a_6+3a_7+a_8+3a_9+a_{10}+3a_{11}+a_{12} \not\equiv 0 \pmod{10}$$

Dette er oplagt, da 3 er primisk med 10. Tallet 3 er netop valgt til brug ved konstruktion af checkcifferet, da det har denne egenskab.
Vi vil nu vise, at ombytning af nabocifre giver en ugyldig kode. Antag at vi ombytter cifrene a_i og a_{i+1}, hvor $a_i \neq a_{i+1}$. For at den nye kode (efter ombytningen) skal være gyldig, skal der altså gælde, at:

$$3a_i + a_{i+1} \equiv a_i + 3a_{i+1}$$

Men dette fører oplagt til en modstrid, da $3a_i + a_{i+1} \equiv a_i + 3a_{i+1} \Rightarrow 2a_i \equiv 2a_{i+1} \Rightarrow a_i \equiv a_{i+1}$, hvilket jo strider imod vores antagelse om, at $a_i \neq a_{i+1}$. Det er oplagt, at cifferchecket ikke afslører alle vilkårlige cifferombytninger. Ombyttes et a_i med et a_j, hvor i og j begge er enten lige eller ulige, vil checkcifferet ikke afsløre ombytningen.

3 ISBN-nummersystemet

Vi skal i følgende afsnit se nærmere på ISBN-systemet, der er et identifikationssystem, der anvendes til at nummerere bøger over hele verden. Systemet gør det nemt at søge, bestille og sammenligne bøger på tværs af landegrænser og sproggrænser, og flere karakteristika om den enkelte bog kan nemt læses ud fra dens ISBN-nummer. Nummeret ledsages som oftest af en stregkode, der er genereret ud fra selve nummeret, hvilket letter f.eks. udlån eller registrering.

3.1 Historien bag ISBN

Da Englands største bogkæde W.H. Smith i 1965 bekendtgjorde, at man ville overgå til computeriserede boghandler med bedre mulighed for opslag og lignende, blev der nedsat et udvalg, der skulle komme med en løsning på problemet med en manglende standard for bognummerering. Udvalget kom i 1966 frem til SBN-systemet (Standard Book Number), der blev implementeret i 1967, og i 1970 blev systemet i et udvalg under International

Organization for Standardization (ISO) bestående af repræsentanter fra 8 lande og UNESCO implementeret som en international standard under navnet ISBN (International Standard Book Number). ISBN er i sin grundform stadig uændret og anvendes i dag i over 150 lande verden over.

3.2 ISBN generelt

ISBN-numre er enten 10 eller 13 cifre lange. Vi skal fokusere på dem af længde 10 og se nærmere på disses egenskaber. Herunder ses et eksempel på et ISBN-nummer med dertilhørende stregkode:

Ovenstående er endda tilføjet en ekstra 5-cifret stregkode, der indeholder en valuta og vejledende udsalgspris.

ISBN-numrene består af 10 cifre, der kan være inddelt på forskellige måder i "blokke"med (som regel) bindestreger. Første "blok"af cifre angiver, hvilket sprog der tales, der hvor bogen er udgivet. F.eks. har engelsksprogede bøger et 0 som første ciffer, franskprogede har 2 osv. Enkelte lande, som eksempelvis Danmark, der har sit eget sprog, har således også sit eget nummer (i Danmarks tilfælde 87). Ovenstående billede er altså fra en engelsksproget bog.
Anden "blok"angiver, hvilket forlag, der har udgivet bogen. Denne blok kan være alt fra 1 til 8 cifre langt, men er som oftest mellem 2 og 5 cifre.
Tredje "blok"angiver forlagets nummer for bogen. Her kan det understreges, at et forlag, der udgiver mange bøger, er interesseret i et kort forlagsnummer, så der er plads til flere bogtitler. Derfor er det også dyrere, jo kortere forlagsnummer man ønsker at købe.
Sidste "blok"har altid længden 1 og er den matematisk set væsentligste blok, nemlig *checkcifferet*. Dette ciffer beregnes ud fra de foregående 9 og afgør, om ISBN-nummeret er gyldigt. Vi skal om lidt se nærmere på dette ciffer, der kan antage værdierne 0,1,2,...9 og X.
Eksempelvis har kursets ene bog "Mathematical Modelling"ISBN-nummeret 0-8218-3650-1. Her angiver 0, at bogen er på engelsk, 8218 angiver, at det er American Mathematical Society, der har udgivet den, 3650 er AMS' interne nummer for bogen, og 1 er checkcifferet.

3.3 Matematikken bag ISBN

Det sidste ciffer bruges til at tjekke validiteten af det pågældende ISBN-nummer. Det virker på følgende måde: nummerets ti cifre ganges med tallene 1 til 10 i omvendt rækkefølge, og de heraf fremkomne tal adderes. Er summen nu lig med 0 modulo 11, er ISBN-nummeret korrekt. Checkcifferet a_{10} til ISBN-nummeret $a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}$ vælges altså, så følgende ligning er opfyldt:

$$10a_1 + 9a_2 + 8a_3 + 7a_4 + 6a_5 + 5a_6 + 4a_7 + 3a_8 + 2a_9 + 1a_{10} = 0 \pmod{11}.$$

Lad os igen tage AMS-eksemplet ovenfor:

$$\begin{gathered} 10 \cdot 0 + 9 \cdot 8 + 8 \cdot 2 + 7 \cdot 1 + 6 \cdot 8 + 5 \cdot 3 + 4 \cdot 6 + \\ 3 \cdot 5 + 2 \cdot 0 + 1 \cdot 1 \pmod{11} = 198 \pmod{11} = 0. \end{gathered}$$

Altså er der (sjovt nok) tal om et gyldigt ISBN-nummer. Grunden, til at checkcifferet også kan være et X, er selvfølgelig, at det kan være nødvendigt at lægge 10 til summen for at få den til at stemme modulo 11.

3.4 Fejlsikkerhed

ISBN-systemet er fejlsikret mod de to mest forekommende fejl, nemlig fejl i et enkelt ciffer og ombytning af to nabocifre. Dette er dog til dels sket på bekostning af, at systemet er nødt til at gøre brug af det ekstra "tal"X, og at ISBN-numrene skal være præcis 10 cifre lange.

Men hvorfor egentlig modulo 11? Se f.eks. på følgende eksempel, hvor vi i stedet har anvendt modulo 5:

$$\begin{gathered} 10 \cdot 0 + 9 \cdot 1 + 8 \cdot 3 + 7 \cdot 1 + 6 \cdot 3 + 5 \cdot \mathbf{9} + 4 \cdot 1 \\ +3 \cdot 3 + 2 \cdot 9 + 1 \cdot 1 \pmod{5} = 90 \pmod{5} = 0. \end{gathered}$$

Vi bemærker nu to ting:

1. Checkcifferet kunne være 1 eller 6 og give samme resultat.
2. Tallet 9 (med fed) kunne være hvad som helst.

Det er nu klart, at vi skal regne modulo et tal større end 10. Lad os prøve et andet ISBN-nummer modulo 12:

$$\begin{gathered} 10 \cdot 0 + 9 \cdot 1 + 8 \cdot 3 + 7 \cdot 1 + 6 \cdot 3 + 5 \cdot 9 + 4 \cdot 1 \\ +3 \cdot \mathbf{3} + 2 \cdot \mathbf{9} + 1 \cdot X \pmod{12} = 144 \pmod{12} = 0. \end{gathered}$$

Igen ser vi to problemer:

1. Det sidste 9-tal (med fed) kunne lige så godt have været 9-6=3, da havde summen givet $132 = 0 \pmod{12}$.
2. Det sidste 3-tal (med fed) kunne lige så godt have været 3+4=7, da havde summen givet $156 = 0 \pmod{12}$.

Grunden til ovenstående problemer er, at 12 har faktorer under 10, der på respektive pladser i nummeret kan give "ekstra"12'ere til summen. Dette problem kan løses ved at regne modulo et primtal, eksempelvis 11.

Problemet med at have byttet om på to nabocifre er også løst, når vi regner modulo 11. Lad a og b være de to nabocifre med multiplikatorer n og $n+1$ (altså i omvendt rækkefølge, dvs. n og $n+1$ er de tal, a og b skal ganges med i summen). De to cifre bidrager til summen med $c_1 = an + b(n+1)$. Når de to cifre ombyttes, bidrager de med $c_2 = bn + a(n+1)$, dvs. når cifrene ombyttes, ændres summen med $c_1 - c_0 = a - b$, ligegyldigt hvor i ISBN-nummeret cifrene optræder. Da differencen $a - b$ aldrig kan blive 11 eller derover, vil fejlen altid kunne opdages. I det tilfælde, hvor der ombyttes to *tilfældige* cifre med afstanden $m < 11$, vil summen ændre sig med faktoren $m(a-b)$, der jo aldrig kan blive et multipla af 11. Altså er ISBN også sikret overfor tilfældige ombytninger af cifre.

3.5 Anvendelser

I daglig anvendelse på f.eks. biblioteker benytter stregkodelæsere af ovenstående metoder til at afgøre, om en stregkode er udtryk for et gyldigt ISBN-nummer eller ej. Det kræver ikke den store programmeringsevne at konstruere et program, der tjekker ISBN-numre:

```
>restart;
>isbn:=proc(B)
 local A,i,S;

 A:=B;
 if nops(A)>10 or nops(A)<10 then
  print("Invalid ISBN, bad lenght!");
 else
  if A[10]='X' then A[10]:=10; end if;
  S:=sum((11-i)*A[i],i=1..10);
  if S mod 11=0 then
   print("Valid ISBN");
  else
   print("Invalid ISBN, bad checksum!");
  end if;
 end if;
 end proc:
>B:=[0,8,2,1,8,3,6,5,0,1];
                [0, 8, 2, 1, 8, 3, 6, 5, 0, 1]
>isbn(B);
                          "Valid ISBN"
>B:=[0,2,0,1,5,2,0,3,2,X];
                [0, 2, 0, 1, 5, 2, 0, 3, 2, X]
>isbn(B);
                          "Valid ISBN"
 >B:=[3,3,5,7,0,2,0,0,1,4];
                [3, 3, 5, 7, 0, 2, 0, 0, 1, 4]
>isbn(B);
                  "Invalid ISBN, bad checksum!"
>B:=[3,3,5,7,1,0,2,0,0,1,4];
              [3, 3, 5, 7, 1, 0, 2, 0, 0, 1, 4]
>isbn(B);
                  "Invalid ISBN, bad lenght!"
```

Ovenstående Maple-program tjekker en stregkode på listeform som input og spytter en godkendelse eller en forkastelse med fejltype ud.

4 Verhoeff Check-ciffer Systemet

I 1968 udviklede J. Verhoeff et checkciffersystem, der ikke kun opfanger de tidligere nævnte fejlkilder, men desuden kan benyttes på id-koder af vilkårlig længe.
Lad $a_1a_2...a_{n-1}a_n$ være id-nummeret med check-cifret a_n. Da skal a_n vælges sådan, at følgende ligning tilfredsstilles:

$$\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma(a_{n-1}) * a_n = 0, \tag{1}$$

hvor $\sigma = (0)(1,4)(2,3)(5,6,7,8,9)$ og $*$ er gruppeoperationen fra Diedergruppen D_5 som virker på tallene 0,1,...,9. Dette kan let illustreres i understående tabel, der tilskrives Cayley:

*	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Vi vil nu teste Verhoeff checkciffersystemet i forhold til de krav, vi har sat op for et pålideligt system.

4.1 Test for cifferfejl

Vi skal nu teste for, at man ved forkert indtastning af et forkert ciffer ikke kan opnå et gyldigt id-nummer. Eller mere præcist:
Vi antager, at id-nummeret $a_1a_2...a_{n-1}a_n$ er gyldigt, altså opfylder det (1). Vi antager nu, at vi ændrer præcist ét tal $a_i, i = (1, ..., n)$, så $a_i' \neq a_i$, og viser, at så opfyldes (1) ikke.

Dette gøres over to trin. Først vises det, at

$$\begin{aligned} & a_i \neq a_i' \\ \Rightarrow\ & \sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i) * ... * \sigma(a_{n-1}) * a_n \\ \neq\ & \sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i') * ... * \sigma(a_{n-1}) * a_n \end{aligned}$$

Dette vises let ved kontraposition. Vi antager

$$\begin{aligned}&\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i) * ... * \sigma(a_{n-1}) * a_n \\ = &\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i') * ... * \sigma(a_{n-1}) * a_n \\ &\Rightarrow \sigma^{n-i}(a_i) = \sigma^{n-i}(a_i')\end{aligned}$$

Vi betragter σ, og det ses let, at $\sigma^p, p \in \mathbb{Z}$ er en bijektiv afbildning af talmængden $0, 1, ..., 9$ på sig selv. Derved er

$$\sigma^{n-i}(a_i) = \sigma^{n-i}(a_i') \Leftrightarrow a_i = a_i'$$

og det ønskede er vist.

Vi skal nu vise, at hvis

$$\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i) * ... * \sigma(a_{n-1}) * a_n = 0$$

og

$$\begin{aligned}&\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i) * ... * \sigma(a_{n-1}) * a_n \\ \neq &\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i') * ... * \sigma(a_{n-1}) * a_n,\end{aligned}$$

så følger det, at

$$\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma^{n-i}(a_i') * ... * \sigma(a_{n-1}) * a_n \neq 0.$$

Vi lader for nemhedens skyld a_1 være det ændrede ciffer a_1' (beviset kan let udledes til hvilket som helst ciffer).

Vi antager

$$\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) * ... * \sigma(a_{n-1}) * a_n = 0,$$

og vi antager $a_1 \neq a_1'$.
Vi betragter ligningen

$$\sigma^{n-1}(a_1') * \sigma^{n-2}(a_2) * ... * \sigma(a_{n-1}) * a_n = 0 \tag{2}$$

og vil vise, at dette er i modstrid med vores antagelser.

Vi betragter et element x i gruppen D_5 og sætter

$$x = \sigma^{n-2}(a_2) * ... * \sigma(a_{n-1}) * a_n$$

Vi ved fra gruppeteorien, at ethvert element x i en gruppe G har et og kun et invers element x^{-1} i gruppen.

Fra vores antagelser ser vi, at

$$\sigma^{n-1}(a_1) * x = 0$$

Da må $\sigma^{n-1}(a_1) = x^{-1}$. Men hvis (2) skal gælde, må også $\sigma^{n-1}(a_1') = x^{-1}$, og altså $\sigma^{n-1}(a_1) = \sigma^{n-1}(a_1')$. Men da viste vi før, at

$$\sigma^{n-1}(a_1) = \sigma^{n-1}(a_1') \Leftrightarrow a_1 = a_1'$$

hvilket er imod vores antagelser, og dermed en modstrid. Det ønskede er vist.

4.2 Test af cifferombytning

Hvis vi helt generelt betragter følgende ligning i en gruppe G:

$$x * y * z * ... * w = 0$$

og herefter ombytter to naboelementer x og y. Kan vi så få

$$y * x * z * ... * w = 0?$$

Det ses let, at dette kun kan forekomme, hvis gruppen er kommutativ, da

$$y * x = x * y = (z * ... * w)^{-1}$$

Vi betragter en given gyldig id-kode $a_1a_2...a_n$, og skal nu i 2 trin vise, at en ombytning af nabocifre ikke kan give en ny gyldig kode. For nemhedens skyld lader vi det igen være de to første cifre der ombyttes (dette kan let udledes til to vilkårlige nabotal).

Først må vi sikre os, at for $a_1 \neq a_2$ må permutationen σ ikke tillade, at

$$\sigma^{n-1}(a_1) = \sigma^{n-1}(a_2)$$

og tilsvarende

$$\sigma^{n-2}(a_1) = \sigma^{n-2}(a_2).$$

Dette er oplagt, da $\sigma^p, p \in \mathbb{N}$ er en bijektiv afbildning af mængden $\{0, 1, ..., 9\}$ på sig selv. Altså kan to forskellige elementer ikke føres over i det samme element.

Hernæst må vi sikre os, at kommutativitet ikke kan forekomme, dvs. som nævnt ovenfor

$$\sigma^{n-1}(a_1) * \sigma^{n-2}(a_2) \neq \sigma^{n-1}(a_2) * \sigma^{n-2}(a_1).$$

Vi bemærker fra Cayley-tabellen, at kommutativitet kun forekommer, hvis begge elementer ligger i mængden $\{0, 1, 2, 3, 4\}$ svarende til $\{id, D, D^2, D^3, D^4\}$

i D_5. σ er en bijektiv afbildning af delmængden $\{0,1,2,3,4\}$ på sig selv, så vi kan herfra nøjes med at undersøge koder, hvor de ombyttede elementer begge ligger i denne delmængde.

Hvis vi betragter

$$\sigma : \{0,1,2,3,4\} \to \{0,1,2,3,4\}, \sigma = (0)(1,4)(2,3)$$

og ser på potenser af σ, ser vi hurtigt, at lige potenser er lig identiteten, og ulige potensen er lig ovenstående fremstilling.

Herfra er det let at se, at kommutativitet ikke kan forekomme. For $a_1, a_2 \in \{0,1,2,3,4\}$ gælder jo, at netop et af cifrene vil fikseres og det andet transponeres. Hvis de to cifre byttes om, fås et nyt element i gruppen (af omfangsmæssige årsager bevises dette ikke her).

Verhoeff checkciffersystemet opfanger altså fejl sket ved ombytning af nabocifre. Gælder det så også generelle ombytninger af to vilkårlige cifre? Nej... (selv om litteraturen siger noget andet).

Betragt følgende id-nummer: 43276.
Dette er gyldigt, da

$$\begin{aligned}\sigma^4(4) * \sigma^3(3) * \sigma^2(2) * \sigma(7) * 6 \\ = 4 * 2 * 2 * 8 * 6 \\ = 1 * 5 * 6 \\ = 6 * 6* = 0\end{aligned}$$

Nu ombyttes a_1 og a_3 og nummeret 23476 testes:

$$\begin{aligned}\sigma^4(2) * \sigma^3(3) * \sigma^2(4) * \sigma(7) * 6 \\ = 2 * 2 * 4 * 8 * 6 \\ = 4 * 7 * 6 \\ = 6 * 6 = 0\end{aligned}$$

som altså også er gyldigt. Verhoeffs system fejler altså denne test.

5 Konklusion

Vi har i denne opgave set på forskellige identifikationssystemer, der er brugbare til forskellige formål. Alle kodesystemerne er i stand til at finde de mest basale fejl, såsom cifferfejl og cifferombytning. Hvor EAN, UPC og ISBN er forholdsvist simple rent beregningsmæssigt, er Verhoeff-systemet væsentligt mere kompliceret, men har derimod den force, at det kan anvendes mere generelt (man er f.eks. ikke låst fast til en bestemt kodelængde).

6 Kilder

- Marian Visich:*Bar Codes and Their Applications*, http://www.math.dartmouth.edu/~mqed/NLA/BarCodes/BarCodes_2d5.pdf
- http://www.246.dk/barcodes.html - Information om diverse typer stregkoder.
- http://www.answers.com/topic/universal-product-code - Om UPC koden.
- http://www.tinohempel.de/info/mathe/ean/ean.htm - Om EAN koden (Tysk).
- http://www.adams1.com/pub/russadam/upccode - Om UPC koden, bla. en liste over de forskellige landekoder.
- http://www.mecsw.com/info/intro.htm - Generelt om stregkoder (overfladisk).
- http://www.academic.marist.edu/mwa/index.htm - Information om diverse typer af ID-koder, bla. UPC og ISBN koder.
- *Secrets of the ISBN - An Error Detection Method*, University of New Brunswick, Department of Electrical and Computer Engineering, 1998. http://www.ee.unb.ca/tervo/ee4243/isbn4243.htm
- Joe Kirtlan: *Mathematics and Writing in Action*, Marist College, http://www.academic.marist.edu/mwa/isbn.htm
- Jeremy Bradbury: *Introduction to Check Digits*, http://www.cs.queensu.ca./home/bradbury/checkdigit/index.html