

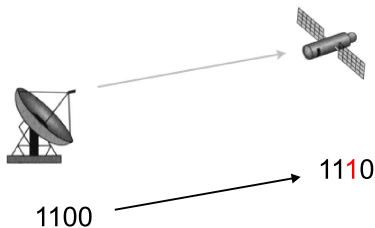
Fejlrettende koder - abstrakt matematik i anvendelse

Kristian Brander og Peter Beelen

25. november 2008

Hvad er fejlrettende koder?

Problemstillingen: Kommunikation over et medie der potentielt forvansker de sendte data.



Fx oplever man ved satellitkommunikation at atmosfæriske forstyrrelser kan forårsage ændringer af de sendte data.

Hvor anvendes fejlrettende koder?

- Fejlrettende koder anvendes i mange af de apparater vi omgiver os med i dagligdagen, fx computere, satelliter, mobiltelefoner, CD- og DVD-afspillere.



Hvor anvendes fejlrettende koder?

- Fejlrettende koder anvendes i mange af de apparater vi omgiver os med i dagligdagen, fx computere, satelliter, mobiltelefoner, CD- og DVD-afspillere.



- På en CD er der således placeret *redundante* nuller og et-taller, der udelukkende bruges hvis der skulle ske fejl på den "rigtige" information, altså på CD'en lyd eller data.

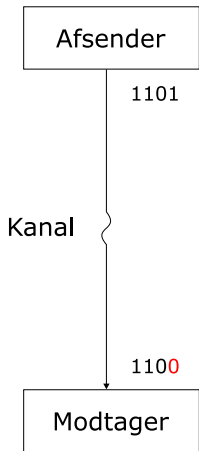
Hvor anvendes fejlrettende koder?

- Fejlrettende koder anvendes i mange af de apparater vi omgiver os med i dagligdagen, fx computere, satelliter, mobiltelefoner, CD- og DVD-afspillere.



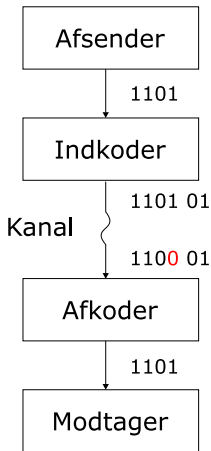
- På en CD er der således placeret *redundante* nuller og et-taller, der udelukkende bruges hvis der skulle ske fejl på den “rigtige” information, altså på CD'en lyd eller data.
- Man kan bore et lille hul i en CD og stadig afspille den!

En matematisk model



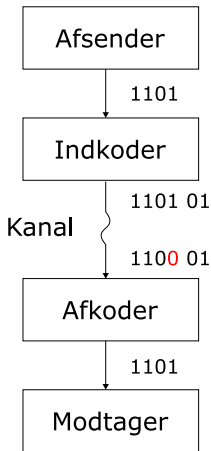
- Man har to parter der ønsker at kommunikere over en upålidelig kanal.

En matematisk model



- Man har to parter der ønsker at kommunikere over en upålidelig kanal.
- Man indskyder en *indkoder* og en *afkoder*.

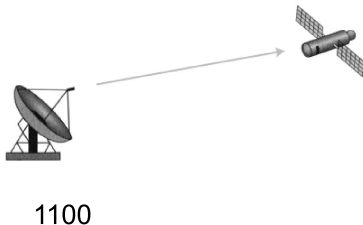
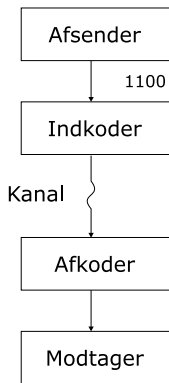
En matematisk model



- Man har to parter der ønsker at kommunikere over en upålidelig kanal.
- Man indskyder en *indkoder* og en *afkoder*.
- Både ind- og afkoder er funktioner i den sædvanlige matematiske forstand.

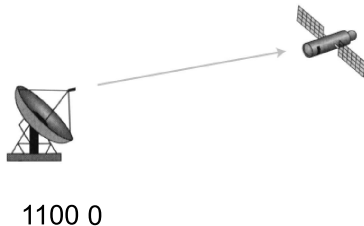
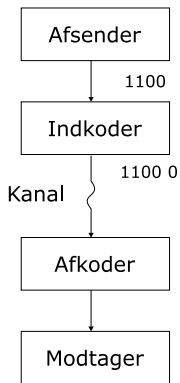
Paritetscheck

I eksemplet med satelliten kan man fx tilføje et ekstra ciffer sådan at antallet af 1-taller i det der sendes altid er lige. Dette kaldes et *paritetscheck* og med et sådant kan afkoderen opdage én fejl.



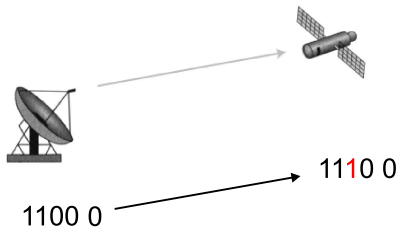
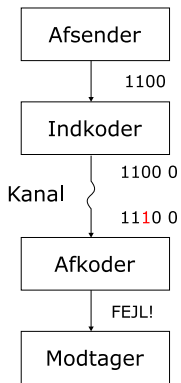
Paritetscheck

I eksemplet med satelliten kan man fx tilføje et ekstra ciffer sådan at antallet af 1-taller i det der sendes altid er lige. Dette kaldes et *paritetscheck* og med et sådant kan afkoderen opdage én fejl.



Paritetscheck

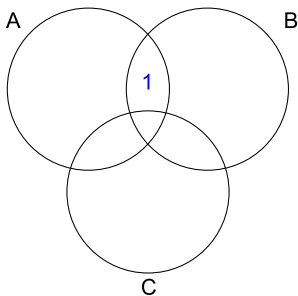
I eksemplet med satelliten kan man fx tilføje et ekstra ciffer sådan at antallet af 1-taller i det der sendes altid er lige. Dette kaldes et *paritetscheck* og med et sådant kan afkoderen opdage én fejl.



Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

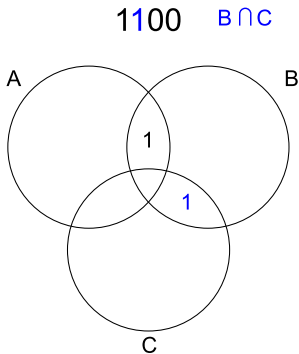
1100 $A \cap B$



Som før ønskes fire bit sendt, og disse placeres i cirkeldiagrammet.

Hamming-kode

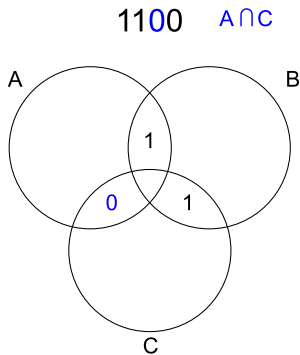
Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Som før ønskes fire bit sendt, og disse placeres i cirkeldiagrammet.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

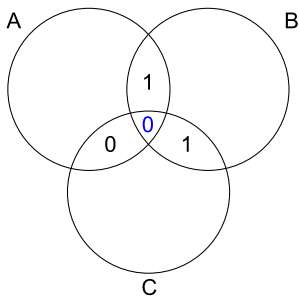


Som før ønskes fire bit sendt, og disse placeres i cirkeldiagrammet.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

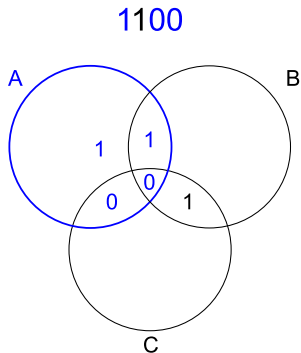
1100 $A \cap B \cap C$



Som før ønskes fire bit sendt, og disse placeres i cirkeldiagrammet.

Hamming-kode

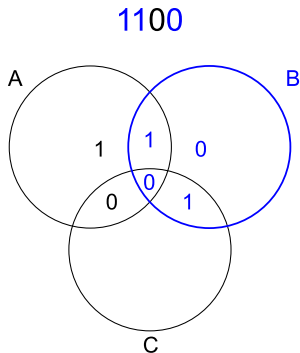
Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Der tilføjes *paritetscheck* – antallet af 1-taller i hver af cirklerne A, B og C skal være lige.

Hamming-kode

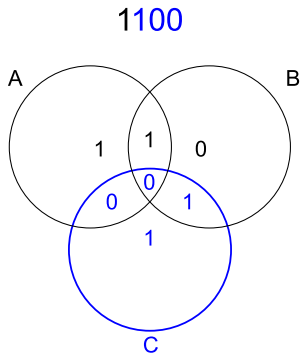
Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Der tilføjes *paritetscheck* – antallet af 1-taller i hver af cirklerne A, B og C skal være lige.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

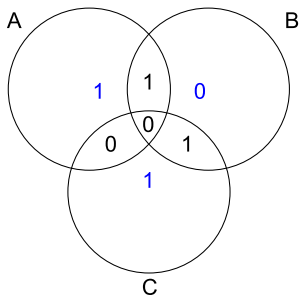


Der tilføjes *paritetscheck* – antallet af 1-taller i hver af cirklerne A, B og C skal være lige.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

1100 101

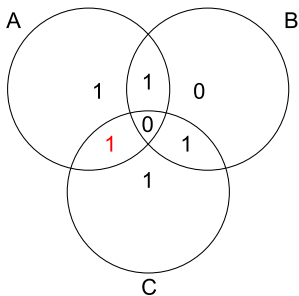


De 3 ekstra bits tilføjes til de oprindelige 4, og de resulterende 7 bits er nu det *kodeord* der sendes afsted.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.

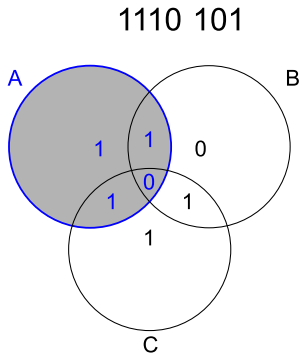
1110 101



Nu tilføjes én fejl på kodeordet.

Hamming-kode

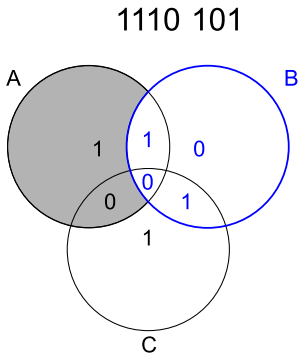
Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Hver cirkel gennemgås og det undersøges om paritetschecket er opfyldt.

Hamming-kode

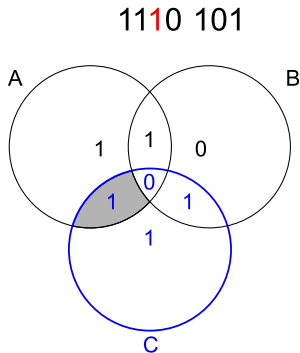
Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Hver cirkel gennemgås og det undersøges om paritetschecket er opfyldt.

Hamming-kode

Vi gennemgår nu en metode der sætter afkoderen i stand til ikke blot at *opdage*, men også *rette* én fejl.



Hver cirkel gennemgås og det undersøges om paritetschecket er opfyldt.

Længde, dimension og alfabet

- Ved en **kode** forstås en samling af ord (eller vektorer) med elementer fra en mængde \mathbb{F} , der kaldes kodens **alfabet**. De enkelte ord i en kode kaldes **kodeord**.

Længde, dimension og alfabet

- Ved en **kode** forstås en samling af ord (eller vektorer) med elementer fra en mængde \mathbb{F} , der kaldes kodens **alfabet**. De enkelte ord i en kode kaldes **kodeord**.
- I eksemplet så vi at hvert kodeord består af 7 elementer i $\mathbb{F} = \{0, 1\}$. Antallet af pladser i hvert kodeord kaldes kodens **længde**.

Længde, dimension og alfabet

- Ved en **kode** forstås en samling af ord (eller vektorer) med elementer fra en mængde \mathbb{F} , der kaldes kodens **alfabet**. De enkelte ord i en kode kaldes **kodeord**.
- I eksemplet så vi at hvert kodeord består af 7 elementer i $\mathbb{F} = \{0, 1\}$. Antallet af pladser i hvert kodeord kaldes kodens **længde**.
- I eksemplet så vi at 4 af de 7 pladser i et kodeord kunne vælges frit, og at de resterende 3 pladser entydigt bestemte ved disse. Antallet af pladser i hvert kodeord der kan vælges frit, kaldes kodens **dimension**.

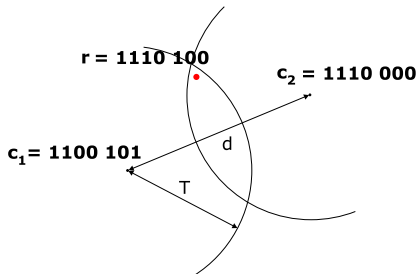
Længde, dimension og alfabet

- Ved en **kode** forstås en samling af ord (eller vektorer) med elementer fra en mængde \mathbb{F} , der kaldes kodens **alfabet**. De enkelte ord i en kode kaldes **kodeord**.
- I eksemplet så vi at hvert kodeord består af 7 elementer i $\mathbb{F} = \{0, 1\}$. Antallet af pladser i hvert kodeord kaldes kodens **længde**.
- I eksemplet så vi at 4 af de 7 pladser i et kodeord kunne vælges frit, og at de resterende 3 pladser entydigt bestemte ved disse. Antallet af pladser i hvert kodeord der kan vælges frit, kaldes kodens **dimension**.
- For at have en måde at afgøre hvor "forskellige" to kodeord c_1 og c_2 er, indfører man **afstanden** imellem dem som

$dist(c_1, c_2) = \text{Antal pladser hvorpå } c_1 \text{ og } c_2 \text{ er forskellige.}$

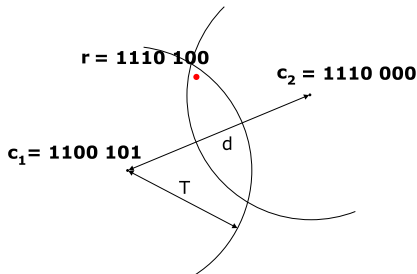
Afstanden mellem kodeord

- Afstanden mellem to kodeord c_1 og c_2 bestemmer på hvor mange pladser der skal ske fejl i c_1 for at "omdanne" det til c_2 .



Afstanden mellem kodeord

- Afstanden mellem to kodeord c_1 og c_2 bestemmer på hvor mange pladser der skal ske fejl i c_1 for at “omdanne” det til c_2 .



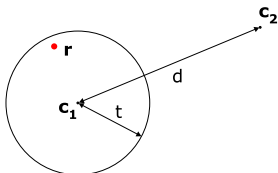
- Hvis der sker flere end $T = \frac{dist(c_1, c_2)}{2}$ fejl, kan modtageren ikke afgøre om det modtagne ord r stammer fra c_1 eller c_2 .
- I figuren er der lavet 2 fejl på kodeordet fra eksemplet, her kaldet c_1 . Man kan vise at c_2 også er et kodeord, og r ligger faktisk tættere på c_2 end på det “rigtige” kodeord c_1 .

Minimumsafstand

- Man indfører **minimumsafstanden** d for en kode som den *mindst mulige afstand mellem to forskellige kodeord*.
- Hvis man antager at der under transmissionen af et kodeord sker højst t fejl, hvor

$$t < \frac{d}{2},$$

er man sikker på at der i afstand højst t fra det modtagne ord ligger *præcis* et kodeord.



- Under denne antagelse er man altså sikker på at r stammer fra c_1 , og altså kan koden rette t fejl.

Minimumsafstand for Hamming-koden

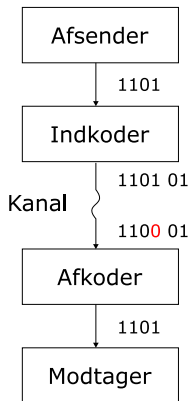
Kodeordene i Hamming-koden fra eksemplet kan vises at være

| | |
|----------|----------|
| 0000 000 | 1010 011 |
| 1000 110 | 1001 001 |
| 0100 011 | 0101 100 |
| 0010 101 | 1110 000 |
| 0001 111 | 1101 010 |
| 1100 101 | 1011 100 |
| 0110 110 | 0111 001 |
| 0011 010 | 1111 111 |

Ved at gå alle par af kodeord igennem og udregne deres afstand, kan man se at kodens minimumsafstand er 3, og dermed kan den rette 1 fejl, i overensstemmelse med eksemplet.

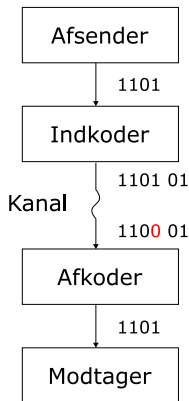
Dagens program ...

Hvordan konstrueres fejlrettende koder, altså ind- og afkodere, der kan rette flere fejl end i eksemplerne?



Dagens program ...

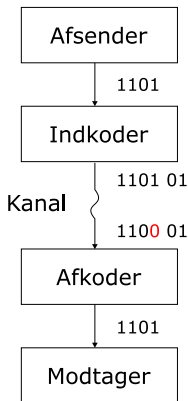
Hvordan konstrueres fejlrettende koder, altså ind- og afkodere, der kan rette flere fejl end i eksemplerne?



- 1 Givet k og t vil vi lave en kode der har dimension k og kan rette t fejl.

Dagens program ...

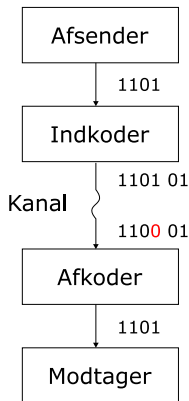
Hvordan konstrueres fejlrettende koder, altså ind- og afkodere, der kan rette flere fejl end i eksemplerne?



- 1 Givet k og t vil vi lave en kode der har dimension k og kan rette t fejl.
- 2 Vil konstruere ind- og afkodere til denne kode.

Dagens program ...

Hvordan konstrueres fejlrrettende koder, altså ind- og afkodere, der kan rette flere fejl end i eksemplerne?



- 1 Givet k og t vil vi lave en kode der har dimension k og kan rette t fejl.
- 2 Vil konstruere ind- og afkodere til denne kode.
- 3 Regne på konkrete eksempler, og se præcis hvad der sker med en meddelelse fra dens vej fra afsender til modtager.

Hvad er et legeme?

- Mængden \mathbb{F} siges at være et legeme såfremt der findes to operationer $+$ og \cdot , samt to elementer $0, 1 \in \mathbb{F}$ således at:
 - 1 For alle $x, y \in \mathbb{F}$ gælder $x + y = y + x$.
 - 2 For alle $x, y, z \in \mathbb{F}$ gælder $(x + y) + z = x + (y + z)$.
 - 3 For alle $x \in \mathbb{F}$ gælder $x + 0 = x$.
 - 4 For alle $x \in \mathbb{F}$ findes et $y \in \mathbb{F}$ således at $x + y = 0$.
 - 5 For alle $x, y \in \mathbb{F}$ gælder $x \cdot y = y \cdot x$.
 - 6 For alle $x, y, z \in \mathbb{F}$ gælder $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
 - 7 For alle $x \in \mathbb{F}$ gælder $x \cdot 1 = x$.
 - 8 For alle $x, y, z \in \mathbb{F}$ gælder $x \cdot (y + z) = x \cdot y + x \cdot z$.
 - 9 For alle $x \in \mathbb{F} \setminus \{0\}$ findes et $y \in \mathbb{F}$ således at $x \cdot y = 1$.

Hvad er et legeme?

- Mængden \mathbb{F} siges at være et legeme såfremt der findes to operationer $+$ og \cdot , samt to elementer $0, 1 \in \mathbb{F}$ således at:
 - ① For alle $x, y \in \mathbb{F}$ gælder $x + y = y + x$.
 - ② For alle $x, y, z \in \mathbb{F}$ gælder $(x + y) + z = x + (y + z)$.
 - ③ For alle $x \in \mathbb{F}$ gælder $x + 0 = x$.
 - ④ For alle $x \in \mathbb{F}$ findes et $y \in \mathbb{F}$ således at $x + y = 0$.
 - ⑤ For alle $x, y \in \mathbb{F}$ gælder $x \cdot y = y \cdot x$.
 - ⑥ For alle $x, y, z \in \mathbb{F}$ gælder $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
 - ⑦ For alle $x \in \mathbb{F}$ gælder $x \cdot 1 = x$.
 - ⑧ For alle $x, y, z \in \mathbb{F}$ gælder $x \cdot (y + z) = x \cdot y + x \cdot z$.
 - ⑨ For alle $x \in \mathbb{F} \setminus \{0\}$ findes et $y \in \mathbb{F}$ således at $x \cdot y = 1$.
- Legemer er altså mængder hvor vi kan regne *ligesom vi plejer*.
- Velkendte eksempler på legemer er \mathbb{Q} og \mathbb{R} , og fælles for disse er at begge indeholder uendeligt mange elementer.

Endelige legemer

Der findes legemer der kun indeholder *endeligt* mange elementer.
Vi vil kigge nærmere på de legemer hvor antallet af elementer er et
primtal p , og et sådant legeme noteres \mathbb{F}_p .

Endelige legemer

Der findes legemer der kun indeholder *endeligt* mange elementer. Vi vil kigge nærmere på de legemer hvor antallet af elementer er et primtal p , og et sådant legeme noteres \mathbb{F}_p .

- **Hvordan ser \mathbb{F}_p ud?**

$$\mathbb{F}_p = \{0, 1, 2, \dots, p - 1\}.$$

Endelige legemer

Der findes legemer der kun indeholder *endeligt* mange elementer. Vi vil kigge nærmere på de legemer hvor antallet af elementer er et primtal p , og et sådant legeme noteres \mathbb{F}_p .

- **Hvordan ser \mathbb{F}_p ud?**

$$\mathbb{F}_p = \{0, 1, 2, \dots, p - 1\}.$$

- **Hvordan regner man i \mathbb{F}_p ?**

Man regner helt som man “plejer” med hele tal, men man tager hele tiden rest ved division med p . Eksempelvis gælder følgende i legemet \mathbb{F}_{11} :

$$2 \cdot 7 + 1 = 14 + 1 = 15 = 4 \pmod{11}.$$

Endelige legemer

Der findes legemer der kun indeholder *endeligt* mange elementer. Vi vil kigge nærmere på de legemer hvor antallet af elementer er et primtal p , og et sådant legeme noteres \mathbb{F}_p .

- **Hvordan ser \mathbb{F}_p ud?**

$$\mathbb{F}_p = \{0, 1, 2, \dots, p - 1\}.$$

- **Hvordan regner man i \mathbb{F}_p ?**

Man regner helt som man “plejer” med hele tal, men man tager hele tiden rest ved division med p . Eksempelvis gælder følgende i legemet \mathbb{F}_{11} :

$$2 \cdot 7 + 1 = 14 + 1 = 15 = 4 \pmod{11}.$$

- Notation $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\}$.

Er \mathbb{F}_p overhovedet et legeme?

- Resten af a ved division med p skrives $a \pmod{p}$.
Operationen at tage rest ved division med p er ombyttelig med $+$ og \cdot , eller med andre ord

$$(a + b) \pmod{p} = (a \pmod{p}) + (b \pmod{p})$$

$$(a \cdot b) \pmod{p} = (a \pmod{p}) \cdot (b \pmod{p})$$

Kravene 1 til 8 gælder i de hele tal, og derfor opfyldes de også af \mathbb{F}_p .

Er \mathbb{F}_p overhovedet et legeme?

- Resten af a ved division med p skrives $a \pmod{p}$.
Operationen at tage rest ved division med p er ombyttelig med $+$ og \cdot , eller med andre ord

$$(a + b) \pmod{p} = (a \pmod{p}) + (b \pmod{p})$$

$$(a \cdot b) \pmod{p} = (a \pmod{p}) \cdot (b \pmod{p})$$

Kravene 1 til 8 gælder i de hele tal, og derfor opfyldes de også af \mathbb{F}_p .

- Vi beviser nu at krav 9 (eksistens af multiplikativ invers) også gælder i \mathbb{F}_p . Lad $a \in \mathbb{F}_p^*$, og betragt mængden

$$S = \{1 \cdot a, 2 \cdot a, \dots, (p-1) \cdot a\} \subseteq \mathbb{F}_p^*.$$

Lad $i \cdot a$ og $j \cdot a$ være to elementer S og antag $i > j$. Det påstås at elementerne er forskellige. Antag nemlig modsat at de er ens, så er

$$i \cdot a = j \cdot a \pmod{p} \Rightarrow (i - j) \cdot a = 0 \pmod{p}$$

Er \mathbb{F}_p overhovedet et legeme?

- Dette betyder at $p \mid (i - j) \cdot a$ og da p er et primtal og $1 \leq a \leq p - 1$ således at p ikke kan gå op i a , fås derfor at $p \mid (i - j)$. Men da $i > j$ og $1 \leq i, j \leq p - 1$ fås

$$i - j \in \{1, 2, \dots, p - 2\},$$

og altså kan p umuligt gå op i $i - j$. Dette er en modstrid og altså slutes at $i \cdot a$ og $j \cdot a$ er forskellige.

Er \mathbb{F}_p overhovedet et legeme?

- Dette betyder at $p \mid (i - j) \cdot a$ og da p er et primtal og $1 \leq a \leq p - 1$ således at p ikke kan gå op i a , fås derfor at $p \mid (i - j)$. Men da $i > j$ og $1 \leq i, j \leq p - 1$ fås

$$i - j \in \{1, 2, \dots, p - 2\},$$

og altså kan p umuligt gå op i $i - j$. Dette er en modstrid og altså sluttes at $i \cdot a$ og $j \cdot a$ er forskellige.

- Dette betyder at der er $p - 1$ elementer i S , og da

$$S \subseteq \mathbb{F}_p^*$$

fås altså at $S = \mathbb{F}_p^*$. Specielt gælder der at $1 \in S$, og altså findes i så $i \cdot a = 1$ hvilket viser at a har en invers.

Polynomier i $\mathbb{F}_p[X]$

Ligesom man kan have polynomier med rationelle eller reelle koefficienter, kan man også have polynomier med koefficienter i \mathbb{F}_p . Sådanne polynomier benævnes $\mathbb{F}_p[X]$.

- Vi betragter polynomiet $f(X) \in \mathbb{F}_{11}[X]$ givet ved

$$f(X) = X^{10} - 1.$$

Polynomier i $\mathbb{F}_p[X]$

Ligesom man kan have polynomier med rationelle eller reelle koefficienter, kan man også have polynomier med koefficienter i \mathbb{F}_p . Sådanne polynomier benævnes $\mathbb{F}_p[X]$.

- Vi betragter polynomiet $f(X) \in \mathbb{F}_{11}[X]$ givet ved

$$f(X) = X^{10} - 1.$$

- Evalueringen af f i visse af elementerne i \mathbb{F}_{11}^* udregnes

$$f(1) = 1^{10} - 1 = 1 - 1 = 0 \pmod{11}$$

$$f(2) = 2^{10} - 1 = 1024 - 1 = 0 \pmod{11}$$

$$f(9) = (-2)^{10} - 1 = 1024 - 1 = 0 \pmod{11}$$

$$f(10) = (-1)^{10} - 1 = 1 - 1 = 0 \pmod{11}$$

Hvis man regner videre kan man faktisk vise at alle elementerne i \mathbb{F}_p^* er rødder i $f(X)$.

Karakterisering af elementerne i \mathbb{F}_p^*

Sætning

Lad $a \in \mathbb{F}_p^*$, så er a en rod i polynomiet $f(X) = X^{p-1} - 1$.

Bevis.

Betragt produktet af alle elementerne i \mathbb{F}_p^*

$$s = 1 \cdot 2 \cdot \dots \cdot (p-1),$$

Fra tidligere ved vi at for $a \in \mathbb{F}_p^*$ gælder at $\{1 \cdot a, \dots, (p-1) \cdot a\}$ er lig med \mathbb{F}_p^* og altså er

$$a^{p-1}s = (1 \cdot a) \cdot (2 \cdot a) \cdot \dots \cdot ((p-1) \cdot a) = 1 \cdot 2 \cdot \dots \cdot (p-1) = s.$$

Dermed fås $a^{p-1} = 1$, hvillket viser at a er en rod i $f(X)$. □

Ordenen af et element i \mathbb{F}_p^*

For $a \in \mathbb{F}_p^*$ findes altså altid en potens af a der giver 1.

Definition (Orden)

Lad $a \in \mathbb{F}_p^*$. Ved *ordenen* af a forstås det mindste *positive* tal m således at $a^m = 1$. Denne orden benævnes $ord(a)$.

Sætning

Lad $a \in \mathbb{F}_p^*$ og lad n være således at $a^n = 1$, så vil $ord(a) \mid n$.

Bevis.

Skriv n på formen $n = q \cdot ord(a) + r$ med $0 \leq r < ord(a)$, så fås

$$1 = a^n = a^{q \cdot ord(a) + r} = (a^{ord(a)})^q \cdot a^r = (1)^q \cdot a^r = a^r.$$

Da ordenen er den *mindste positive* potens af a der giver 1 og da $r < ord(a)$ betyder dette at $r = 0$, og altså $ord(a) \mid n$. □

Ordener i \mathbb{F}_{11}

Sætning

Lad $a \in \mathbb{F}_p^*$, så vil $\text{ord}(a) \mid p - 1$.

Bevis.

Vi ved at $a^{p-1} = 1$ og sætningen fra forrige slide giver dermed $\text{ord}(a) \mid p - 1$ som ønsket. □

Eksempel

Ordenen af elementet 2 i \mathbb{F}_{11} findes ved at udregne potenser af 2 og se hvornår der fås 1. Bemærk at det ifølge sætningen kun er nødvendigt at beregne 2^d for d der går op i $p - 1 = 10$.

$$2^2 = 4 \neq 1 \pmod{11}$$

$$2^5 = 32 \neq 1 \pmod{11}$$

Altså er $\text{ord}(2) = 10$.

Primitivt element

Definition (Primitivt element)

Et element $a \in \mathbb{F}_p^*$ siges at være primitivt såfremt $\text{ord}(a) = p - 1$.

- Bemærk at da $\text{ord}(2) = 10$ i \mathbb{F}_{11} fås at alle tallene

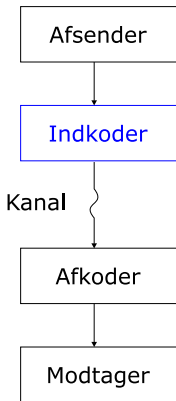
$$S = \{2^0, 2^1, \dots, 2^9\},$$

er forskellige, for hvis $2^i = 2^j$ fås $2^{i-j} = 1$ med $i - j < \text{ord}(a)$, hvilket er umuligt. Da $S \subseteq \mathbb{F}_{11}^*$ og antallet af elementer i begge mængder er 10 gælder derfor

$$\mathbb{F}_{11}^* = \{2^0, 2^1, \dots, 2^9\}.$$

- Vi har at 2 er et primitivt element \mathbb{F}_{11} , og man kan faktisk vise at ethvert endeligt legeme har et primitivt element.

Abstrakt matematik i anvendelse



- Efter at have stiftet bekendtskab med endelige legemer, kan vi konstruere en *indkoder*.
- Vi ønsker altså en funktion der for hver meddelelse giver et *kodeord*.
- Konstruktionen blev opdaget af Reed og Solomon, der også har lagt navn til koden, og er fra 1960.
- Selve grundidéen i konstruktionen er simpel og koderne er så gode at de i dag anvendes i fx CD- og DVD-afspillere.

Reed–Solomon koder

- Vi konstruerer en kode, der består af vektorer hvor hver plads er et element i et endeligt legeme \mathbb{F}_p .
- Hver meddelelse består af k elementer i \mathbb{F}_p .
- Kodeordene består af $n = p - 1$ elementer i \mathbb{F}_p .

Definition (Reed–Solomon kode)

Lad β være et primitivt element i \mathbb{F}_p^* . Kodeordene i Reed–Solomon koden er vektorerne af formen

$$\{(g(\beta^0), g(\beta^1), g(\beta^2), \dots, g(\beta^{n-1})) \mid g \in \mathbb{F}_p[X], \deg g < k\}.$$

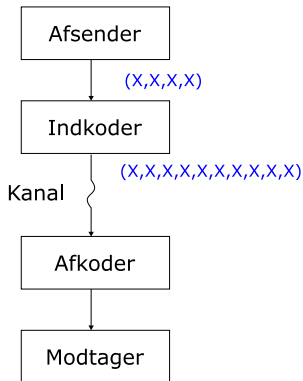
Bemærk at $\beta^0 = 1, \beta^1, \beta^2, \dots, \beta^{n-1}$ er alle de *forskellige* elementer i \mathbb{F}_p^* .

Eksempel: Reed–Solomon indkodning

- Arbejder over legemet \mathbb{F}_{11}^* , og vælger

$$n = 10, k = 4, \beta = 2$$

- Koden kan sende $11^4 = 14.641$ forskellige meddelelser.



Eksempel: Reed–Solomon indkodning

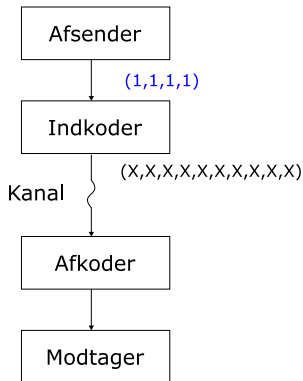
- Arbejder over legemet \mathbb{F}_{11}^* , og vælger

$$n = 10, k = 4, \beta = 2$$

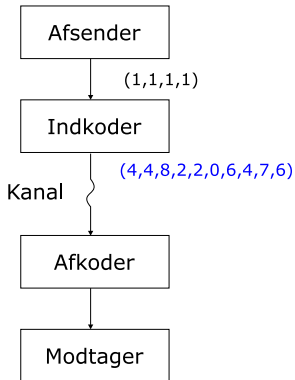
- Koden kan sende $11^4 = 14.641$ forskellige meddelelser.

- Meddelelsen:**

$$(1, 1, 1, 1) \longrightarrow g(X) = X^3 + X^2 + X + 1.$$



Eksempel: Reed–Solomon indkodning



- Arbejder over legemet \mathbb{F}_{11}^* , og vælger

$$n = 10, k = 4, \beta = 2$$

- Koden kan sende $11^4 = 14.641$ forskellige meddelelser.
- **Meddelelsen:**

$$(1, 1, 1, 1) \longrightarrow g(X) = X^3 + X^2 + X + 1.$$

- **Kodeordet:**

$$(g(2^0), g(2^1), g(2^2), \dots, g(2^9)) = (4, 4, 8, 2, 2, 0, 6, 4, 7, 6).$$

Egenskaber ved Reed–Solomon koder

Hvor tolerant overfor forekomster af fejl er en Reed–Solomon kode? Med andre ord, hvor mange fejl kan koden rette?

Egenskaber ved Reed–Solomon koder

Hvor tolerant overfor forekomster af fejl er en Reed–Solomon kode? Med andre ord, hvor mange fejl kan koden rette?

- Det vides fra tidligere at dette kan afgøres ud fra kodens minimumsafstand.
- Minimumsafstanden har en anden karakterisation. Lad c og c' være to kodeord, så er

$$\begin{aligned}c &= (c_0, c_1, \dots, c_{n-1}) \\c' &= (c'_0, c'_1, \dots, c'_{n-1}) \\c - c' &= (c_0 - c'_0, c_1 - c'_1, \dots, c_{n-1} - c'_{n-1})\end{aligned}$$

og altså er $\text{dist}(c, c')$ lig med antallet af pladser hvorpå $c - c'$ er forskelligt fra nul.

- Antallet af pladser i et ord der ikke er nul kaldes ordets *vægt*.

Egenskaber ved Reed–Solomon koder

- Forskellen den i 'te plads i to kodeord frembragt af f og g er:

$$\begin{aligned} f(\beta^i) - g(\beta^i) &= (f_{k-1}(\beta^i)^{k-1} + \dots + f_0) - \\ &\quad (g_{k-1}(\beta^i)^{k-1} + \dots + g_0) \\ &= (f_{k-1} - g_{k-1})(\beta^i)^{k-1} + \dots + (f_0 - g_0) \\ &= (f - g)(\beta^i) \end{aligned}$$

- Dette betyder at differensen mellem kodeordene frembragt af f og g igen er et kodeord, nemlig:

$$\begin{aligned} (f(\beta^0), \dots, f(\beta^{n-1})) - (g(\beta^0), \dots, g(\beta^{n-1})) &= \\ ((f - g)(\beta^0), \dots, (f - g)(\beta^{n-1})). \end{aligned}$$

Egenskaber ved Reed–Solomon koder

- Forskellen den i 'te plads i to kodeord frembragt af f og g er:

$$\begin{aligned} f(\beta^i) - g(\beta^i) &= (f_{k-1}(\beta^i)^{k-1} + \dots + f_0) - \\ &\quad (g_{k-1}(\beta^i)^{k-1} + \dots + g_0) \\ &= (f_{k-1} - g_{k-1})(\beta^i)^{k-1} + \dots + (f_0 - g_0) \\ &= (f - g)(\beta^i) \end{aligned}$$

- Dette betyder at differensen mellem kodeordene frembragt af f og g igen er et kodeord, nemlig:

$$\begin{aligned} (f(\beta^0), \dots, f(\beta^{n-1})) - (g(\beta^0), \dots, g(\beta^{n-1})) &= \\ ((f - g)(\beta^0), \dots, (f - g)(\beta^{n-1})). \end{aligned}$$

- Dette betyder at minimumsafstanden kan karakteriseres som:

$$\begin{aligned} d &= \min_{c \neq c'} \{\text{Antal pladser hvorpå } c \text{ og } c' \text{ er forskellige.}\} \\ &= \min_{c \neq 0} \{\text{Antal pladser hvorpå } c \text{ ikke er nul}\}. \end{aligned}$$

Egenskaber ved Reed–Solomon koder

Vi kan finde en nedre grænse på minimumsafstanden i en Reed–Solomon kode

$$\begin{aligned}d &= \min_{c \neq 0} \{\text{Antal pladser hvorpå } c \text{ ikke er nul}\} \\&= n - \max_{c \neq 0} \{\text{Antal pladser hvorpå } c \text{ er nul}\} \\&= n - \max_{\deg g \leq k-1} \{\text{Antal potenser hvor } g(\beta^i) = 0\} \\&\geq n - (k - 1) = n - k + 1.\end{aligned}$$

Egenskaber ved Reed–Solomon koder

- På den anden side fås ved valget

$$f(X) = (X - 1)(X - \beta)(X - \beta^2) \cdots (X - \beta^{k-2})$$

et polynomium af grad $k - 1$ hvis tilhørende kodeord er

$$(f(1), f(\beta), f(\beta^2), \dots, f(\beta^{n-1})) = (\underbrace{0, 0, \dots, 0}_{k-1 \text{ pladser}}, c_k, c_{k+1}, \dots, c_n),$$

hvor c_k, c_{k+1}, \dots, c_n alle er forskellige fra nul.

Egenskaber ved Reed–Solomon koder

- På den anden side fås ved valget

$$f(X) = (X - 1)(X - \beta)(X - \beta^2) \cdots (X - \beta^{k-2})$$

et polynomium af grad $k - 1$ hvis tilhørende kodeord er

$$(f(1), f(\beta), f(\beta^2), \dots, f(\beta^{n-1})) = (\underbrace{0, 0, \dots, 0}_{k-1 \text{ pladser}}, c_k, c_{k+1}, \dots, c_n),$$

hvor c_k, c_{k+1}, \dots, c_n alle er forskellige fra nul.

- Dette kodeord har vægt $n - (k - 1)$, og altså er:

$$d \leq n - (k - 1) = n - k + 1,$$

og dermed er minimumsafstanden for en Reed–Solomon kode

$$d = n - k + 1.$$

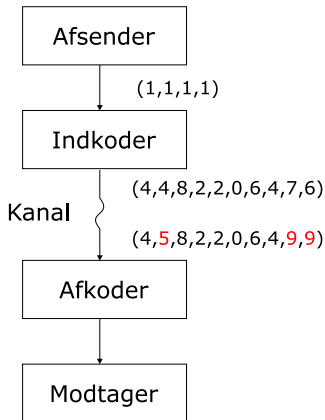
Fejl under transmissionen

Eksempel (Fortsat ...)

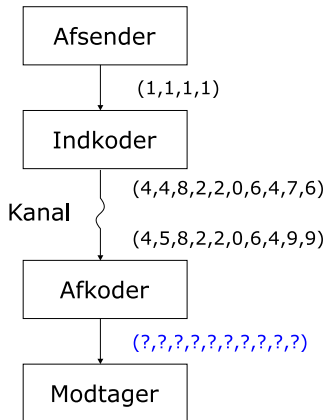
Koden i eksemplet fra før har længde $n = 10$ og dimension $k = 4$. Dermed er kodens minimumsafstand

$$d = n - k + 1 = 10 - 4 + 1 = 7,$$

og da $t = 3$ er strengt mindre end $d/2$ betyder det at koden kan rette tre fejl. Vi laver nu 3 fejl i kodeordet fra eksemplet.



Fejl under transmissionen



Eksempel (Fortsat ...)

Koden i eksemplet fra før har længde $n = 10$ og dimension $k = 4$. Dermed er kodens minimumsafstand

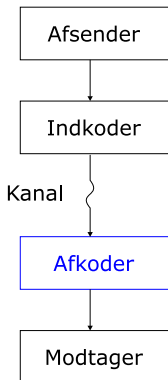
$$d = n - k + 1 = 10 - 4 + 1 = 7,$$

og da $t = 3$ er strengt mindre end $d/2$ betyder det at koden kan rette tre fejl. Vi laver nu 3 fejl i kodeordet fra eksemplet.

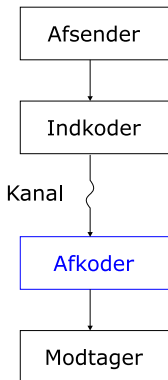
Hvordan kan afkoderen bestemme det oprindelige kodeord?

Afkodning af Reed–Solomon kode

- Givet en modtaget vektor $(r_0, r_1, \dots, r_{n-1})$, antager at der under transmissionen højst er påført $t < \frac{d}{2}$ fejl.



Afkodning af Reed–Solomon kode

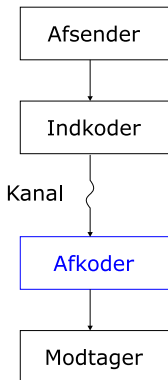


- Givet en modtaget vektor $(r_0, r_1, \dots, r_{n-1})$, antager at der under transmissionen højst er påført $t < \frac{d}{2}$ fejl.
- Vi ønsker at finde det entydigt bestemte polynomium $g(X) \in \mathbb{F}_p[X]$ af grad højst $k - 1$, der har frembragt dette kodeord, dvs. som opfylder

$$g(\beta^i) = r_i,$$

for alle de pladser hvor der ikke er sket fejl, dvs. for alle i pånær højst t pladser.

Afkodning af Reed–Solomon kode



- Givet en modtaget vektor $(r_0, r_1, \dots, r_{n-1})$, antager at der under transmissionen højst er påført $t < \frac{d}{2}$ fejl.
- Vi ønsker at finde det entydigt bestemte polynomium $g(X) \in \mathbb{F}_p[X]$ af grad højst $k - 1$, der har frembragt dette kodeord, dvs. som opfylder

$$g(\beta^i) = r_i,$$

for alle de pladser hvor der ikke er sket fejl, dvs. for alle i på nær højst t pladser.

- For nemheds skyld betragtes kun tilfældet hvor d er ulige, og altså opfylder antallet af fejl der kan rettes at $2t + 1 = d$.

Polynomier af to variable $\mathbb{F}_p[X, Y]$

For at løse afkodningsproblemet tager man udgangspunkt i det ikke åbentlyst beslægtede problem:

- Givet en modtaget vektor $(r_0, r_1, \dots, r_{n-1})$, find et polynomium forskelligt fra nulpolynomiet på formen

$$Q(X, Y) = Q_0(X) + YQ_1(X),$$

der opfylder

- ① $Q(\beta^i, r_i) = 0$ for alle $i \in \{0, 1, \dots, n-1\}$.
 - ② $\deg Q_0(X) \leq n-1-t$.
 - ③ $\deg Q_1(X) \leq n-1-t-(k-1) = n-t-k$.
- Her er β et primitivt element for det legeme koden er defineret over.
 - Et sådant polynomium kaldes et *interpoleringspolynomium*.
 - Definerer $l_0 = n-1-t$ og $l_1 = n-t-k$.

Interpoleringspolynomium

Det bemærkes at

$$\begin{aligned}l_0 + l_1 &= (n - 1 - t) + (n - t - k) = 2n - k - (2t + 1) \\ &= 2n - k - (n - k + 1) = n - 1.\end{aligned}$$

Interpoleringspolynomium

Det bemærkes at

$$\begin{aligned}l_0 + l_1 &= (n - 1 - t) + (n - t - k) = 2n - k - (2t + 1) \\ &= 2n - k - (n - k + 1) = n - 1.\end{aligned}$$

For en stund udskyder spørgsmålet om hvordan man rent faktisk for en givet vektor $(r_0, r_1, \dots, r_{n-1})$ finder et interpoleringspolynomium. Først nogle egenskaber ved dette:

Sætning

Hvis det sendte ord er $(c_0, c_1, \dots, c_{n-1})$ er frembragt af polynomiet $g(X)$ og det modtagne ord $(r_0, r_1, \dots, r_{n-1})$ er forskelligt herfra på højest t pladser, så er

$$g(X) = -\frac{Q_0(X)}{Q_1(X)}.$$

Interpoleringspolynomium

Bevis.

Polynomiet $T(X) = Q(X, g(X)) = Q_0(X) + g(X)Q_1(X)$, er polynomium udelukkende i variabelen X . Dets grad er

$$\begin{aligned} \deg T(X) &\leq \max \{ \deg Q_0(X), \deg g(X) \cdot \deg Q_1(X) \} \\ &\leq \max \{ \deg Q_0(X), \deg g(X) + \deg Q_1(X) \} \\ &\leq \max \{ n - 1 - t, (k - 1) + (n - 1 - t - k) \} \\ &= n - 1 - t. \end{aligned}$$

Hvis den i 'te plads i den modtagne vektor er blandt de mindst $n - t$ pladser i vektoren hvor der ikke er fejl, er $r_i = c_i = g(\beta^i)$ og dermed er

$$T(\beta^i) = Q(\beta^i, g(\beta^i)) = Q(\beta^i, r_i) = 0,$$

og altså har $T(X)$ mindst $n - t$ rødder.



Interpoleringspolynomium

Bevis fortsat ...

Altså er antallet af rødder i $T(X)$ større end den størst mulige grad af $T(X)$, og altså må $T(X)$ være nulpolynomiet. Specielt er

$$0 = Q(X, g(X)) = Q_0(X) + g(X)Q_1(X) \Rightarrow g(X) = -\frac{Q_0(X)}{Q_1(X)}.$$

□

Af sætningen følger at

$$\begin{aligned} Q(X, Y) &= Q_0(X) + YQ_1(X) = \\ &= Q_1(X)\left(Y + \frac{Q_0(X)}{Q_1(X)}\right) = Q_1(X)(Y - g(X)). \end{aligned}$$

Fejllokaliseringspolynomium

Lad nu i være en af de pladser hvor der er sket en fejl under transmissionen, dvs. $r_i \neq c_i$, så er

$$0 = Q(\beta^i, r_i) = Q_1(\beta^i)(r_i - g(\beta^i)) = Q_1(\beta^i) \underbrace{(r_i - c_i)}_{\neq 0},$$

og altså er $Q_1(\beta^i) = 0$. Dette betyder at de pladser hvorpå der er sket fejl, må være rødder i polynomiet $Q_1(X)$, der derfor kaldes et *fejllokaliseringspolynomium*.

Hvordan bestemmes $Q(X, Y)$?

Vi kan skrive polynomierne $Q_0(X)$ og $Q_1(X)$ på formen

$$Q_0(X) = a_{l_0}X^{l_0} + \cdots + a_1X + a_0$$

$$Q_1(X) = b_{l_1}X^{l_1} + \cdots + b_1X + b_0.$$

Kravet om at $Q(\beta^i, r_i) = 0$ for alle $i \in \{0, 1, \dots, n-1\}$ kan nu omformuleres til

$$\begin{aligned} 0 &= Q_0(\beta^i) + r_i Q_1(\beta^i) \\ &= a_{l_0}\beta^{il_0} + \cdots + a_1\beta^i + a_0 + r_i b_{l_1}\beta^{ih_1} + \cdots + r_i b_1\beta^i + r_i b_0 \end{aligned}$$

Denne benævnes *den i 'te ligning*.

Hvordan bestemmes $Q(X, Y)$?

Dette giver i alt n ligninger hvori de ubekendte er koefficienterne a_i og b_i . Anvendes nu at $2t = d - 1 = n - k$ fås at antallet af ubekendte er

$$(l_0 + 1) + (l_1 + 1) = l_0 + l_1 + 2 = n + 1.$$

Altså er der flere ubekendte end ligninger, og man kan derfor finde koefficienter der ikke alle er nul, der løser alle n ligninger.

Hvordan bestemmes $Q(X, Y)$?

Dette giver i alt n ligninger hvori de ubekendte er koefficienterne a_i og b_i . Anvendes nu at $2t = d - 1 = n - k$ fås at antallet af ubekendte er

$$(l_0 + 1) + (l_1 + 1) = l_0 + l_1 + 2 = n + 1.$$

Altså er der flere ubekendte end ligninger, og man kan derfor finde koefficienter der ikke alle er nul, der løser alle n ligninger.

Eksempel

I eksemplet med koden defineret over \mathbb{F}_{11} er $n = 10$ og altså kan man principielt finde alle koefficienterne a_i og b_i , og dermed også bestemme polynomiet $Q(X, Y)$, ved at løse et ligningssystem med 10 ligninger og 11 ubekendte. Regningerne kan dog være temmelig tidskrævende!

Afkobling af ligningssystemet

Ligningssystemet består af “to dele”, ét der involverer a_i 'erne og et der involverer b_i 'erne. Det viser sig nu at disse kan *afkobles*.

Sætning

Lad $a \in \mathbb{F}_p^* \setminus \{0, 1\}$, så vil

$$a^0 + a^1 + \dots + a^{p-2} = 0.$$

Bevis.

Lad S betegne summen $S = a^0 + a^1 + \dots + a^{p-2}$, så er

$$aS - S = (a^1 + a^2 + \dots + a^{p-1}) - (a^0 + a^1 + \dots + a^{p-2}) = a^{p-1} - 1.$$

Da $a \in \mathbb{F}_p^*$ har det orden $p - 1$ og altså fås $(a - 1)S = 1 - 1 = 0$ hvilket, da det er antaget at $a \neq 1$, betyder at $S = 0$ som ønsket. □

Afkobling af ligningssystemet

Fremgangsmåde ved afkobling af ligningssystemet

- For hvert $j \in \{1, 2, \dots, l_1\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$.

Afkobling af ligningssystemet

Fremgangsmåde ved afkobling af ligningssystemet

- For hvert $j \in \{1, 2, \dots, l_1\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$.
- For hvert sådant j fremkommer derved n nye ligninger og disse summeres.

Afkobling af ligningssystemet

Fremgangsmåde ved afkobling af ligningssystemet

- For hvert $j \in \{1, 2, \dots, l_1\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$.
- For hvert sådant j fremkommer derved n nye ligninger og disse summeres.
- Den derved fremkomne ligning kaldes den j 'te ligning, og der er altså l_1 ligninger i det nye ligningssystem.

Afkobling af ligningssystemet

Fremgangsmåde ved afkobling af ligningssystemet

- For hvert $j \in \{1, 2, \dots, l_1\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$.
- For hvert sådant j fremkommer derved n nye ligninger og disse summeres.
- Den derved fremkomne ligning kaldes den j 'te ligning, og der er altså l_1 ligninger i det nye ligningssystem.

Bemærk: Da $l_0 + l_1 = n - 1$ fås at hvis $j \in \{1, 2, \dots, l_1\}$ og $i \in \{0, 1, \dots, l_0\}$ er

$$i + j \in \{1, 2, \dots, n - 1\},$$

hvilket betyder at alle potenserne β^{i+j} forskellige fra 1.

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{r} a_{l_0}(\beta^0)^{l_0} + \cdots + a_1\beta^0 + a_0 \\ a_{l_0}(\beta^1)^{l_0} + \cdots + a_1\beta^1 + a_0 \\ \vdots \\ a_{l_0}(\beta^{n-1})^{l_0} + \cdots + a_1\beta^{n-1} + a_0 \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{l} (\beta^0)^j \cdot (a_{l_0}(\beta^0)^{l_0} + \dots + a_1\beta^0 + a_0) \\ (\beta^1)^j \cdot (a_{l_0}(\beta^1)^{l_0} + \dots + a_1\beta^1 + a_0) \\ \vdots \\ (\beta^{n-1})^j \cdot (a_{l_0}(\beta^{n-1})^{l_0} + \dots + a_1\beta^{n-1} + a_0) \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{l} a_{l_0}(\beta^0)^{j+l_0} + \dots + a_1(\beta^0)^{j+1} + a_0(\beta^0)^j \\ a_{l_0}(\beta^1)^{j+l_0} + \dots + a_1(\beta^1)^{j+1} + a_0(\beta^1)^j \\ \vdots \\ a_{l_0}(\beta^{n-1})^{j+l_0} + \dots + a_1(\beta^{n-1})^{j+1} + a_0(\beta^{n-1})^j \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{r} a_{l_0}(\beta^0)^{j+l_0} + \dots + a_1(\beta^0)^{j+1} + a_0(\beta^0)^j \\ a_{l_0}(\beta^1)^{j+l_0} + \dots + a_1(\beta^1)^{j+1} + a_0(\beta^1)^j \\ \vdots \\ a_{l_0}(\beta^{n-1})^{j+l_0} + \dots + a_1(\beta^{n-1})^{j+1} + a_0(\beta^{n-1})^j \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{array}{r} a_0(\beta^0)^j + a_0(\beta^1)^j + \dots + a_0(\beta^{n-1})^j = \\ a_0((\beta^j)^0 + (\beta^j)^1 + \dots + (\beta^j)^{n-1}) = 0 \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{cccc} a_{l_0}(\beta^0)^{j+l_0} + \dots + a_1(\beta^0)^{j+1} + a_0(\beta^0)^j & & & \\ a_{l_0}(\beta^1)^{j+l_0} + \dots + a_1(\beta^1)^{j+1} + a_0(\beta^1)^j & & & \\ \vdots & & \vdots & \vdots \\ a_{l_0}(\beta^{n-1})^{j+l_0} + \dots + a_1(\beta^{n-1})^{j+1} + a_0(\beta^{n-1})^j & & & \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{array}{rcl} a_1(\beta^0)^{j+1} + a_1(\beta^1)^{j+1} + \dots + a_1(\beta^{n-1})^{j+1} & = & \\ a_1((\beta^{j+1})^0 + (\beta^{j+1})^1 + \dots + (\beta^{j+1})^{n-1}) & = & 0 \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{ccccccc} a_{l_0}(\beta^0)^{j+l_0} & + \dots + & a_1(\beta^0)^{j+1} & + & a_0(\beta^0)^j & & \\ a_{l_0}(\beta^1)^{j+l_0} & + \dots + & a_1(\beta^1)^{j+1} & + & a_0(\beta^1)^j & & \\ & & \vdots & & \vdots & & \vdots \\ a_{l_0}(\beta^{n-1})^{j+l_0} & + \dots + & a_1(\beta^{n-1})^{j+1} & + & a_0(\beta^{n-1})^j & & \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{array}{rcl} a_{l_0}(\beta^0)^{j+l_0} + a_{l_0}(\beta^1)^{j+l_0} + \dots + a_{l_0}(\beta^{n-1})^{j+l_0} & = & \\ a_{l_0}((\beta^{j+l_0})^0 + (\beta^{j+l_0})^1 + \dots + (\beta^{j+l_0})^{n-1}) & = & 0 \end{array}$$

Afkobling af ligningssystemet

Vi kigger først på hvad der sker med den del af ligningssystemet der omfatter a_i 'erne.

$$\begin{array}{r} a_{l_0}(\beta^0)^{j+l_0} + \dots + a_1(\beta^0)^{j+1} + a_0(\beta^0)^j \\ a_{l_0}(\beta^1)^{j+l_0} + \dots + a_1(\beta^1)^{j+1} + a_0(\beta^1)^j \\ \vdots \\ a_{l_0}(\beta^{n-1})^{j+l_0} + \dots + a_1(\beta^{n-1})^{j+1} + a_0(\beta^{n-1})^j \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{aligned} a_{l_0}(\beta^0)^{j+l_0} + a_{l_0}(\beta^1)^{j+l_0} + \dots + a_{l_0}(\beta^{n-1})^{j+l_0} &= \\ a_{l_0}((\beta^{j+l_0})^0 + (\beta^{j+l_0})^1 + \dots + (\beta^{j+l_0})^{n-1}) &= 0 \end{aligned}$$

Altså indgår koefficienterne a_i ikke i den j 'te ligning.

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{rcccc} r_0 b_{l_1} (\beta^0)^{l_1} & + \cdots + & \beta^0 r_0 b_1 & + & r_0 b_0 \\ r_1 b_{l_1} (\beta^1)^{l_1} & + \cdots + & \beta^1 r_1 b_1 & + & r_1 b_0 \\ \vdots & & \vdots & & \vdots \\ r_{n-1} b_{l_1} (\beta^{n-1})^{l_1} & + \cdots + & \beta^{n-1} r_{n-1} b_1 & + & r_{n-1} b_0 \end{array}$$

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{r} (\beta^0)^j \cdot (r_0 b_{l_1} (\beta^0)^{l_1} + \dots + \beta^0 r_0 b_1 + r_0 b_0) \\ (\beta^1)^j \cdot (r_1 b_{l_1} (\beta^1)^{l_1} + \dots + \beta^1 r_1 b_1 + r_1 b_0) \\ \vdots \\ (\beta^{n-1})^j \cdot (r_{n-1} b_{l_1} (\beta^{n-1})^{l_1} + \dots + \beta^{n-1} r_{n-1} b_1 + r_{n-1} b_0) \end{array}$$

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{cccc} r_0 b_{l_1} (\beta^0)^{j+l_1} & + \dots + & r_0 b_1 (\beta^0)^{j+1} & + & r_0 b_0 (\beta^0)^j \\ r_1 b_{l_1} (\beta^1)^{j+l_1} & + \dots + & r_1 b_1 (\beta^1)^{j+1} & + & r_1 b_0 (\beta^1)^j \\ \vdots & & \vdots & & \vdots \\ r_{n-1} b_{l_1} (\beta^{n-1})^{j+l_1} & + \dots + & r_{n-1} b_1 (\beta^{n-1})^{j+1} & + & r_{n-1} b_0 (\beta^{n-1})^j \end{array}$$

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{ccccccc} r_0 b_{l_1} (\beta^0)^{j+l_1} & + \cdots + & r_0 b_1 (\beta^0)^{j+1} & + & r_0 b_0 (\beta^0)^j & & \\ r_1 b_{l_1} (\beta^1)^{j+l_1} & + \cdots + & r_1 b_1 (\beta^1)^{j+1} & + & r_1 b_0 (\beta^1)^j & & \\ \vdots & & \vdots & & \vdots & & \\ r_{n-1} b_{l_1} (\beta^{n-1})^{j+l_1} & + \cdots + & r_{n-1} b_1 (\beta^{n-1})^{j+1} & + & r_{n-1} b_0 (\beta^{n-1})^j & & \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{aligned} r_0 b_0 (\beta^0)^j + r_1 b_0 (\beta^1)^j + \cdots + r_{n-1} b_0 (\beta^{n-1})^j &= \\ b_0 (r_0 (\beta^j)^0 + r_1 (\beta^j)^1 + \cdots + r_{n-1} (\beta^j)^{n-1}) &= b_0 \cdot r(\beta^j) \end{aligned}$$

$$r(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}.$$

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{ccccccc} r_0 b_1 (\beta^0)^{j+1} & + \dots + & r_0 b_1 (\beta^0)^{j+1} & + & r_0 b_0 (\beta^0)^j & & \\ r_1 b_1 (\beta^1)^{j+1} & + \dots + & r_1 b_1 (\beta^1)^{j+1} & + & r_1 b_0 (\beta^1)^j & & \\ \vdots & & \vdots & & \vdots & & \\ r_{n-1} b_1 (\beta^{n-1})^{j+1} & + \dots + & r_{n-1} b_1 (\beta^{n-1})^{j+1} & + & r_{n-1} b_0 (\beta^{n-1})^j & & \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{aligned} r_0 b_1 (\beta^0)^{j+1} + r_1 b_1 (\beta^1)^{j+1} + \dots + r_{n-1} b_1 (\beta^{n-1})^{j+1} &= \\ b_1 (r_0 (\beta^{j+1})^0 + r_1 (\beta^{j+1})^1 + \dots + r_{n-1} (\beta^{j+1})^{n-1}) &= b_1 \cdot r(\beta^{j+1}) \end{aligned}$$

$$r(X) = r_0 + r_1 X + \dots + r_{n-1} X^{n-1}.$$

Afkobling af ligningssystemet

Nu ser vi på bidraget til den j 'te ligning fra den del af ligningssystemet der omfatter b_i 'erne.

$$\begin{array}{rcccc} r_0 b_h (\beta^0)^{j+h_1} & + \cdots + & r_0 b_1 (\beta^0)^{j+1} & + & r_0 b_0 (\beta^0)^j \\ r_1 b_h (\beta^1)^{j+h_1} & + \cdots + & r_1 b_1 (\beta^1)^{j+1} & + & r_1 b_0 (\beta^1)^j \\ \vdots & & \vdots & & \vdots \\ r_{n-1} b_h (\beta^{n-1})^{j+h_1} & + \cdots + & r_{n-1} b_1 (\beta^{n-1})^{j+1} & + & r_{n-1} b_0 (\beta^{n-1})^j \end{array}$$

Finder nu den j 'te ligning ved at summere de ovenstående ligninger. Går frem "søjlevis":

$$\begin{aligned} r_0 b_h (\beta^0)^{j+h_1} + r_1 b_h (\beta^1)^{j+h_1} + \cdots + r_{n-1} b_h (\beta^{n-1})^{j+h_1} = \\ b_h (r_0 (\beta^{j+h_1})^0 + r_1 (\beta^{j+h_1})^1 + \cdots + r_{n-1} (\beta^{j+h_1})^{n-1}) = b_h \cdot r(\beta^{j+h_1}) \end{aligned}$$

$$r(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}.$$

Eksempel

Altså bliver den j 'te ligning

$$0 = b_{l_1} r(\beta^{j+l_1}) + \dots + b_1 r(\beta^{j+1}) + b_0 r(\beta^j).$$

Pointen er at værdierne $r(\beta^j), \dots, r(\beta^{j+l_1})$ kan beregnes direkte udfra den modtagne vektor. Altså udgør disse ligninger igen et ligningssystem med b_i 'erne som ubekendte. Har altså nu l_1 ligninger med $l_1 + 1$ ubekendte.

Eksempel

I eksemplet fra tidligere har vi $n = 10$, $k = 4$ og $t = 3$. Det betyder at $l_1 = n - t - k = 3$. Det sendte kodeord er $c = (4, 4, 8, 2, 2, 0, 6, 4, 7, 6)$, mens det der modtages er $r = (4, 5, 8, 2, 2, 0, 6, 4, 9, 9)$.

Eksempel fortsat

Eksempel (Fortsat ...)

Det modtagne ord svarer til polynomiet

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4.$$

Vi skal løse $l_1 = 3$ ligninger med $l_1 + 1 = 4$ ubekendte:

$$0 = b_3 r(\beta^{1+3}) + b_2 r(\beta^{1+2}) + b_1 r(\beta^{1+1}) + b_0 r(\beta^1)$$

$$0 = b_3 r(\beta^{2+3}) + b_2 r(\beta^{2+2}) + b_1 r(\beta^{2+1}) + b_0 r(\beta^2)$$

$$0 = b_3 r(\beta^{3+3}) + b_2 r(\beta^{3+2}) + b_1 r(\beta^{3+1}) + b_0 r(\beta^3)$$

Eksempel fortsat

Eksempel (Fortsat ...)

Det modtagne ord svarer til polynomiet

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4.$$

Vi skal løse $l_1 = 3$ ligninger med $l_1 + 1 = 4$ ubekendte:

$$0 = b_3r(2^{1+3}) + b_2r(2^{1+2}) + b_1r(2^{1+1}) + b_0r(2^1)$$

$$0 = b_3r(2^{2+3}) + b_2r(2^{2+2}) + b_1r(2^{2+1}) + b_0r(2^2)$$

$$0 = b_3r(2^{3+3}) + b_2r(2^{3+2}) + b_1r(2^{3+1}) + b_0r(2^3)$$

Eksempel fortsat

Eksempel (Fortsat ...)

Det modtagne ord svarer til polynomiet

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4.$$

Vi skal løse $l_1 = 3$ ligninger med $l_1 + 1 = 4$ ubekendte:

$$0 = b_3 \cdot 7 + b_2 \cdot 6 + b_1 \cdot 9 + b_0 \cdot 4$$

$$0 = b_3 \cdot 9 + b_2 \cdot 7 + b_1 \cdot 6 + b_0 \cdot 9$$

$$0 = b_3 \cdot 8 + b_2 \cdot 9 + b_1 \cdot 7 + b_0 \cdot 6$$

Eksempel fortsat

Eksempel (Fortsat ...)

Det modtagne ord svarer til polynomiet

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4.$$

Vi skal løse $l_1 = 3$ ligninger med $l_1 + 1 = 4$ ubekendte:

$$0 = b_3 \cdot 7 + b_2 \cdot 6 + b_1 \cdot 9 + b_0 \cdot 4$$

$$0 = b_3 \cdot 9 + b_2 \cdot 7 + b_1 \cdot 6 + b_0 \cdot 9$$

$$0 = b_3 \cdot 8 + b_2 \cdot 9 + b_1 \cdot 7 + b_0 \cdot 6$$

Man kan ved at isolere og sætte ind, isolere og sætte, isolere osv. indse at $(b_3, b_2, b_1, b_0) = (1, 0, 3, 8)$ er en løsning, svarende til polynomiet

$$Q_1(X) = b_3X^3 + b_2X^2 + b_1X + b_0 = X^3 + 3X + 8.$$

Eksempel fortsat

Eksempel (Fortsat ...)

Rødderne i dette polynomium findes ved at evaluere det i alle elementerne i \mathbb{F}_{11}^*

| | | | | | |
|------------|-----|---|------------|-----|---|
| $Q_1(2^0)$ | $=$ | 1 | $Q_1(2^5)$ | $=$ | 4 |
| $Q_1(2^1)$ | $=$ | 0 | $Q_1(2^6)$ | $=$ | 5 |
| $Q_1(2^2)$ | $=$ | 7 | $Q_1(2^7)$ | $=$ | 9 |
| $Q_1(2^3)$ | $=$ | 5 | $Q_1(2^8)$ | $=$ | 0 |
| $Q_1(2^4)$ | $=$ | 5 | $Q_1(2^9)$ | $=$ | 0 |

Eksempel fortsat

Eksempel (Fortsat ...)

Rødderne i dette polynomium findes ved at evaluere det i alle elementerne i \mathbb{F}_{11}^*

| | | | | | |
|------------|-----|---|------------|-----|---|
| $Q_1(2^0)$ | $=$ | 1 | $Q_1(2^5)$ | $=$ | 4 |
| $Q_1(2^1)$ | $=$ | 0 | $Q_1(2^6)$ | $=$ | 5 |
| $Q_1(2^2)$ | $=$ | 7 | $Q_1(2^7)$ | $=$ | 9 |
| $Q_1(2^3)$ | $=$ | 5 | $Q_1(2^8)$ | $=$ | 0 |
| $Q_1(2^4)$ | $=$ | 5 | $Q_1(2^9)$ | $=$ | 0 |

Heraf ses at $2^1, 2^8$ og 2^9 er rødder i $Q_1(X)$ og altså sluttes at de eneste pladser i kodeordet hvor der kan være opstået fejl er på plads 1, 8 eller 9.

Eksempel fortsat

Eksempel (Fortsat ...)

Rødderne i dette polynomium findes ved at evaluere det i alle elementerne i \mathbb{F}_{11}^*

$$\begin{array}{ll} Q_1(2^0) = 1 & Q_1(2^5) = 4 \\ Q_1(2^1) = 0 & Q_1(2^6) = 5 \\ Q_1(2^2) = 7 & Q_1(2^7) = 9 \\ Q_1(2^3) = 5 & Q_1(2^8) = 0 \\ Q_1(2^4) = 5 & Q_1(2^9) = 0 \end{array}$$

Heraf ses at $2^1, 2^8$ og 2^9 er rødder i $Q_1(X)$ og altså sluttes at de eneste pladser i kodeordet hvor der kan være opstået fejl er på plads 1, 8 eller 9.

$$\begin{array}{l} c = (4, 4, 8, 2, 2, 0, 6, 4, 7, 6) \\ r = (4, 5, 8, 2, 2, 0, 6, 4, 9, 9) \end{array}$$

Hvordan findes fejlenes størrelse?

- Vi har nu værktøjer til at bestemme på hvilke pladser i en modtaget vektor, der er opstået fejl. Men for at kunne bestemme det oprindelige kodeord, må man også bestemme *størrelsen* af disse fejl.
- Hvis $(r_0, r_1, \dots, r_{n-1})$ er den modtagne vektor og $(c_0, c_1, \dots, c_{n-1})$ er det afsendte kodeord frembragt af polynomiet $g(X)$, er størrelsen af fejlen på plads i per definition

$$e_i = r_i - c_i = r_i - g(\beta^i).$$

Hvordan findes fejlenes størrelse?

- Vi har nu værktøjer til at bestemme på hvilke pladser i en modtaget vektor, der er opstået fejl. Men for at kunne bestemme det oprindelige kodeord, må man også bestemme *størrelsen* af disse fejl.
- Hvis $(r_0, r_1, \dots, r_{n-1})$ er den modtagne vektor og $(c_0, c_1, \dots, c_{n-1})$ er det afsendte kodeord frembragt af polynomiet $g(X)$, er størrelsen af fejlen på plads i per definition

$$e_i = r_i - c_i = r_i - g(\beta^i).$$

- Kaldes koefficienterne i det frembringende polynomium for $g(X) = g_{k-1}X^{k-1} + \dots + g_1X + g_0$ fås derfor

$$r_i = e_i + \left(g_{k-1}(\beta^i)^{k-1} + \dots + g_1\beta^i + g_0 \right)$$

for hvert $i \in \{0, 1, \dots, n-1\}$.

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{rcccl} g_{k-1}(\beta^0)^{k-1} & + \cdots + & g_1(\beta^0)^1 & & + g_0 \\ g_{k-1}(\beta^1)^{k-1} & + \cdots + & g_1(\beta^1)^1 & & + g_0 \\ \vdots & & \vdots & & \vdots \\ g_{k-1}(\beta^{n-1})^{k-1} & + \cdots + & g_1(\beta^{n-1})^1 & & + g_0 \end{array}$$

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{rcccc} g_{k-1}(\beta^0)^{k-1}(\beta^0)^j & + \dots + & g_1(\beta^0)^1(\beta^0)^j & + g_0(\beta^0)^j \\ g_{k-1}(\beta^1)^{k-1}(\beta^1)^j & + \dots + & g_1(\beta^1)^1(\beta^1)^j & + g_0(\beta^1)^j \\ \vdots & & \vdots & \vdots \\ g_{k-1}(\beta^{n-1})^{k-1}(\beta^{n-1})^j & + \dots + & g_1(\beta^{n-1})^1(\beta^{n-1})^j & + g_0(\beta^{n-1})^j \end{array}$$

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{r} g_{k-1}(\beta^0)^{k-1+j} + \dots + g_1(\beta^0)^{1+j} + g_0(\beta^0)^j \\ g_{k-1}(\beta^1)^{k-1+j} + \dots + g_1(\beta^1)^{1+j} + g_0(\beta^1)^j \\ \vdots \\ g_{k-1}(\beta^{n-1})^{k-1+j} + \dots + g_1(\beta^{n-1})^{1+j} + g_0(\beta^{n-1})^j \end{array}$$

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{r} g_{k-1}(\beta^0)^{k-1+j} + \dots + g_1(\beta^0)^{1+j} + g_0(\beta^0)^j \\ g_{k-1}(\beta^1)^{k-1+j} + \dots + g_1(\beta^1)^{1+j} + g_0(\beta^1)^j \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ g_{k-1}(\beta^{n-1})^{k-1+j} + \dots + g_1(\beta^{n-1})^{1+j} + g_0(\beta^{n-1})^j \end{array}$$

Summerer igen "søjlevis":

$$\begin{array}{r} g_0(\beta^0)^j + g_0(\beta^1)^j + \dots + g_0(\beta^{n-1})^j = \\ g_0((\beta^j)^0 + (\beta^j)^1 + \dots + (\beta^j)^{n-1}) = 0 \end{array}$$

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{r} g_{k-1}(\beta^0)^{k-1+j} + \dots + g_1(\beta^0)^{1+j} + g_0(\beta^0)^j \\ g_{k-1}(\beta^1)^{k-1+j} + \dots + g_1(\beta^1)^{1+j} + g_0(\beta^1)^j \\ \vdots \\ g_{k-1}(\beta^{n-1})^{k-1+j} + \dots + g_1(\beta^{n-1})^{1+j} + g_0(\beta^{n-1})^j \end{array}$$

Summerer igen "søjlevis":

$$\begin{array}{r} g_1(\beta^0)^{j+1} + g_1(\beta^1)^{j+1} + \dots + g_1(\beta^{n-1})^{j+1} = \\ g_1((\beta^{j+1})^0 + (\beta^{j+1})^1 + \dots + (\beta^{j+1})^{n-1}) = 0 \end{array}$$

Hvordan findes fejlenes størrelse?

Vi følger samme idé som før og eliminerer den del af ligningssystemet der involverer g_i 'erne.

- For hvert $j \in \{1, \dots, n - k\}$ ganges den i 'te ligning igennem med $(\beta^i)^j$ og ligningerne summeres.
- Bemærk at for sådanne j og for $i \in \{0, 1, \dots, k - 1\}$ er $i + j \in \{1, 2, \dots, n - 1\}$, så $\beta^{i+j} \neq 1$.
- Betragter den del af ligningssystemet der involverer g_i 'erne.

$$\begin{array}{r} g_{k-1}(\beta^0)^{k-1+j} + \dots + g_1(\beta^0)^{1+j} + g_0(\beta^0)^j \\ g_{k-1}(\beta^1)^{k-1+j} + \dots + g_1(\beta^1)^{1+j} + g_0(\beta^1)^j \\ \vdots \\ g_{k-1}(\beta^{n-1})^{k-1+j} + \dots + g_1(\beta^{n-1})^{1+j} + g_0(\beta^{n-1})^j \end{array}$$

Summerer igen "søjlevis":

$$\begin{aligned} g_{k-1}(\beta^0)^{k-1+j} + g_{k-1}(\beta^1)^{k-1+j} + \dots + g_{k-1}(\beta^{n-1})^{k-1+j} &= \\ g_{k-1}((\beta^{k-1+j})^0 + (\beta^{k-1+j})^1 + \dots + (\beta^{k-1+j})^{n-1}) &= 0 \end{aligned}$$

Hvordan findes fejlenes størrelse?

Dermed er g_i 'ernes bidrag til ligningssystemet "elimineret".

Betragtes den del af ligningssystemet der omfatter r_i og e_i 'erne fås:

$$\begin{array}{rcccc} r_0 & = & e_0 & + & \dots \\ \vdots & & \vdots & & \\ r_1 & = & e_1 & + & \dots \\ r_{n-1} & = & e_{n-1} & + & \dots \end{array}$$

Hvordan findes fejlenes størrelse?

Dermed er g_i 'ernes bidrag til ligningssystemet "elimineret".

Betragtes den del af ligningssystemet der omfatter r_i og e_i 'erne fås:

$$\begin{aligned} r_0(\beta^0)^j &= e_0(\beta^0)^j + \dots \\ &\vdots \\ r_1(\beta^0)^j &= e_1(\beta^0)^j + \dots \\ r_{n-1}(\beta^0)^j &= e_{n-1}(\beta^0)^j + \dots \end{aligned}$$

Den søjlevise summation giver nu:

$$\begin{aligned} e_0(\beta^0)^j + e_1(\beta^1)^j + \dots + e_{n-1}(\beta^{n-1})^j &= \\ e_0(\beta^j)^0 + e_1(\beta^j)^1 + \dots + e_{n-1}(\beta^j)^{n-1} &= e(\beta^j). \end{aligned}$$

$$e(X) = e_0 + e_1X + \dots + e_{n-1}X^{n-1}.$$

Hvordan findes fejlenes størrelse?

Dermed er g_i 'ernes bidrag til ligningssystemet "elimineret".

Betragtes den del af ligningssystemet der omfatter r_i og e_i 'erne fås:

$$\begin{array}{rcl} r_0(\beta^0)^j & = & e_0(\beta^0)^j + \dots \\ \vdots & & \vdots \\ r_1(\beta^0)^j & = & e_1(\beta^0)^j + \dots \\ r_{n-1}(\beta^0)^j & = & e_{n-1}(\beta^0)^j + \dots \end{array}$$

Den søjlevise summation giver nu:

$$\begin{aligned} r_0(\beta^0)^j + r_1(\beta^1)^j + \dots + r_{n-1}(\beta^{n-1})^j &= \\ r_0(\beta^j)^0 + r_1(\beta^j)^1 + \dots + r_{n-1}(\beta^j)^{n-1} &= r(\beta^j). \end{aligned}$$

$$r(X) = r_0 + r_1X + \dots + r_{n-1}X^{n-1}.$$

Hvordan findes fejlenes størrelse?

- Så for $j \in \{1, 2, \dots, n - k\}$ er $e(\beta^j) = r(\beta^j)$.
- Da vi har antaget at der højst forekommer t fejl, ved vi at højst t af koefficienterne e_i er forskellige fra nul.
- De indeks hvorpå fejlene forekommer kaldes i_1, i_2, \dots, i_t , og med denne notation fås:

$$\begin{aligned}r(\beta) &= e(\beta) = e_{n-1}\beta^{n-1} + \dots + e_1\beta + e_0 \\r(\beta^2) &= e(\beta^2) = e_{n-1}(\beta^2)^{n-1} + \dots + e_1\beta^2 + e_0 \\&\quad \vdots \quad \quad \quad \vdots \\r(\beta^t) &= e(\beta^t) = e_{n-1}(\beta^t)^{n-1} + \dots + e_1\beta^t + e_0\end{aligned}$$

Hvordan findes fejlenes størrelse?

- Så for $j \in \{1, 2, \dots, n - k\}$ er $e(\beta^j) = r(\beta^j)$.
- Da vi har antaget at der højst forekommer t fejl, ved vi at højst t af koefficienterne e_i er forskellige fra nul.
- De indeks hvorpå fejlene forekommer kaldes i_1, i_2, \dots, i_t , og med denne notation fås:

$$\begin{aligned}r(\beta) &= e_{i_t}(\beta)^{i_t} + \dots + e_{i_2}(\beta)^{i_2} + e_{i_1}(\beta)^{i_1} \\r(\beta^2) &= e_{i_t}(\beta^2)^{i_t} + \dots + e_{i_2}(\beta^2)^{i_2} + e_{i_1}(\beta^2)^{i_1} \\&\vdots \\r(\beta^t) &= e_{i_t}(\beta^t)^{i_t} + \dots + e_{i_2}(\beta^t)^{i_2} + e_{i_1}(\beta^t)^{i_1}\end{aligned}$$

Hvordan findes fejlenes størrelse?

- Så for $j \in \{1, 2, \dots, n - k\}$ er $e(\beta^j) = r(\beta^j)$.
- Da vi har antaget at der højst forekommer t fejl, ved vi at højst t af koefficienterne e_i er forskellige fra nul.
- De indeks hvorpå fejlene forekommer kaldes i_1, i_2, \dots, i_t , og med denne notation fås:

$$\begin{aligned}r(\beta) &= e_{i_t}(\beta)^{i_t} + \dots + e_{i_2}(\beta)^{i_2} + e_{i_1}(\beta)^{i_1} \\r(\beta^2) &= e_{i_t}(\beta^2)^{i_t} + \dots + e_{i_2}(\beta^2)^{i_2} + e_{i_1}(\beta^2)^{i_1} \\&\vdots \\r(\beta^t) &= e_{i_t}(\beta^t)^{i_t} + \dots + e_{i_2}(\beta^t)^{i_2} + e_{i_1}(\beta^t)^{i_1}\end{aligned}$$

- Dette kan betragtes som et ligningssystem i variablene $e_{i_1}, e_{i_2}, \dots, e_{i_t}$, og herudfra kan fejlstørrelserne findes.

Eksempel

Eksempel (Fortsat ...)

I eksemplet fra tidligere har vi

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4,$$

og vi ved at fejlene vil forekomme på plads $i_1 = 1$, $i_2 = 8$ eller $i_3 = 9$. Vi skal løse et ligningssystemet:

$$\begin{aligned}r(\beta) &= e_{i_3}(\beta)^{i_3} + e_{i_2}(\beta)^{i_2} + e_{i_1}(\beta)^{i_1} \\r(\beta^2) &= e_{i_3}(\beta^2)^{i_3} + e_{i_2}(\beta^2)^{i_2} + e_{i_1}(\beta^2)^{i_1} \\r(\beta^3) &= e_{i_3}(\beta^3)^{i_3} + e_{i_2}(\beta^3)^{i_2} + e_{i_1}(\beta^3)^{i_1}\end{aligned}$$

Eksempel

Eksempel (Fortsat ...)

I eksemplet fra tidligere har vi

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4,$$

og vi ved at fejlene vil forekomme på plads $i_1 = 1$, $i_2 = 8$ eller $i_3 = 9$. Vi skal løse et ligningssystemet:

$$\begin{aligned}r(2) &= e_9 \cdot (2^1)^9 + e_8 \cdot (2^1)^8 + e_1 \cdot (2^1)^1 \\r(2^2) &= e_9 \cdot (2^2)^9 + e_8 \cdot (2^2)^8 + e_1 \cdot (2^2)^1 \\r(2^3) &= e_9 \cdot (2^3)^9 + e_8 \cdot (2^3)^8 + e_1 \cdot (2^3)^1\end{aligned}$$

Eksempel

Eksempel (Fortsat ...)

I eksemplet fra tidligere har vi

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4,$$

og vi ved at fejlene vil forekomme på plads $i_1 = 1$, $i_2 = 8$ eller $i_3 = 9$. Vi skal løse et ligningssystemet:

$$4 = e_9 \cdot 6 + e_8 \cdot 3 + e_1 \cdot 2$$

$$9 = e_9 \cdot 3 + e_8 \cdot 9 + e_1 \cdot 4$$

$$6 = e_9 \cdot 7 + e_8 \cdot 5 + e_1 \cdot 8$$

Eksempel

Eksempel (Fortsat ...)

I eksemplet fra tidligere har vi

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4,$$

og vi ved at fejlene vil forekomme på plads $i_1 = 1$, $i_2 = 8$ eller $i_3 = 9$. Vi skal løse et ligningssystemet:

$$4 = e_9 \cdot 6 + e_8 \cdot 3 + e_1 \cdot 2$$

$$9 = e_9 \cdot 3 + e_8 \cdot 9 + e_1 \cdot 4$$

$$6 = e_9 \cdot 7 + e_8 \cdot 5 + e_1 \cdot 8$$

Som før kan man ved at isolere og sætte ind, se at der kun er én løsning og at denne er $(e_1, e_8, e_9) = (1, 2, 3)$.

Eksempel

Eksempel (Fortsat ...)

I eksemplet fra tidligere har vi

$$r(X) = 9X^9 + 9X^8 + 4X^7 + 6X^6 + 2X^4 + 2X^3 + 8X^2 + 5X + 4,$$

og vi ved at fejlene vil forekomme på plads $i_1 = 1$, $i_2 = 8$ eller $i_3 = 9$. Vi skal løse et ligningssystemet:

$$4 = e_9 \cdot 6 + e_8 \cdot 3 + e_1 \cdot 2$$

$$9 = e_9 \cdot 3 + e_8 \cdot 9 + e_1 \cdot 4$$

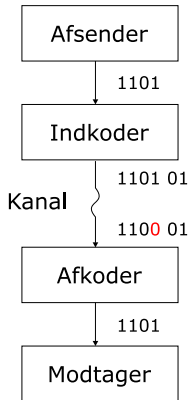
$$6 = e_9 \cdot 7 + e_8 \cdot 5 + e_1 \cdot 8$$

Som før kan man ved at isolere og sætte ind, se at der kun er én løsning og at denne er $(e_1, e_8, e_9) = (1, 2, 3)$.

$$c = (4, 4, 8, 2, 2, 0, 6, 4, 7, 6)$$

$$r = (4, 5, 8, 2, 2, 0, 6, 4, 9, 9)$$

Afrunding



- Vi har fulgt en meddelelse gennem alle faser fra afsender til modtager.
- I praksis anvendes oftest Reed–Solomon koder defineret over endelige legemer hvis størrelse er en potens af 2.
- I praksis anvendes en anden, men beslægtet, metode til *afkodning* end den der blev gennemgået her.

Supplerende materiale

- **Litteratur:**

A Course In Error–Correcting Codes

Jørn Justesen & Tom Høholdt

EMS Textbooks in Mathematics

Bogen er på engelsk og behandler blandt meget andet Reed–Solomon. Der forudsættes kendskab til lineær algebra.

Supplerende materiale

- **Litteratur:**

A Course In Error–Correcting Codes

Jørn Justesen & Tom Høholdt

EMS Textbooks in Mathematics

Bogen er på engelsk og behandler blandt meget andet Reed–Solomon. Der forudsættes kendskab til lineær algebra.

- **Applet:**

En applet der demonstrerer anvendelsen af Reed–Solomon koder i DVD–afspillere kan ses på

www2.mat.dtu.dk/people/T.Hoeholdt/DVD/index.html

Supplerende materiale

- **Litteratur:**

A Course In Error–Correcting Codes

Jørn Justesen & Tom Høholdt

EMS Textbooks in Mathematics

Bogen er på engelsk og behandler blandt meget andet Reed–Solomon. Der forudsættes kendskab til lineær algebra.

- **Applet:**

En applet der demonstrerer anvendelsen af Reed–Solomon koder i DVD–afspillere kan ses på

www2.mat.dtu.dk/people/T.Hoeholdt/DVD/index.html

- **Slides og opgaver:**

Slides anvendt i dag samt nogle opgaver kan findes på henholdsvis

www.mat.dtu.dk/people/P.Beelen/slides.pdf

www.mat.dtu.dk/people/P.Beelen/opgaver.pdf

Mere om primitive elementer

I det følgende vises at ethvert endeligt legeme \mathbb{F}_p har et primitivt element. Dertil vises først en række hjælpesætninger.

Sætning

Lad $a \in \mathbb{F}_p^*$ så vil $\text{ord}(a^n) = \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)}$, hvor *sfd* betegner største fælles divisor.

Bevis

- Der gælder

$$(a^n)^{\frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)}} = (a^{\text{ord}(a)})^{\frac{n}{\text{sfd}(\text{ord}(a), n)}} = 1,$$

og altså ved vi fra tidligere at $\text{ord}(a^n) \mid \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)}$.

Mere om $\text{ord}(a)$

- På den anden side gælder også

$$1 = (a^n)^{\text{ord}(a^n)} = a^{n \cdot \text{ord}(a^n)},$$

således at $\text{ord}(a) \mid n \cdot \text{ord}(a^n)$.

- Dette betyder at

$$\frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)} \mid \text{ord}(a^n),$$

og altså må $\text{ord}(a^n) = \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)}$, som ønsket.

Mere om $\text{ord}(a)$

- På den anden side gælder også

$$1 = (a^n)^{\text{ord}(a^n)} = a^{n \cdot \text{ord}(a^n)},$$

således at $\text{ord}(a) \mid n \cdot \text{ord}(a^n)$.

- Dette betyder at

$$\frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)} \mid \text{ord}(a^n),$$

og altså må $\text{ord}(a^n) = \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)}$, som ønsket.

Sætning

Lad m, n være hele tal der opfylder $\text{sfd}(m, n) = 1$. Da er det mindste fælles multiplum af m og n lig med nm .

Mere om $\text{ord}(a)$

Sætning

Lad $a, b \in \mathbb{F}_p^*$ være således at $\text{sfd}(\text{ord}(a), \text{ord}(b)) = 1$ så er

$$\text{ord}(ab) = \text{ord}(a) \cdot \text{ord}(b).$$

Bevis

- Først bemærkes at

$$(ab)^{\text{ord}(a) \cdot \text{ord}(b)} = (a^{\text{ord}(a)})^{\text{ord}(b)} \cdot (b^{\text{ord}(b)})^{\text{ord}(a)} = 1 \cdot 1 = 1,$$

og altså vil $\text{ord}(ab) \mid \text{ord}(a) \cdot \text{ord}(b)$, og dermed $\text{ord}(ab) \leq \text{ord}(a) \cdot \text{ord}(b)$.

- Lad nu n være et tal således at $(ab)^n = 1$ og sæt $x = a^n$ og dermed også $x = b^{-n}$.

Bevis fortsat

- Så fås af den forrige hjælpesætning

$$\text{ord}(x) = \text{ord}(a^n) = \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), n)} \mid \text{ord}(a).$$

$$\text{ord}(x) = \text{ord}(b^{-n}) = \frac{\text{ord}(b)}{\text{sfd}(\text{ord}(b), -n)} \mid \text{ord}(b).$$

Altså er $\text{ord}(x)$ en divisor i både $\text{ord}(a)$ og $\text{ord}(b)$ og da disse er primiske må $\text{ord}(x) = 1$, således at $x = 1$.

- Heraf fås $a^n = 1$ så $\text{ord}(a) \mid n$ og tilsvarende da $b^{-n} = 1$ fås $\text{ord}(b) \mid n$. Altså er n et multiplum af $\text{ord}(a)$ og $\text{ord}(b)$ der er primiske, og altså fås at

$$n \geq \text{mfm}(\text{ord}(a), \text{ord}(b)) = \text{ord}(a) \cdot \text{ord}(b).$$

Heraf sluttes at $\text{ord}(ab) = \text{ord}(a) \cdot \text{ord}(b)$ som ønsket.

Eksistens af primitivt element

Sætning

Der findes i \mathbb{F}_p^* et primitivt element.

Bevis.

Lad m være den største orden af et element i \mathbb{F}_p^* og lad a være et element af orden m . Lad dernæst b være et vilkårligt element i \mathbb{F}_p^* , og kald dets orden n .

Det vises nu indirekte at $n \mid m$, så antag modsat n ikke går op i m , så findes et primtal q der går op i n til en højere potens end det går op i m , og altså findes

$$m = q^i m', \quad n = q^j n',$$

hvor $j > i$ og $\text{sfd}(q, m') = \text{sfd}(q, n') = 1$. Nu fås at

$$\text{ord}(a^{q^i}) = \frac{\text{ord}(a)}{\text{sfd}(\text{ord}(a), q^i)} = \frac{m}{q^i} = m'.$$

Eksistens af primitivt element

Bevis fortsat ...

$$\text{ord}(b^{n'}) = \frac{\text{ord}(b)}{\text{sfd}(\text{ord}(b), n')} = \frac{n}{n'} = q^j.$$

Altså er $\text{sfd}(\text{ord}(a^{q^i}), \text{ord}(b^{n'})) = \text{sfd}(m', q^j) = 1$ og altså fås

$$\text{ord}(a^{q^i} b^{n'}) = \text{ord}(a^{q^i}) \cdot \text{ord}(b^{n'}) = m' \cdot q^j > m' q^i = m,$$

hvilket er en modstrid. Altså sluttes at $n \mid m$ og dette betyder at b er en rod i polynomiet $f(X) = X^m - 1$ da

$$b^m = (b^n)^{\frac{m}{n}} = 1.$$

Da b var valgt vilkårligt betyder det at alle $p - 1$ elementer i \mathbb{F}_p^* er rødder i $f(X)$. \square

Eksistens af primitivt element

Bevis fortsat ...

Da et polynomium højst kan have lige så mange rødder som størrelsen af dets grad fås

$$m = \deg f(X) \geq q - 1.$$

Ydermere da m er ordenen af et element i \mathbb{F}_p fås at $m \mid q - 1$ og specielt at $m \leq q - 1$. Altså sluttet $m = q - 1$ som ønsket. \square