Felrättande koder

Man bör inte äta för mycket

Jag har en bul
i

Jag hur en bil
a
y

I vanligt tal är felen olika lätta att rätta beroende på var i meningarna de förekommer. Använder man tex binär representation försvinner det problemet. Det är också lämpligt att dela upp texten så att alla ord är lika långa.

Antalet binära ord av längd n är 2^n st. V: betecknar mängden av sådana ord med V^n.

Ex: $V^2 = \{00, 01, 10, 11\}$

$V^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

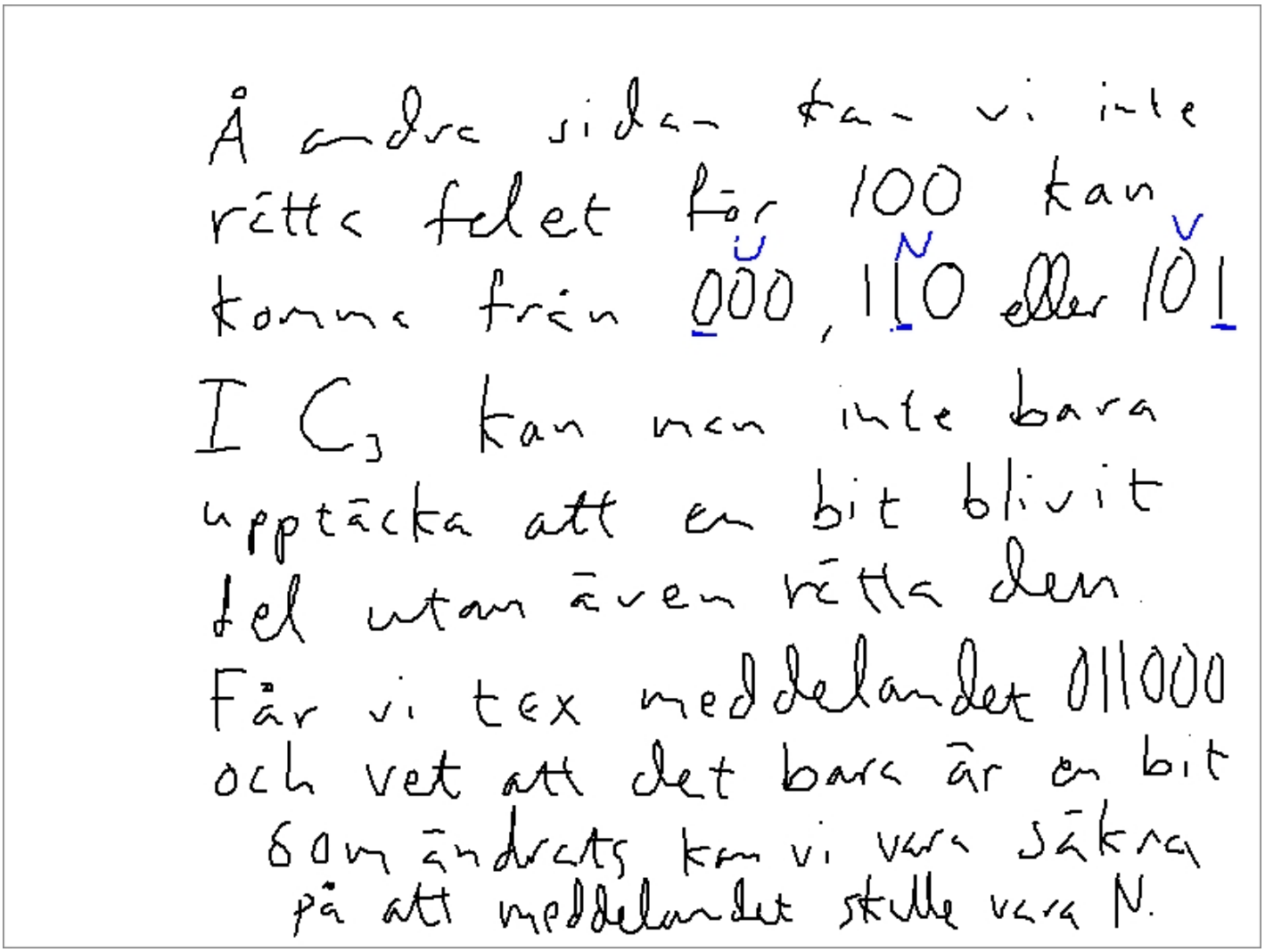
En delmängd, C, till V^n kallas en binär kod med längd n, och orden i C kallas kodord.

Målet är att välja C så att det blir lätt att både upptäcka och rätta till fel.

Ex: Antag vi vill kunna skicka meddelandena UPP, NED, HÖGER, VÄNSTER. 011000

		U	N	H	V
C_1	längd 2	00	01	10	11
C_2	längd 3	000	110	011	101
C_3	längd 6	000000	111000	001110	110011

Koden C_1 innehåller kortast ord och är därför snabbast att sända. Tyvärr, kommer minsta fel att feltolkas, tex om första biten blir fel kommer UPP att tolkas som HÖGER. I C_2 kan vi upptäcka om en bit ändras för tex 100 är inget kodord.


Å andra sidan kan vi inte
rätta felet för 100 kan
komma från 000, 110 eller 101
I C_3 kan man inte bara
upptäcka att en bit blivit
fel utan även rätta den.
Får vi tex meddelandet 011000
och vet att det bara är en bit
som ändrats kan vi vara säkra
på att meddelandet skulle vara N.

I praktiken vet man förstås inte hur många bitar som ändrats men att få få fel är mycket sannolikare. Om $v, w \in V^n$ så betecknar vi antalet bitar som skiljer dem åt med $\partial(v, w)$.

Ex: $1 = \partial(00, 01) = \partial(00, 10) = \partial(01, 11) = \partial(10, 11)$

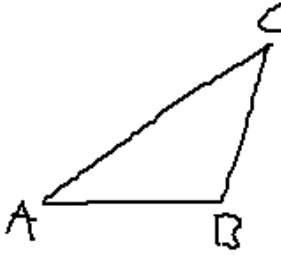
$\partial(00, 11) = \partial(01, 10) = 2$

Avståndet d har egenskaperna

i) $d(v,w) = 0 \iff v = w$

ii) $d(v,w) = d(w,v)$

iii) $d(v,w) \leq d(v,u) + d(u,w)$



Övn 17.1.4

$|AC| \leq |AB| + |BC|$

(vi har samma egenskaper hos $|x-y|$.)

Givet en kod C så låter vi d beteckna det minimala

avståndet mellan orden i C

$$\delta = \min\{d(a,b);\ a, b \in C, a \neq b\}$$

$$\delta_{C_1} = 1,\ \delta_{C_2} = 2,\ \delta_{C_3} = 3$$

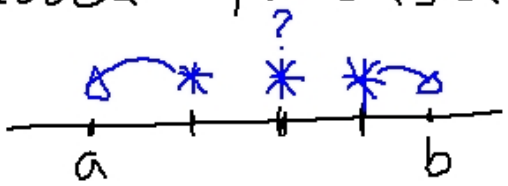
Det är klart att en kod kan upptäcka högst $\delta - 1$ fel för om vi fått δ fel kanske vi fått ett annat kodord.

Om man får ett felaktigt ord

så är det rimligast att anta att det ursprungliga kodordet var det som ligger närmast i Hamming till det mottagna ordet.

Tex i C_3. Om vi får ordet 011 000 så kan vi anta att det sända ordet var 111 000 dvs IVED. Den här metoden att rätta kallas närmaste granne metoden.

Sats: En kod C kan rätta e fel med närmaste granne metoden förutsatt att $\delta \geq 2e+1$.



$\delta = 4$ $e = 1$

Linjära koder

Låt oss definiera addition i V^n som komponentvis addition modulo 2. Tex

$001 + 101 = 100$. $V^n \simeq (\mathbb{Z}_2)^n$

V^n en grupp.

Def: En kod $C \subseteq V^n$ kallas linjär om $a, b \in C \Rightarrow a+b \in C$, dvs om C är en delgrupp till V^n.

Ex: $C_1 = V^2$ så linjär

$\{000, 110, 011, 101\} = C_2$ linjär för

$011 + 110 = 101$

$101 + 011 = 110$

$101 + 110 = 011$

C_3 ej linjär för

$111000 + 001110 = 110110 \notin C_3$.

Eftersom en linjär kod C är en delgrupp säger Lagranges sats att $|C| \mid |V^n| = 2^n \Rightarrow$

$\Rightarrow |C| = 2^k$ för något k.

$|C|$ kallas dimensionen hos C

Ex: Låt C vara en linjär kod av längd n och dimension 2^k. Visa att $2^{n-k} \geq 1 + \binom{n}{1} + \ldots + \binom{n}{e}$ där e är maximala antalet fel som C rättar.

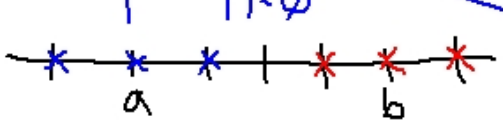
Lösn: Antalet sätt att ändra r bitar i ett ord av längd n är $\binom{n}{r}$ st så antalet ord vi kan få genom att ändra högst e st bitar är $1 + \binom{n}{1} + \ldots + \binom{n}{e}$.

Om nu C rättar e fel så kommer mängderna av modifierade ord inte skära varandra. Det betyder att

antal nya ord vi får från ett givet kodord

$$|V^n| \geq 2^k \cdot \left(1 + \binom{n}{1} + \ldots + \binom{n}{e}\right)$$

$|C|$



$$\Rightarrow \quad 2^{n-k} \geq \left(1 + \binom{n}{1} + \ldots + \binom{n}{e}\right),$$

vilket skulle visas.