Representasjon av tegn og tekst

seksjon 4.3

Tegn representeres i i komp.
en datamaskin vha. heltall som
peker inn i en tegntabell.

ASCII-tabellen, 7 bits, 128 tegn.
ISOLATIN - 8 bits, 256 tegn
ISOLATIN1 inneholder norske og andre
vesteuropeiske spesialtegn

Unicode - tabell som inneholder alle verdens tegn. Unicode har plass til ca 10^6 tegn, men bare ca. 100 000 er i bruk.

Hvis Unicode-kodene brukes direkte må hvert tegn ha 3 bytes.

En Unicode-kode kalles et kodepunkt.

I UTF-8 kodes kodepunktene på en slik måte at ASCII-tegn trenger en byte, andre latinske tegn 2 bytes, kinesiske tegn 3 bytes, og noen andre 4 bytes.

Tapsfri kompresjon

Idéen bak kompresjon er at de hyppigste tegnene får de korteste kodene.

Størrelse på filer.

Eks Lyd. På CD.

Lydsignalet måles 44100 pr. sekund i stereo, og hver måling lagres som et heltall med 2 bytes.

$44100 \cdot 2 \cdot 2 = 176\,000$ bytes pr. sekund

10MB = 10 000 000 bytes pr. min

Ca 40 MB for en låt, 600 MB på en CD.

Eks film. Klassisk TV-signal (PAL)

112 GB = $112 \cdot 10^9$ bytes pr. time.

Stort behov for komprimert lagring.

Huffman koding

seksjon 7.2 kompendiet.

$\bar{X} = \{x_1, x_2, \dots, x_m\}$ - tekst

Hver x_i er tatt fra et alfabet A, og $A = \{\alpha_1, \alpha_2, \dots \alpha_n\}$.

Antall forekomster av α_i i $\bar{x}$ er frekvensen $f(\alpha_i)$.

Kompresjon: finn kode $c(\alpha_i)$ for α_i og lagre x ved at hver x_j lagres ved sin kode.

Ex. Anta at $\bar{x}$ = DBACDBD

$f(A) = 1,\ f(B) = 2,\ f(C) = 1,\ f(D) = 3$

Kort kode til vanligste tegn.

$c(D) = 0,\ c(B) = 1,\ c(C) = 01,\ c(A) = 10$

Lagre kompakt: $z = 0110101010$

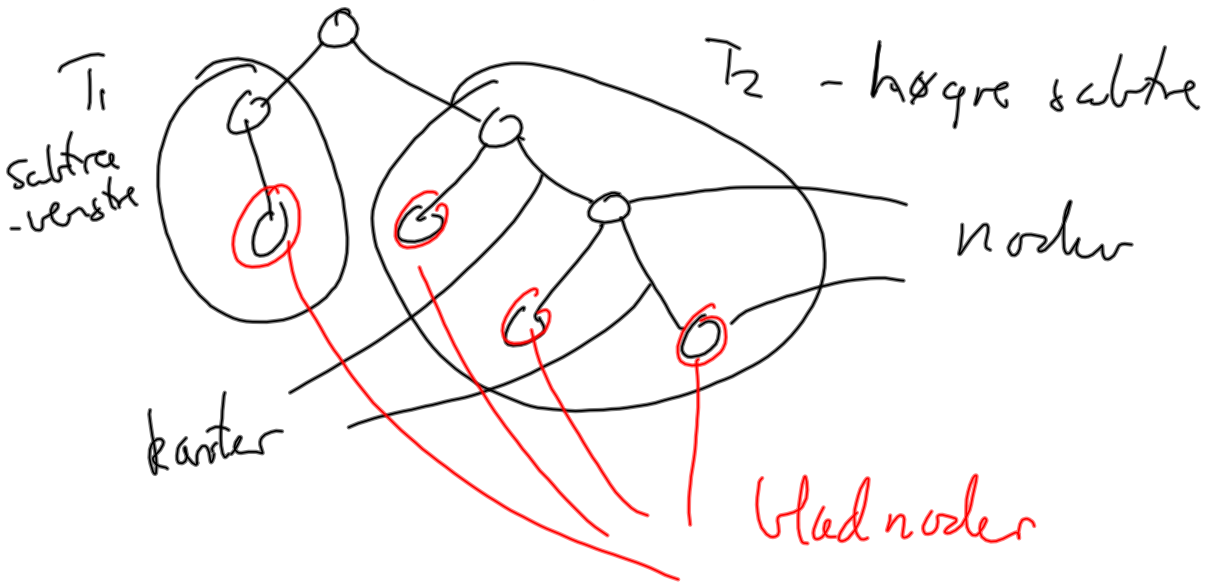
Ex. Annen koding

$c(D) = 1,\ c(B) = 01,\ c(C) = 001,\ c(A) = 000$

Da blir $z = 101000001101$

Koder må ha prefiks egenskapen, En kode må ikke være starten på en annen kode.

Binær trær


Binært tre, T
T1
- venstre subtre
T2 - høyre subtre
noder
kanter
blad noder