Kompresjon

Ex. 7.2, 7.4. Har teksten $x = DBACDBD$.

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

Bruker kodene $c(D) = 0,\ c(B) = 1,\ c(C) = 01,$
Erstatter hvert tegn med sin kode $c(A) = 10$

$z = 01100 1010$

Kan ikke tolke tilbake.

Ex. 7.4, $c(D) = 1,\ c(B) = 01,\ c(C) = 001,\ c(A) = 000$

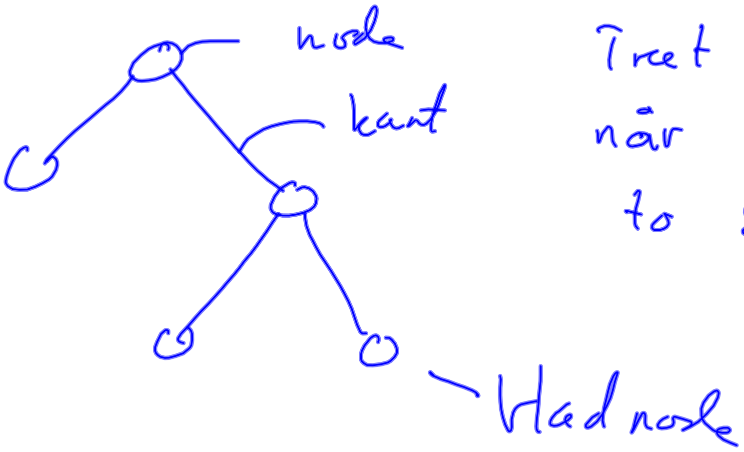
$z = 101000001101 1$

Med denne kodingen kan vi tolke tilbake.
Her er ingen koder starten på en annen
kode - prefiksegenskapen.

nov 20-10:12

Binære trær

Et tre består av noder og kanter.



Treet er binært når hver node har to subtrær.

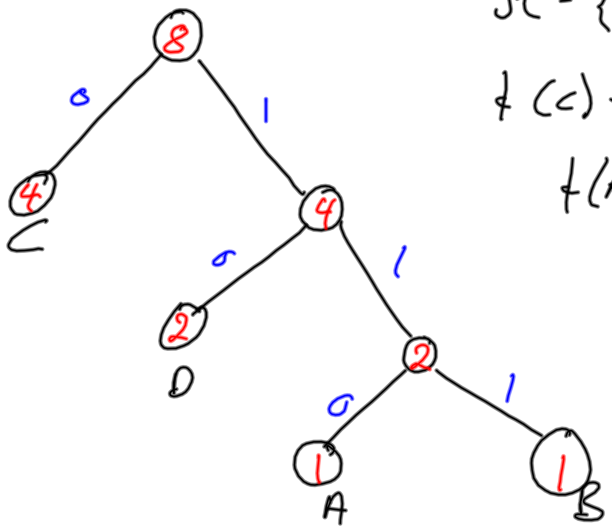
Huffman trær:

Et Huffman tre er et binært tre assosiert med et alfabet.

1. Hver bladnode svarer til ett symbol i alfabetet og vica versa.
2. Hver node har en vekt som er summen av vektene i de to subtrærne
3. Hver node har to eller ingen subtrær.
4. Venstre kant - 0
   Høyre kant - 1

Koden til et symbol fremkommer ved å følge kanter fra roten ned til symbolet

Ex.



$\Omega = \{A, B, C, D\}$

$f(C) = 4, \quad f(D) = 2$

$f(A) = 1, \quad f(B) = 1$

nov 20-10:41

Huffman-algoritmen.

Alg. 7.9. Anta at vi har et alfabet $\mathcal{A} = \{\alpha_i\}_{i=1}^{n}$ med frekvenser $f(\alpha_i)$. Huffmantre lages ved:

1. Lag en-node Huffmantrær med hvert av de n symbolene α_i, $i=1,\dots,n$ og tilhørende vekt. Gir en skog av n trær.
2. Gjenta inntil det bare er et tre:
   a) Velg to trær T_0 og T_1 med minimale vekter og erstatt dem med ett tre som har T_0 som venstre subtre og T_1 som høyre subtre.
3. Treet som framkommer er et Huffman-tre for det gitte alfabetet.

Theorem Huffmankoding er optimalt innenfor klassen av skjemaer basert på binærtrær med tegn i bladnodene.

nov 20-10:49

Informasjonsentropi

Anta at vi har en tekst X og alfabet $\{\alpha_i\}_{i=1}^{n}$ med frekvenser $f(\alpha_i)$. Anta at lengden til koden til α_i er $\ell(\alpha_i)$.

Da er den totale lengden til koden av X

$$B = \sum_{i=1}^{n} f(\alpha_i)\,\ell(\alpha_i)$$

Bedre mål er totalt antall bit delt på lengden m av teksten X, antall bit pr. tegn i gjennomsnitt

$$X = \{x_1, x_2, \dots, x_m\}$$

$$\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$$

$$\boxed{\begin{aligned}\tilde{B} &= \frac{1}{m}\sum_{i=1}^{n} f(\alpha_i)\,\ell(\alpha_i) \\ &= \sum_{i=1}^{n} \frac{f(\alpha_i)}{m}\,\ell(\alpha_i) \\ &= \sum_{i=1}^{n} P(\alpha_i)\,\ell(\alpha_i)\end{aligned}}$$

der $P(\alpha_i)$ er sannsynligheten for å finne en α_i i teksten X.

Shannons teorem. Det minimale antall bit pr. tegn som er mulig å oppnå ved koding basert på alfabetet $A = \{\alpha_1, \dots, \alpha_n\}$ der $P_i = P(\alpha_i)$ er sannsynligheten til α_i er gitt ved.

$$H = H(P_1, \dots, P_n) = -\sum_{i=1}^{n} P(\alpha_i) \log_2 P(\alpha_i)$$

$-\log_2 P(\alpha_i) = \log_2(1/P(\alpha_i))$

$$\tilde{B} = \sum_{i=1}^{n} P(\alpha_i)\, \ell(\alpha_i) \text{ - bit pr. tegn.}$$

Hva er $\log_2 x$.

$$2^{\log_2 x} = x$$

Ta ln

$$\log_2 x \cdot \ln 2 = \ln x$$

$$\log_2 x = \ln x / \ln 2$$