Shannonin ensimmäinen lause

Pro gradu
Maija-Liisa Metso
Matemaattisten tieteiden laitos
Oulun yliopisto
Syksy 2014

Sisältö

Tiivistelmä

Tässä työssä on esitelty Shannonin ensimmäinen lause, joka on eräs informaatioteorian perusteista. Työssä on myös esitelty Shannonin ensimmäisen lauseen todistaminen. Todistus pohjautuu Kraftin ja McMillanin epäyhtälöihin, joiden todistukset on myös esitelty. Shannonin ensimmäinen lause esittää rajat optimaaliselle koodaukselle eli kuinka lyhyeksi tietty viesti on mahdollista koodata häviöttömästi. Optimaalisesta koodauksesta on esitelty esimerkkinä Huffman-koodi. Työn lopussa on esitelty kaksi käytännön esimerkkiä Huffman-koodin käytöstä. Tässä pro gradu -tutkielmassa on käytetty lähteenä pääasiassa teosta [1].

1 Johdanto informaatioteoriaan

Informaatioteoria antaa vastauksen kahteen kysymykseen: (1) mikä on tiedon pakkauksen eli kompression maksimi ja (2) mikä on suurin saavutettavissa oleva tiedonsiirtonopeus. Ensimmäisen kysymyksen yhteydessä puhutaan käsitteestä entropia (H) ja toisen kysymyksen kohdalla on kyse kanavan kapasiteetista (C). Sekä entropia että kanavan kapasiteetti ovat informaatioteorian ydinkäsitteitä. Tässä työssä tutustutaan Claude Shannonin työssään asettamiin rajoihin kanavan kapasiteetille ja nämä rajat on määritelty entropian käsitteen kautta. Shannonin ensimmäisessä laissa käsitellään tiedonsiirron alarajaa eli kuinka tehokkaasti tietoa voidaan pakata ilman, että tietoa häviää. Shannonin toinen laki puolestaan käsittelee kuinka paljon tietoa voidaan maksimissaan siirtää virheettä tietyissä olosuhteissa eli tietynlaisessa kanavassa. Shannonin toinen laki on jätetty tämän työn ulkopuolelle ja tässä työssä keskitytään Shannonin ensimmäisen lain matemaattiseen todistamiseen. Lisäksi tässä työssä käsitellään eräs optimaalisen koodauksen toteuttava algoritmi eli Huffmanin koodausalgoritmi. Työn lopussa on käytännön esimerkkejä Huffman-koodauksen käytöstä.

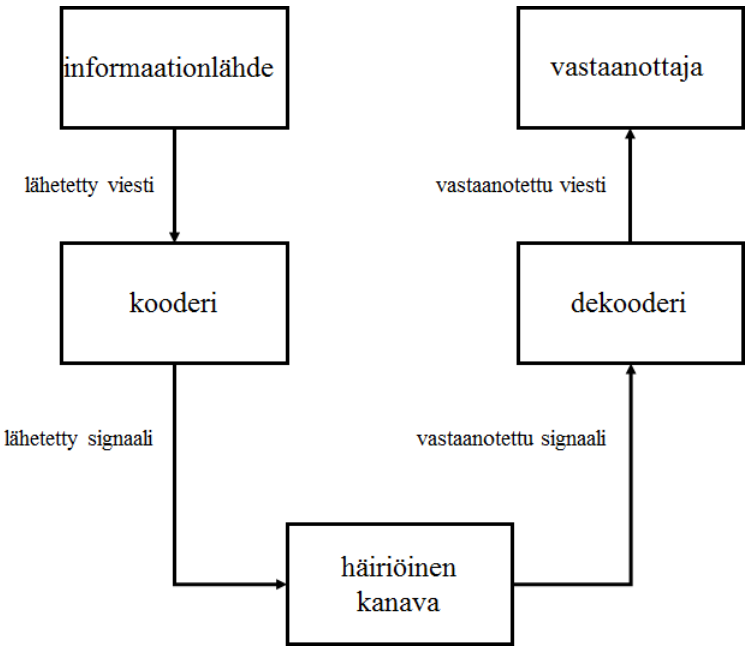
Informaatioksi voidaan käsittää kaikki tieto, jolla on jokin järjestys, joka voidaan tulkita äänenä, tekstinä, kuvana tai liikkuvana kuvana. Teknisesti ajateltuna informaatiolla tarkoitetaan laajasti kaikkea mikä voidaan lähettää informaatiokanavan kautta vastaanottajalle. Informaatio voi olla siis ääntä, kuvaa tai videota. Kaikki data mikä on havaittavissa joko kuulemalla tai näkemällä on siirrettävissä digitaalisesti. Esimerkiksi hajuja ja makuja ei vielä voida koodata muotoon, jossa ne voitaisiin siirtää, joten ne eivät kuulu informaation piiriin tässä yhteydessä.

Shannonin ensimmäinen laki määrittelee tiedonsiirtorajat informaatiolle ja se toimii kaikelle tiedolle, joka voidaan järjestää informaatioksi. Yleisim-

min Shannonin ensimmäistä lakia sovelletaan digitaalisissa sovelluksissa, joissa tieto koodataan aina nolliksi ja ykkösiksi. Tällöin puhutaan binaarisesta tiedosta. Tässä työssä käsitelläänkin pääasiassa binaaritietoa, mutta samat lait on myös laajennettavissa useamman merkin aakkostoille.

Informaatioteoriaan liittyy olennaisesti tiedonsiirto. Informaatio siirretään tiedonsiirtokanavaa pitkin ja tiedonsiirtokanava voidaan ymmärtää kaikkina välineinä, joiden kautta tietoa voidaan siirtää tai tallentaa ja siirtää. Tiedonsiirtokanavana voi toimia esimerkiksi ethernet, puhelin- tai radiokanava, perinteinen puhelinlinja tai DVD-laite. 2000-luvulla on yleisesti siirrytty käyttämään digitaalisia kanavia, joten informaatio koodataan ensin analogisesta muodosta digitaaliseksi ja vastaanotettaessa dekoodataan digitaalisesta muodosta analogiseksi. Ihmisen reseptorit eli korva ja silmä havannoivat analogisen ääni- tai kuvasignaalin, mutta tiedonsiirtokanavassa digitaalinen siirto kulkee nopeammin ja siitä on helpompi havaita siirtoprosessissa tapahtuneet virheet, jotka voidaan korjata dekoodauksen yhteydessä.

Kuvassa 1 on tiedonsiirtoprosessin vaiheet informaatiolähteestä, koodauksen, siirtokanavan ja dekoodauksen kautta vastaanottajalle.



Kuva 1: Tiedonsiirto koodaamalla ja dekoodaamalla viesti.

Informaatioteoriaa sovelletaan myös muuallakin kuin tietoliikenteessä. Sovelluksia löytyy mm. termodynamiikasta, tietotekniikasta, matematiikasta, taloustieteista, tilastotieteestä sekä todennäköisyyslaskennasta.

1.1 Informaatioteorian historiaa

Informaatioteoria nykyisen modernin tieteen muodossa alkoi 1900-luvun alussa, kun H. Nyquist ja R.V.L. Hartley tutkivat lennätinteknologioita AT&T:llä. Nyquistin työ liittyen lennätinteknologian nopeuteen julkaistiin vuonna 1924 [7] ja vuonna 1928 Hartley puolestaan julkaisi työstään artikkelin *Transmission of Information* [2]. Näitä artikkeleja pidetään ensimmäisinä tieteellisinä julkaisuina informaatioteorian alalta [4].

Toinen maailmansota siivitti informaatioteorien kehittymistä. Saksalaiset olivat kehittäneet Enigman - koneen, joka koodasi viestit salauksella, jota ulkopuolisten oli melkein mahdotonta purkaa. Liittoutuneet kehittivät kuitenkin salausta purkavan koneen ja onnistunut viestinpurku johti osaltaan lopulta liittoutuneiden voittoon.

Sodan jälkeen Claude Shannon (1916-2001) kehitti Bellin laboratoriossa informaatioteorian perustan määrittelemällä tiedonsiirron rajat, joista hän julkaisi vuonna 1948 artikkelin *A Mathematical Theory of Communications* [8]. Samaan aikaan informaatioteorian kehitykseen vaikutti myös Norbet Wiener, joka tutki signaalin erottamista häiriöstä häiriöisessä kanavassa [9].

Informaatioteorian päätarkoitus on tutkia tehokasta tiedonsiirtoa ja siihen liittyen tiedonsiirron nopeutta ja virhetodennäköisyyttä. Koska Shannonin työ on toiminut pohjana tähän aiheeseen, niin Shannonia pidetään informaatioteorian isänä. Informaatioteorian tutkimus on kehittynyt 1950-luvulta lähtien ja muodostaa nykyään merkittävän tieteenalan, johon esimerkiksi langattoman tiedonsiirron lainalaisuudet perustuvat [4].

2 Informaatio

Informaatioteoriassa määritellyn informaation idea on, että jonkin tapahtuman epävarmuus antaa enemmän tietoa kuin varman tiedon tapahtuminen. Informaatioteoriassa, jos epävarma tapahtuma tapahtuu, niin tällöin poistuu paljon epävarmuutta, joten itse informaatiosta saadaan tällöin paljon tietoa. Jos taas varma tapahtuma tapahtuu, niin silloin poistuu vain vähän epävarmuutta, joten tästä tapahtumasta saadaan vain vähän informaatiota.

Epävarmuuden eli informaation mitta on entropia. Seuraavassa kappaleessa on entropian matemaattinen määritelmä eli miten entropia saadaan laskettua jonkin satunnaismuuttujan pistetodennäköisyysfunktiosta.

2.1 Entropia

Olkoon $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus, jossa

- Ω on alkeistapausten joukko eli perusjoukko
- $\mathcal{F}$ on tapahtumien joukko (Ω:n osajoukkojen σ-algebra)
- $\mathbb{P}$ on todennäköisyys eli $\mathbb{P}$ on kuvaus $\mathcal{F} \to [0, 1]$

Entropian käsitettä määritettäessä käsitellään tässä työssä satunnaismuuttujia, joiden arvojoukko on äärellinen. Satunnaismuuttuja on kuvaus

$$X : \Omega \to \mathcal{X}$$

jossa $\mathcal{X}$ on äärellinen joukko. Lisäksi X:lle pätee

$$\{X = x\} = \{\omega \in \Omega \mid X(\omega) = x\} \in \mathcal{F}$$

kaikilla $x \in \mathcal{X}$.

Jokaiseen satunnaismuuttujaan X liittyy pistetodennäköisyys $p(x)$. Satunnaismuuttujan X pistetodennäköisyysfunktio on kuvaus $p : \mathcal{X} \to [0, 1]$, $p(x) = P\{X = x\}$. Pistetodennäköisyysfunktio voidaan merkitä myös käyttäen seuraavanlaista merkintää: $X \sim p(x)$.

Entropia määritellään pistetodennäköisyysfunktion avulla seuraavasti:

Määritelmä 2.1. Satunnaismuuttujan X entropia on

$$H(X) = -\sum_{x \epsilon \mathcal{X}} p(x) \log p(x). \tag{1}$$

Huomautus. Tässä yhteydessä on sovittu, että $0 \log 0 = 0$ (koska $\lim_{t \to 0^+} t \log t = 0$).

Toisaalta voidaan ajatella, että $H(X)$ on odotusarvo

$$H(X) = -\mathbb{E} \log p(X) = \mathbb{E}(-\log p(X)).$$

Nyt $\log p(X)$ on satunnaismuuttuja

$$\omega \longmapsto \log p(X(\omega)) = \log(\mathbb{P}\{X = X(\omega)\}).$$

Entropia $H(X)$ on satunnaismuuttujan X *arvojen keskimääräinen epävarmuus tai informaatiosisältö.*

Lause 2.2. *Olkoon* $0 < p_i,\ q_i \leq 1,\ i = 1, \ldots, m,\ \sum_{p=1}^m p_i = \sum_{i=1}^m q_i = 1$. *Silloin*

$$-\sum_{i=1}^m p_i \log p_i \leq -\sum_{i=1}^m p_i \log q_i \tag{2}$$

ja yhtäsuuruus pätee jos ja vain jos $p_i = q_i$ *kaikilla* i.

Todistus. Todistuksessa käytetään luonnollista logaritmia ln. Tunnetusti $\ln x \leq x - 1$ kaikilla $x > 0$, jossa yhtäsuuruus on voimassa jos ja vain jos $x = 1$. Tästä seuraa

$$\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$$

jossa yhtäsuuruus pätee jos ja vain jos $p_i = q_i$ $(i = 1, \ldots, m)$. Summaamalla useiden pistetodennäköisyyksien yli saadaan

$$\sum_{i=1}^m p_i \ln \frac{q_i}{p_i} \leq \sum_{i=1}^m p_i \Big(\frac{q_i}{p_i} - 1\Big) = \sum_{i=1}^m q_i - \sum_{i=1}^m p_i = 0$$

jossa yhtäsuuruus pätee jos ja vain jos $p_i = q_i$ kaikilla i. Väite saadaan nyt soveltamalla vasemmalla puolella yhtälöä logaritmin sääntöjä.

□

3 Häiriöttömän lähteen koodaus

Tietoa siirrettäessä yksittäisiä bittejä tai muita merkkejä asetetaan peräkkäin ja näin muodostuu koodisanoja. Satunnaismuuttujan X arvojoukko on $\mathcal{X}$ ja arvojoukon koko on $|\mathcal{X}| = m$. Arvojoukkoa $\mathcal{X}$ sanotaan *viestiaakkostoksi* ja viesti on jono symboleita $x_1 \dots x_k$, jossa $x_k \in \mathcal{X},\ k \in \mathbb{N}_+$. Kun kaikki mahdolliset viestit huomioidaan saadaan joukko

$$\mathcal{X}^* = \{x_1 \dots x_k | x_i \in \mathcal{X},\ i = 1, \dots, k,\ k \in \mathbb{N}_+\}.$$

Viestiä siirrettäessä viesti koodataan ja vastaanotettaessa dekoodataan. Koodausprosessissa viesti pyritään lähettämään mahdollisimman lyhyenä, jotta kanavassa ei siirtyisi turhaa tietoa. Shannonin ensimmäinen lause antaa alarajan virheettömästi dekoodattavan koodin keskimääräiselle pituudelle.

Viesti koodataan käyttäen *koodiaakkostoa* $\mathcal{D}$, jossa on $|\mathcal{D}| = D < \infty$ symbolia ja tällöin merkitään

$$\mathcal{D}^* = \{d_1 \dots d_k | d_i \in \mathcal{D},\ i = 1, \dots, k,\ k \in \mathbb{N}_+\}.$$

Määritelmä 3.1. Satunnaismuuttujan X *lähdekoodi* tai lyhyesti *koodi* on kuvaus $C : \mathcal{X} \to \mathcal{D}^*$.

Symbolia x vastaava koodisana on $C(x)$ ja sen pituutta merkitään $l(x)$. Joskus koodista puhuttaessa tarkoitetaan myös kuvaa $C(\mathcal{X})$.

Informaatiolähteestä peräisin oleva viesti koodataan symboli kerrallaan:

$$x_1 \dots x_k \to C(x_1) \dots C(x_k)$$

Kuvaus $\tilde{C} : \mathcal{X}^* \to \mathcal{D}^*$ on koodin C *laajennus*,

$$\tilde{C}(x_1 \dots x_k) = C(x_1) \dots C(x_k),$$

$x_1, \dots, x_k \in \mathcal{X}, k \in \mathbb{N}_+$.

Määritelmä 3.2. Koodi C on *ei-singulaarinen*, jos se on injektio, eli ehdosta $x_1 \neq x_2$ seuraa, että $C(x_1) \neq C(x_2), x_1, x_2 \in \mathcal{X}$.

Ei-singulaarisuus on minimivaatimus dekoodauksen täydelliselle onnistumiselle.

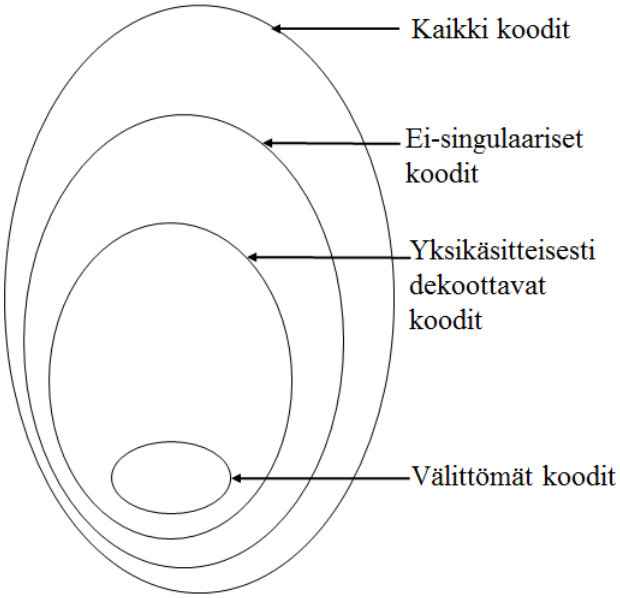
Määritelmä 3.3. Koodi C on *yksikäsitteisesti dekoodattavissa*, jos sen laajennus $\tilde{C}$ on ei-singulaarinen.

Huomautus. Yksikäsitteisesti dekoodattava koodi on aina ei-singulaarinen.

Seuraavassa määritelmässä etuliitteellä tarkoitetaan koodisanan alkuosaa.

Määritelmä 3.4. Koodi C on välitön, jos minkään symbolin $x \in \mathcal{X}$ koodisana ei ole jonkin toisen symbolin $x' \in \mathcal{X}$ koodisanan etuliite.

Kuvassa 2 on määritelty ei-singulaaristen, yksikäsitteisesti dekoodattavissa olevien koodien sekä välittömien koodien suhteet toisiinsa. Eli välittömät koodit (kuvassa 2 sisimmäinen alue) ovat aina myös yksikäsitteisesti dekoodattavia ja ei-singulaaria. Yksikäsitteisesti dekoodattavat koodit (kuvassa 2 toiseksi sisimmäinen alue) puolestaan ovat aina ei-singulaarisia, mutta eivät välttämättä välittömiä. Sen sijaan ei-singulaariset eivät välttämättä ole yksikäsitteisesti dekoodattavia eikä välittömiä (toki ne voivat myös sitä olla).



Kuva 2: Koodien luokat.

3.1 Kraftin epäyhtälö

Vuonna 1949 L.G. Kraft esitti epäyhtälön välittömille koodeille.

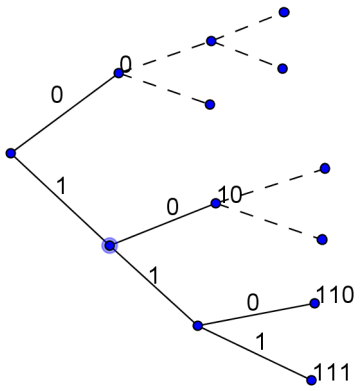
Olkoon $C : \mathcal{X} \to \mathcal{D}^*$ koodi, jolle $|\mathcal{X}| = m$, $|\mathcal{D}| = D$. Merkitään symbolien $x \in \mathcal{X}$ koodisanojen $C(x)$ pituuksia $l_1, \ldots, l_m$.

Lause 3.5. *(Kraftin epäyhtälö). On olemassa välitön koodi sanan pituuksilla $l_1, \ldots, l_m$, jos ja vain jos*

$$\sum_{i=1}^{m} D^{-l_i} \leq 1. \tag{3}$$

Todistus. Oletetaan, että on olemassa välitön koodi. Edelleen oletetaan, että käytetty koodiaakkosto on $\mathcal{D} = \{0, 1, \ldots, D-1\}$ ja koodisanojen pituudet ovat väliltä $l_1 \leq \ldots \leq l_m$. Koodiaakkosto voidaan havainnollistaa puurakenteena, jossa jokainen koodisana on solmukohta, jonka kertaluku eli jokaisen solmun lapsilehtien lukumäärä on D ja syvyys eli puun tasojen lukumäärä on l_m.

Kun tarkastellaan välitöntä koodia, niin koodisanan alipuu leikkautuu pois, koska tällöin tarkasteltavana oleva koodisana on alipuussa jokaisen koodisanan etuliite. Joten välitön koodi muodostuu vain niistä koodisanoista, jotka saadaan käymällä koodipuuta läpi vasemmalta oikealle edeten aina uutta haaraa pitkin uuteen solmuun, jossa on koodisana, jonka etuliite ei ole vielä käytössä.



Kuva 3: Välitön koodipuu, kun $|\mathcal{D}| = 2$ ja puun syvyys on $l_m = 3$.

Kuvassa 3 on esimerkki välittömästä koodista, jolle $\mathcal{D} = \{0, 1\}$ eli $|\mathcal{D}| = 2$ ja jossa puun syvyys on $l_m = 3$. Kun edetään puuta juuresta eteenpäin, niin

välitön koodi on $\{0, 10, 110, 111\}$. Eli jokaisen koodisanan valinnan jälkeen tämän kyseisen koodisanan alipuu leikkautuu pois. Kuvassa 3 nämä haarat on merkitty katkoviivoilla. Kolmannelta tasolta voidaan siis ottaa molemmat lehdet, koska kummankaan 110 ja 111 koodisanan alkuosa ei ole käytössä. Sen sijaan tasolta 2 voidaan käyttää vain koodisana 10 ja tasolta 1 vain koodisana 0.

Nyt koodisana, joka on tasolla l_i, leikkaa pois sen alla olevat $D^{l_m - l_i}$ "päätesolmua" eli lehteä. Täydellisessä puussa on päätesolmuja yhteensä D^{l_m} kappaletta. Jokainen välittömän koodin koodisana leikkaa pois sen alla olevat $D^{l_m - l_i}$ päätesolmua. Tällöin

$$\sum_{i=1}^{m} D^{l_m - l_i} \leq D^{l_m},$$

josta saadaan

$$\sum_{i=1}^{m} D^{-l_i} \leq 1.$$

Näin (3) on todistettu.

Toiseen suuntaan todistettaessa valitaan $l_1, \ldots, l_m \in \mathbb{N}_+$ siten, että (3) pätee. Edelleen oletetaan, että $l_1 \leq \ldots \leq l_m$. Välitön koodi luodaan valitsemalla m kappaletta solmuja kertalukua D olevasta puusta, jonka syvyys on l_m. Ensin valitaan jokin solmu ensimmäiseltä tasolta (l_1) koodisanaksi. Tällöin kyseisen haaran $D^{l_m - l_1}$ päätesolmua jää pois. Jos $m = 1$ lopetetaan ja jos $m \geq 2$, niin (3) mukaan

$$\sum_{i=1}^{m} D^{-l_i} \leq 1.$$

Kun tämä kerrotaan puolittain tekijällä D^{l_m}, saadaan

$$D^{l_m} \sum_{i=1}^{m} D^{-l_i} \leq D^{l_m},$$

joka on sievennettynä

$$\sum_{i=1}^{m} D^{l_m - l_i} \leq D^{l_m}.$$

Tässä

$$\sum_{i=1}^{m} D^{l_m - l_i} = D^{l_m - l_i} + \sum_{i=2}^{m} D^{l_m - l_1}.$$

Nyt $D^{l_m-l_1} < D^{l_m}$, joten jäljellä on vielä päätesolmuja. Siten tasolla l_2 on jäljellä solmuja, joten valitaan sieltä solmu koodisanaksi. Jos $m = 2$ lopetetaan.

Tilanteessa, jossa $m \geq 3$ saadaan

$$D^{l_m-l_1} + D^{l_m-l_2} + \sum_{i=3}^{m} D^{l_m-l_i} \leq D^{l_m},$$

jolloin $D^{l_m-l_1} + D^{l_m-l_2} < D^{l_m}$ ja päätesolmuja on vielä jäljellä. Tästä voimme päätellä, että tasolla l_3 on vielä solmuja. Valitaan tasolta yksi solmu koodisanaksi ja jatketaan kuten edellä, kunnes kaikki koodisanat on valittu.

□

3.2 McMillanin epäyhtälö

Vuonna 1956 Kraftin epäyhtälöön esitti yleistyksen B. McMillan julkaisussaan [6].

Lause 3.6. *(McMillanin epäyhtälö). On olemassa yksikäsitteisesti dekoodattava koodi koodisanan pituuksilla $l_1, \ldots, l_m$, jos ja vain jos (3) pätee, eli*

$$\sum_{i=1}^{m} D^{-l_i} \leq 1.$$

Todistus. Kun (3) pätee, löytyy välitön koodi sananpituuksilla $l_1, \ldots, l_m$, joka on siis myös yksikäsitteisesti dekoodattava.

Olkoon koodi C yksikäsitteisesti dekoodattavissa. Olkoon

$$\sum_{i=1}^{m} D^{-l_j} = \sum_{j=1}^{r} m_j D^{-j}, \tag{4}$$

jossa $r = max\{l_1, \ldots, l_m\}$ ja

$$m_j = |\{i | l_i = j\}|$$

on j:n pituisten koodisanojen lukumäärä. Kun $n \in \mathbb{N}_+$, saadaan

$$\begin{aligned} \left(\sum_{j=1}^{r} m_j D^{-j} \right)^n &= \sum_{i_1,\ldots,i_n=1}^{r} m_{i_1} D^{-i_1} \ldots m_{i_n} D^{-i_n} \\ &= \sum_{k=n}^{nr} \left(\sum_{i_1+\ldots+i_n=k} m_{i_1} \ldots m_{i_n} \right) D^{-k}. \end{aligned} \tag{5}$$

Huomataan, että $1 \leq i_1, \ldots, i_n \leq r$, jolloin $n \leq i_1 + \ldots + i_n \leq nr$. Nyt merkitään

$$N_k = \sum_{i_1+\ldots+i_n=k} m_{i_1} \ldots m_{i_n}.$$

Tällöin pätee:

$$N_k = \Big|\{x_1 \ldots x_n | \sum_{i=1}^{n} l(x_i) = k\}\Big|,$$

eli N_k on koodattuna k:n pituisten viestien lukumäärä. Koska koodi on yksikäsitteisesti dekoodattavissa, niin $N_k \leq D^k$. Nyt ehdon (5) nojalla

$$\left(\sum_{j=1}^{r} m_j D^{-j}\right)^n = \sum_{n=k}^{nr} N_k D^{-k} \leq \sum_{k=n}^{nr} 1 = nr - n + 1 \leq nr.$$

Sijoittamalla ylläoleva ehtoon (4) saadaan

$$\sum_{i=1}^{m} D^{-l_j} \leq (nr)^{\frac{1}{n}} = r^{\frac{1}{n}} n^{\frac{1}{n}}$$

Tästä seuraa väite, koska $n \in \mathbb{N}_+$ oli mielivaltainen ja

$$\lim_{n\to\infty} r^{\frac{1}{n}} n^{\frac{1}{n}} = \lim_{n\to\infty} r^{\frac{1}{n}} \lim_{n\to\infty} n^{\frac{1}{n}} = 1,$$

sillä $n^{\frac{1}{n}} = e^{\frac{1}{n}\ln n}$ ja $\frac{1}{n}\ln n = -\frac{1}{n}\ln\frac{1}{n} \to 0$, kun $n \to \infty$.

□

4 Shannonin ensimmäinen lause

Olkoon $X \sim p(x)$ ja olkoon X:n arvojoukko $\mathcal{X}, |\mathcal{X}| = m$. Tarkastellaan koodiaakkostoa $\mathcal{D}$, jonka pituus on $|\mathcal{D}| = D$ ja kuvausta $C : \mathcal{X} \to \mathcal{D}^*$. Nyt merkitään symbolien $x \in \mathcal{X}$ koodisanojen pituuksia $l(x)$ merkinnöillä $l_1, \ldots, l_m$ ja todennäköisyyksiä $p(x)$ merkinnöillä $p_1, \ldots, p_m$.

Määritelmä 4.1. Satunnaismuuttujan X koodin C *keskimääräinen koodin pituus on*

$$L = L(C) = \sum_{x \in \mathcal{X}} p(x) l(x) = \sum_{i=1}^{m} p_i l_i.$$

Lause 4.2. *Yksikäsitteisesti dekoodattavissa olevalle koodille pätee*

$$L \geq \frac{H(X)}{\log D} \tag{6}$$

ja yhtäsuuruus pätee jos ja vain jos $p_i = D^{-l_i}, i = 1, \ldots, m$.

Todistus. Merkitään

$$q_i = \frac{D^{-l_i}}{\sum_{j=1}^{m} D^{-l_j}}, \ i = 1, \ldots, m.$$

Tällöin $\sum_{i=1}^{m} q_i = 1$ ja epäyhtälön (2) perusteella

$$-\sum_{i=1}^{m} p_i \log p_i \leq -\sum_{i=1}^{m} p_i \log q_i.$$

Tällöin

$$\begin{aligned} H(X) &\leq -\sum_{i=1}^{m} p_i \log \left[\frac{D^{-l_i}}{\sum_{j=1}^{m} D^{-l_j}} \right] \\ &= -\sum_{i=1}^{m} p_i (-l_i) \log D + \sum_{i=1}^{m} p_i \log \left(\sum_{j=1}^{m} D^{-l_j} \right) \\ &= \log D \sum_{i=1}^{m} l_i p_i + \log \left(\sum_{j=1}^{m} D^{-l_j} \right). \end{aligned}$$

Nyt $\sum_{i=1}^{m} l_i p_i = L$ ja koska McMillanin epäyhtälön nojalla on $\sum_{j=1}^{m} D^{-l_j} \leq 1$, saadaan $H(X) \leq L \log D$. Täten (6) pätee.

Yllä oleva yhtälö pätee jos ja vain jos $p_i = q_i$ kaikilla $i = 1, \ldots, m$ kuten Lauseessa 2.2 todettiin. Lisäksi McMillanin epäyhtälössä täytyy olla voimassa yhtäsuuruus (Lause 3.6). Yhtälö pätee jos ja vain jos $p_i = D^{-l_i}$ kaikilla $i = 1, \ldots, m$.

□

Nyt voimme todeta, että keskimäärin lyhimmässä koodissa, joka on optimaalisin, on koodin todennäköisyyden p_i ja koodisanan pituuden l_i välillä suhde:

$$p_i \text{ suuri} \leftrightarrow l_i \text{ pieni}, \; p_i \text{ pieni} \; \leftrightarrow \; l_i \text{ suuri}.$$

D-kantainen entropia määritellään kaavalla

$$H_D(X) = -\sum_{x \in \mathcal{X}} p(x) \log_D p(x)$$

missä

$$\log_D t = \frac{\log t}{\log D}.$$

Tästä saamme

$$H_D(X) = \frac{H(X)}{\log D}.$$

Nyt epäyhtälö (6) saadaan muotoon $L \geq H_D(X)$.

Jos $p_i > 0, \; i = 1, \ldots, m$, niin optimaaliset koodisanojen pituudet olisivat sellaiset $l_i, \; i = 1, \ldots, m$, joille

$$D^{-l_i} = p_i$$

Tällöin

$$-l_i \log D = \log p_i,$$

josta saamme

$$l_i = \frac{\log \frac{1}{p_i}}{\log D}, \; i = 1, \ldots, m.$$

Koska l_i yllä olevassa kaavassa harvoin on kokonaisluku, joudumme pyöristämään sitä ylöspäin seuraavaan kokonaislukuun. Nyt saamme

$$l_i = \left\lceil \frac{\log \frac{1}{p_i}}{\log D} \right\rceil, \; i = 1, \ldots, m, \tag{7}$$

jolloin

$$\frac{\log \frac{1}{p_i}}{\log D} \leq l_i < \frac{\log \frac{1}{p_i}}{\log D} + 1, \ i = 1, \ldots, m.$$

Tästä saadaan

$$\sum_{i=1}^{m} \frac{p_i \log \frac{1}{p_i}}{\log D} \leq \sum_{i=1}^{m} l_i p_i < \sum_{i=1}^{m} \frac{p_i \log \frac{1}{p_i}}{\log D} + \sum_{m=1}^{m} p_i$$

ja kun korvataan entropian lauseke $H(X)$:llä, saadaan

$$\frac{H(X)}{\log D} \leq L < \frac{H(X)}{\log D} + 1.$$

Nyt pätee

$$\sum_{i=1}^{m} D^{-\left\lceil \frac{\log \frac{1}{p_i}}{\log D} \right\rceil} \leq \sum_{i=1}^{m} D^{-\frac{\log \frac{1}{p_i}}{\log D}} = \sum_{i=1}^{m} D^{\log_D p_i} = \sum_{i=1}^{m} p_i = 1,$$

joten Kraftin epäyhtälön nojalla on olemassa välitön koodi koodisanan pituuksilla (7).

Tällaista koodia sanotaan *Shannonin koodiksi*. Shannonin koodilla päästään yhden päähän keskimäärin lyhimmästä mahdollisesta koodisanan pituudesta.

Lause 4.3. *Keskimäärin lyhimmän välittömän koodin keskimääräiselle pituudelle L^* pätee*

$$\frac{H(X)}{\log D} \leq L^* < \frac{H(X)}{\log D} + 1. \tag{8}$$

Tuloksia (6) ja (8) sanotaan *Shannonin ensimmäiseksi lauseeksi*. Muita nimityksiä on "Source Coding Theorem"ja "Noiseless Coding Theorem".

4.1 Optimaalinen koodaus

Shannonin koodi ei kaikissa tilanteissa ole optimaalinen. Shannonin mukaan optimaalinen koodisanan pituus on $\lceil \log \frac{1}{p_i} \rceil$, mutta kaikissa tapauksissa tämä ei pidä paikkaansa. Esimerkiksi kahden symbolin viestissä, jossa todennäköisyydet ovat $p_1 = 0,9999$ ja $p_2 = 0,0001$, koodisanojen pituuksien tulisi vastaavasti olla $l_1 = 1$ bittiä ja $l_2 = 14$ bittiä.

Huffman esitteli julkaisussaan [3] oman versionsa optimaalisesta koodista, joka yleensä on lyhyempi kuin Shannonin koodi. Huffmanin mukaan optimaalinen välitön koodi voidaan rakentaa yksinkertaisella algoritmilla, jonka toimintaperiaate on esitetty alla olevassa taulukossa.

Koodisanan Pituus	Koodisana	X	Todennäköisyydet				
2	01	1	0,25	0,3	0,45	0,55	1
2	10	2	0,25	0,25	0,3	0,45	
2	11	3	0,2	0,25	0,25		
3	000	4	0,15	0,2			
3	001	5	0,15				

Kuva 4: Huffmanin koodin toimintaperiaate viidellä lähdesymbolilla.

Taulukon neljännessä sarakkeessa on kunkin aakkosen todennäköisyydet. Näistä kaksi vähiten todennäköistä todennäköisyyttä yhdistetään vaiheessa 1 ja järjestetään todennäköisyydet suuruusjärjestykseen (sarake 5). Vaiheessa 2 vastaavasti lasketaan yhteen kaksi sillä hetkellä epätodennäköisintä todennäköisyysarvoa ja saadaan uusi lista todennäköisyyksistä järjestettynä suurimmasta pienimpään (sarake 6). Vaihe 3 etenee laskemalla kaksi epätodennäköisintä todennäköisyyttä yhteen (sarake 7) ja vaihe 4 etenee samalla periaatteella, jolloin päädytään kaikkien todennäköisyyksien summaan. Tämän jälkeen taulukkoa käydään läpi oikealta vasemmalle ja aina haarautumiskohdassa ylempi haara saa arvon 0 ja alempi 1. Jokaisessa haarautumiskohdassa koodisana siis pitenee yhdellä. Kun taulukkoa edetään oikealta vasemmalla, niin lyhimmät koodisanat ovat 01, 10 ja 11 ja pisimmät 000 ja 001. Kahden pisimmän koodin tulee olla yhtä pitkiä, jotta niiden alkuosat eivät ole toistensa etuliitteitä ja saavutetaan välitön koodi.

4.1.1 Binaaristen Huffman-koodien optimaalisuus

Tässä kappaleessa todistetaan binaaristen Huffman-koodien optimaalisuus. Optimaalisilla koodeilla on yhteisiä piirteitä, jotka ovat voimassa myös ei-binaariselle Huffman-koodille.

Oletetaan, että lähdesymbolit on järjestetty niin, että niiden todennäköisyydet $p_1 \geq p_2 \geq \ldots \geq p_m$. Koodi on optimaalinen, jos $\sum p_i l_i$ saa minimiarvon. Tässä p_i on kunkin lähdesymbolin todennäköisyys ja l_i on kutakin lähdesymbolia vastaavan koodisanan pituus.

Lemma 4.4. *Mille tahansa jakaumalle on olemassa optimaalinen välitön koodi, joka toteuttaa seuraavat ehdot:*

1. *Jos $p_j > p_k$, niin silloin $l_j \leq l_k$.*
2. *Kaksi pisintä koodisanaa ovat yhtä pitkät.*
3. *Kaksi pisintä koodisanaa eroavat toisistaan vain viimeisen bitin osalta ja nämä koodisanat vastaavat kahta vähiten todennäköisintä lähdesymbolia.*

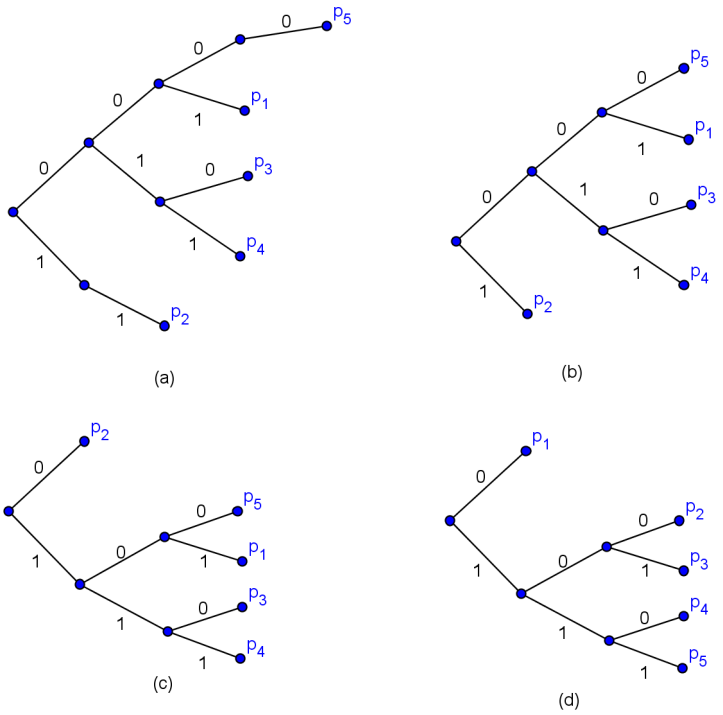
Todistus. Tarkastellaan optimaalista koodia C_m

- *Jos $p_j > p_k$, niin $l_j \leq l_k$.* Nyt valitaan koodisana C'_m, jossa alkuperäisen koodisanan C_m indeksit j ja k on vaihdettu keskenään. Silloin

$$\begin{aligned} L(C'_m) - L(C_m) &= \sum p_i l'_i - \sum p_i l_i \\ &= p_j l_k + p_k l_j - p_j l_j - p_k l_k \\ &= p_j(l_k - l_j) + p_k(l_j - l_k) \\ &= p_j(l_k - l_j) - p_k(l_k - l_j) \\ &= (p_j - p_k)(l_k - l_j) \end{aligned}$$

  Koska $p_j > p_k$, niin $p_j - p_k > 0$ ja koska C_m on optimaalinen, niin sitä vastaa myös lyhin mahdollinen koodisana, jolloin $L(C'_m) - L(C_m) \geq 0$. Joten täytyy olla $l_k \geq l_j$. Nyt siis $(p_j - p_k) > 0$ ja $l_k - l_j \geq 0$ eli $L(C'_m) - L(C_m) = (p_j - p_k)(l_k - l_j) \geq 0$ pitää paikkaansa. Optimaalinen koodi toteuttaa Lemman 4.4 kohdan 1 ehdon.

- *Kaksi pisintä koodisanaa ovat yhtä pitkät.* Jos kaksi pisintä koodisanaa eivät ole samanpituisia ja pidemmästä leikataan viimeinen bitti pois, niin päädytään kahteen yhtä pitkään koodisanaan ja samalla saavutetaan pienempi odotettu koodisanan pituus. Tämän takia kahden



Kuva 5: Optimaalisten koodien ominaisuuksia. Oletetaan, että $p_1 \geq p_2 \geq \ldots \geq p_m$. Kohdassa (a) on välitön koodi, jossa todennäköisyyksiä ei ole järjestetty suurimmasta pienimpään. Ensin koodipuusta leikataan ylimääräiset haarat eli ne joilla ei ole sisaruksia ja päädytään lyhyempään koodipuuhun (b). Kuvassa (c) puun haarat on järjestetty niin, että lyhimmät haarat on ylhäällä ja sen jälkeen todennäköisyydet järjestetään niin, että kaikista todennäköisin koodisana on lyhimmässä haarassa (d)

pisimmän koodisanan tulee olla samanpituisia. Tämän lemman ensimmäisen kohdan perusteella näiden kahden pisimmän koodisanan tulee kuulua kahdelle vähiten todennäköisimmälle lähdesymbolille.

- *Kaksi pisintä koodisanaa eroavat toisistaan vain viimeisen bitin osalta ja nämä koodisanat vastaavat kahta vähiten todennäköisintä lähdesymbolia.* Kaikki optimaaliset koodit eivät toteuta tätä ehtoa, mutta uudelleen järjestelemällä voidaan löytää koodi, joka toteuttaa ehdon.

  Jos on olemassa maksimipituinen koodisana, jolla ei ole sisarta eli koodipuun pisimmässä haarassa ei ole kahta koodisanaa, voidaan pisimmän koodisanan viimeinen bitti tuhota ja silti koodi on välitön eli toinen koodi ei ole toisen etuliite. Pisimmän koodisanan viimeisen bitin

tuhoaminen pienentää keskimääräistä koodinpituutta ja tällöin alkuperäinen koodi ei ollut optimaalisin. Tällöin optimaalisen koodin pisimmällä koodisanalla on aina sisar.

Kun pisimmällä koodisanalla on sisar eli kaksi pisintä koodisanaa ovat samanmittaisia, niin nämä koodisanat vastaavat kahta vähiten todennäköisinta lähdesymbolia ja koodisanojen vaihtaminen keskenään ei muuta koodisanojen pituuden odotusarvoa $\sum p_i l_i$. Eli kahta vähiten todennäköisintä lähdesymbolia vastaavat koodisanat ovat keskenään samanmittaisia ja ne eroavat toisistaan vain viimeisen bitin osalta.

Yhteenvetona voidaan todeta, että jos $p_1 \geq p_2 \geq \ldots \geq p_m$, niin silloin löytyy optimaalinen koodi, jolle $l_1 \leq l_2 \leq \ldots \leq l_{m-1} = l_m$ ja koodisanat $C(x_{m-1})$ ja $C(x_m)$ eroavat toisistaan vain viimeisen bitin osalta.

□

Kuvassa 5 on esitelty Lemman 4.4 määrittelemät ominaisuudet. Yllä todistettiin, että Lemman 4.4 määritelmät toteutuvat kaikille optimaalisille koodeille, joten voidaan rajoittua tarkastelemaan koodeja, jotka toteuttavat nämä määritelmät.

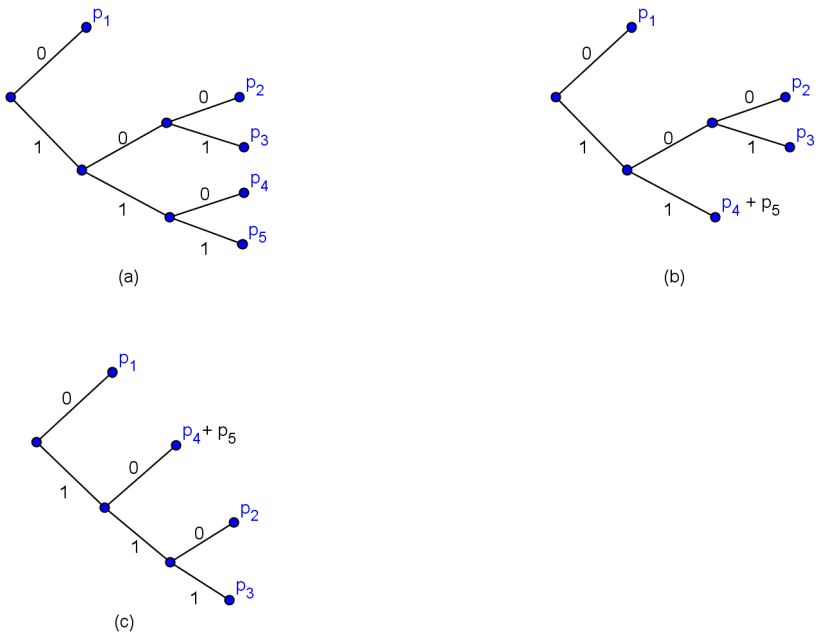
Koodeille C_m, jotka toteuttavat Lemman 4.4 määritellään yhdistetty koodi C_{m-1} $m-1$:lle symbolille seuraavasti: otetaan kahden pisimmän koodisanan yhteinen etuliiteja merkitään sen todennäköisyydeksi $p_{m-1} + p_m$. Kaikki muut koodisanat pysyvät samoina. Vastaavuudet ovat siis seuraavat:

	C_{m-1}		C_m	
p_1	w'_1	l'_1	$w_1 = w'_1$	$l_1 = l'_1$
p_2	w'_2	l'_2	$w_2 = w'_2$	$l_2 = l'_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_{m-2}	w'_{m-2}	l'_{m-2}	$w_{m-2} = w'_{m-2}$	$l_{m-2} = l'_{m-2}$
$p_{m-1} + p_m$	w'_{m-1}	l'_{m-1}	$w_{m-1} = w'_{m-1}0$	$l_{m-1} = l'_{m-1} + 1$
			$w_m = w'_{m-1}1$	$l_m = l'_{m-1} + 1$

Yllä w tarkoittaa binaarista koodisanaa ja l merkitsee koodisanan pituutta. Koodin C_m pituuden odotusarvo on

$$\begin{aligned} L(C_m) &= \sum_{i=1}^{m} p_i l_i \\ &= \sum_{i=1}^{m-2} p_i l'_i + p_{m-1}(l'_{m-1} + 1) + p_m(l'_{m-1} + 1) \\ &= \sum_{i=1}^{m-1} p_i l'_i + p_{m-1} + p_m \\ &= L(C_{m-1}) + p_{m-1} + p_m. \end{aligned}$$

Koodin C_m pituuden odotusarvo eroaa koodin C_{m-1} pituuden odotusarvosta aina tietyn kiinteän määrän verran, joka on $(p_{m-1}+p_m)$. Tämä arvo on riippumaton koodista C_{m-1}. Nyt $L(C_m)$:n minimointi tarkoittaa, että samalla minimoidaan myös $L(C_{m-1})$. Täten m:n symbolin optimointi on pienentynyt ongelmaan, jossa optimoidaan $m-1$ symbolia, joiden todennäköisyydet ovat $(p_1,\ p_2, \ldots,\ p_{m-2},\ p_{m-1}+p_m)$.



Kuva 6: Huffman-koodauksen vaiheet. Oletetaan, että $p_1 \geq p_2 \geq \ldots \geq p_m$. Kuvassa (a) on optimaalinen koodi, joka on järjestetty niin, että suurin todennäköisyys on ylimmässä haarassa ja muut todennäköisyydet laskevassa järjestyksessä siitä alaspäin. Kun kaksi vähiten todennäköisintä haaraa yhdistetään saadaan koodi kuten kuvassa (b). Kun koodit järjestetään uudelleen laskevaan järjestykseen saadaan kanoninen koodi kuten kuvassa (c) $m-1$:lle symbolille.

Kuvassa 6 on Huffman-koodauksen vaiheet, jossa optimointiongelma on pienentynyt m:stä symbolista $m-1$:een symboliin. Tulokseksi saatu $m-1$ symbolia toteuttaa Lemman 4.4 ehdot, niin voidaan jatkaa optimaalisen koodin etsimistä $m-2$:lle symbolille. Tällöin yhdistetään kahden vähiten todennäköisen symbolin todennäköisyydet ja saadaan uusi todennäköisyyksien lista, joka edelleen järjestetään suurimmasta todennäköisyydestä pienimpään. Nyt saadulle $m-2$:n symbolin optimaaliselle koodille voidaan tehdä sama, jos Lemman 4.4 ehdot toteutuvat. Näin etenemällä päädytään lopulta kah-

teen symboliin, joista toinen merkitään 0:lla ja toinen 1:llä. Koska jokaisessa vaiheessa on toteutettu optimaalisuuden ehdot, niin alkuperäinen m:n symbolin koodi on optimaalinen. Nyt olemme todistaneet seuraavan teoreeman binaarisille aakkostoille.

Teoreema 4.5. *Huffman koodi on välitön ja optimaalinen, toisin sanoen, jos C^* on Huffman-koodi ja C' on mikä tahansa muu välitön koodi, niin $L(C^*) \leq L(C')$.*

Yllä oleva todistus on binaariselle aakkostolle, mutta Huffman-koodauksen todistus voidaan laajentaa koskemaan myös D:n symbolin aakkostoa.

4.1.2 D:n symbolin aakkoston Huffman-koodit

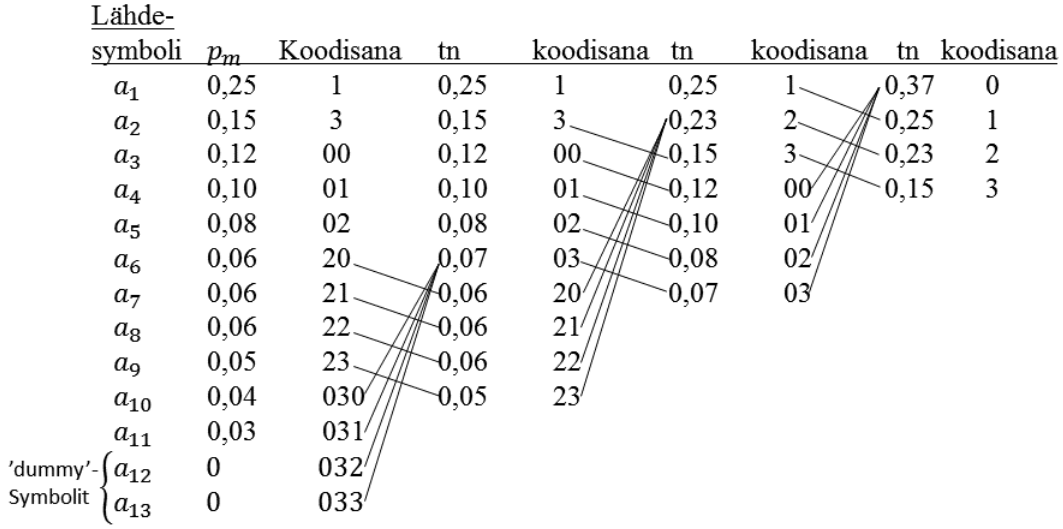
Huffman-koodi D:n symbolin aakkostolle luodaan samoilla periaatteilla kuin binaariselle aakkostolle. Koodin luonti eroaa siinä, että alussa on huomioitava, että lähdesymboleja täytyy olla M kappaletta alla olevan kaavan mukaisesti [4]:

$$M = r + \alpha(r - 1) \tag{9}$$

missä r on aakkoston koko ja α on toistojen lukumäärä.

Koska Huffman-koodauksessa tavoitteena on pienentää lähdesymbolien määrä siten, että niitä on jäljellä r kappaletta, alussa symboleja on oltava kaavassa (9) esitetty määrä. Muutoin käy niin, että viimeisessä vaiheessa symboleja on vähemmän kuin r-kappaletta, jolloin kaikkia aakkoston symboleja ei esiinny koodisanojen ensimmäisellä paikalla.

Kuvassa 7. on Huffmanin koodausprosessi kun lähdesymbolit ovat $A = \{a_1, a_2, \ldots, a_{11}\}$ ja koodiaakkosto on $\{0, 1, 2, 3\}$. Tällöin Huffmanin koodausprosessin mukaisia toistokertoja tulee 3, joten kaavan (9) mukaan lähdesymboleja tulisi olla $M = 4 + 3(4 - 1) = 13$ kappaletta. Tämän takia lähdesymbolilistaan joudutaan lisäämään kaksi symbolia a_{12} ja a_{13}, joiden todennäköisyyksiksi merkitään 0.



Kuva 7: Huffman-koodin luonti, kun lähdesymboleja on neljä.

Huffman-koodin luonti etenee kuten binaarisessa tapauksessa. Alla on lueteltu optimaalisen Huffman-koodin luonnin vaiheet, jossa lähdesymbolien

määrää pienennetään vaiheittain. On huomioitava, että vaiheessa 3 koodin luonti eroaa binaarisesta tapauksesta. Huffman koodin luonti sisältää seuraavat vaiheet viitteen [4] mukaan:

1. Lähdesymbolit järjestetään todennäköisyyksien mukaan alenevaan järjestykseen siten, että $p_1 \geq p_2 \geq \ldots \geq p_m$.

2. Määritetään r:n koodisymbolin kiinteä lista, esimerkiksi jos $r = 4$ niin $\{0, 1, 2, 3\}$.

3. Koska lähdesymbolien määrä halutaan vähentää r:n kappaleeseen, tarkistetaan, että lähdesymboleja on kaavassa (9) määritelty määrä. Jos symboleja on vähemmän, niin silloin lisätään niin sanottuja 'dummy' -symboleja niin, että lähdesymboleja on M-kappaletta.

   Huffman-koodin periaatteen mukaan lasketaan yhteen r:n vähiten todennäköisimmän lähdesymbolin todennäköisyydet. Seuraavassa vaiheessa tämä yhteenlaskettu todennäköisyys sijoitetaan muiden lähdesymbolien todennäköisyyslistaan omalle paikalleen eli niin, että sitä ylempänä on isommat todennäköisyydet ja alapuolella pienemmät todennäköisyydet. Kuvassa 7 neljännessä sarakkeessa on ensimmäisen vaiheen jälkeinen todennäköisyyksien lista. Nyt käsiteltävänä on $(r-1)$ symbolia vähemmän kuin ensimmäisessä vaiheessa. Jokaisessa vaiheessa symbolien määrä siis pienenee $(r-1)$:llä kappaleella. Tämän takia kaavassa 9 edellä kuvatun prosessin toistojen lukumäärä, joka on merkitty kirjaimella α, kerrotaan $(r-1)$:llä.

   Koodausprosessia jatketaan laskemalla yhteen r:n vähiten todennäköisimmän symbolin todennäköisyydet ja uudet todennäköisyydet listataan taas uudestaan suurimmasta pienimpään (kuvassa 7 sarakkeet 5-8). Näin edetään kunnes taulukossa on enää r kappaletta todennäköisyyksiä (kuvassa 7 viimeinen sarake).

4. Seuraavassa vaiheessa joka symbolille määrätään koodisana. Taulukon viimeisessä sarakkeessa (kuva 7) olevat r-kappaletta koodisanoja saa jokainen oman symbolin D-aakkostosta. Kun taulukkoa käydään läpi käänteisessä järjestyksessä viimeisestä vaiheesta ensimmäiseen, niin jokainen todennäköisyyksien yhteenlasku lisää koodisanan pituutta yhdellä, niin että ylin todennäköisyyksistä merkitään ensimmäisellä aakkosella, toiseksi ylin toisella jne. Kuvassa 7 näkyvät määritellyt koodisanat omissa sarakkeissaan.

5. Lopuksi jätetään huomiotta 'dummy'-koodisanat.

Huffman koodin optimaalisuus, joka todistettiin kappaleessa 4.1.1, on todistettavissa myös ei-binaarisessa tapauksessa eli kun käytetään koodiaakkostoa, jossa on enemmän kuin kaksi merkkiä.

Kun koodiaakkostossa on D symbolia, niin lemma 4.4 kuuluu seuraavasti:

Lemma 4.6. *Olkoon koodiaakkostossa D symbolia. Mille tahansa jakaumalle on olemassa optimaalinen välitön koodi, joka toteuttaa seuraavat ehdot:*

1. *Jos $p_j > p_k$, niin silloin $l_j \leq l_k$.*
2. *D pisintä koodisanaa ovat yhtä pitkät.*
3. *D pisintä koodisanaa eroavat toisistaan vain viimeisen bitin osalta ja nämä koodisanat vastaavat D kappaletta vähiten todennäköisintä lähdesymbolia.*

Lemman 4.6 todistus etenee samalla lailla kuin Lemman 4.4 todistus. Kohdan 1 todistus on täsmälleen sama kuten kappaleessa 4.1.1. Kohtien 2 ja 3 todistuksessa on otettava huomioon, että myös 'dummy'-koodisanat tulee olla mukana todistuksessa. Tällöin D pisintä koodisanaa ovat yhtä pitkät. Jos pisimpiä koodisanoja on vähemmän kuin D kappaletta, niin tällöin niistä jokaisesta voidaan ottaa yksi bitti pois ja päästään tilanteeseen, jossa pisimpiä koodisanoja on D kappaletta.

Jos 'dummy'-koodisanat eivät ole mukana todistuksessa, niin todistus hankaloituu, koska silloin pisimpiä koodisanoja voi olla vähintään kaksi tai enintään D kappaletta. Kohtien 2 ja 3 todistus on siis näiltä osin eri kuin Lemman 4.4 todistus. Muilta osin todistus seuraa Lemman 4.4 todistusta.

4.1.3 Käytännön esimerkkejä Huffman-koodin käytöstä

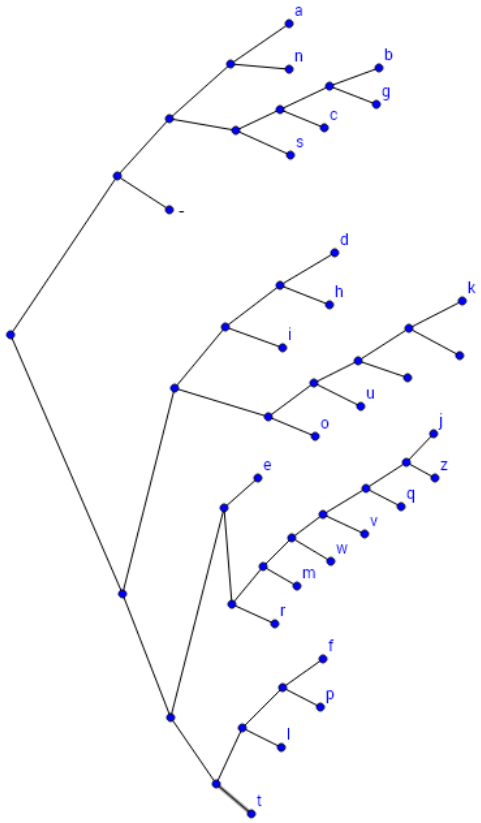
Huffman-koodilla on useita käytännön sovelluksia ja sitä käytetään paljon esim. tekstitiedon koodamiseen, koska Huffman koodi on häviötön tiedonpakkausmenetelmä. Tämä tarkoittaa sitä, että Huffman-koodia käyttämällä saavutetaan pienempi tiedostokoko, mutta tietoa ei häviä. Osa kompressioeli pakkausmenetelmistä hävittää osan tiedosta, kuitenkin niin, että tiedon häviämisestä ei ole haittaa pakatulle tiedolle. Esimerkiksi kuvien ja videoiden pakkaamisessa käytetään usein tietoa hävittäviä pakkausmenetelmiä. Tällöin hävinnyt tieto lisää kuvan tai videon epätarkkuutta, mutta niin vähän, ettei sitä voi silmin havaita.

Yleisin käyttökohde Huffman-koodille on jonkin tietyn viestin pakkaaminen. Tällöin kirjainten esiintymislukumääristä on laskettu uudet todennäköisyydet kullekin kirjaimelle ja Huffman-koodin periaatteen mukaan useimmin esiintyvä kirjain saa lyhimmän koodin. Tällä tavoin viestiä voidaan tiivistää ja näin tarvitaan vähemmän bittejä kuin esimerkiksi, jos kirjaimet on koodattu ASCII-merkeiksi.

Seuraavan taulukon toisessa sarakkeessa on kullekin englannin kielessä esiintyvälle kirjaimelle laskettu todennäköisyys kirjaimen keskimääräisten esiintymiskertojen lukumäärän mukaan.

a_i	p_i	l_i	$c(a_i)$	a_i	p_i	l_i	$c(a_i)$
a	0,0575	4	0000	o	0,0689	4	1011
b	0,0128	6	001000	p	0,0192	6	111001
c	0,0263	5	00101	q	0,0008	9	110100001
d	0,0285	5	10000	r	0,0508	5	11011
e	0,0913	4	1100	s	0,0567	4	0011
f	0,0173	6	111000	t	0,0706	4	1111
g	0,0133	6	001001	u	0,0334	5	10101
h	0,0313	5	10001	v	0,0069	8	11010001
i	0,0599	4	1001	w	0,0119	7	1101001
j	0,0006	10	1101000000	x	0,0073	7	1010001
k	0,0084	7	1010000	y	0,0164	6	101001
l	0,0335	5	11101	z	0,0007	10	1101000001
m	0,0235	6	110101	-	0,1928	2	01
n	0,0596	4	0001				

Taulukon todennäköisyyksien perusteella on rakennettu Huffman-puu (kuva 8) ja määritelty uudet koodisanat jokaiselle kirjaimelle (taulukossa $c(a_i)$-sarake). Kuvan 8 puussa ylempi haara vastaa bittiä 0 ja alempi haara bittiä 1. Esimerkki on peräisin lähteestä [5].

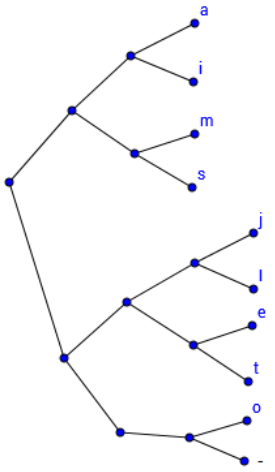


Kuva 8: Huffman puu englannin kielen aakkosille.

Toinen esimerkki on harjoitustyöstä, joka on käytössä sähkö- ja tietotekniikan koulutusohjelmissa kurssilla 'Algoritmit ja tietorakenteet'. Harjoitustyössä opiskelijan tulee koodata oma nimensä käyttäen binaarista Huffman-koodia enintään nelibittiseksi binaarikoodiksi. Ensin opiskelijan on luotava taulukko, jossa käy ilmi hänen nimessään esiintyvät kirjaimet ja kirjaimen esiintymisten lukumäärä. Tämän perusteella jokaiselle kirjaimelle lasketaan esiintymistodennäköisyys. Tämän jälkeen opiskelija generoi Huffman-koodin, jossa eniten esiintyvä kirjain saa lyhimmän koodisanan ja vähiten esiintyvä kirjain pisimmän koodisanan. Kirjain - koodisana vastaavuudet kirjoitetaan kooditauluun, joka on käytännössä tiedosto. Opiskelija koodaa Python-kielellä ohjelman, jossa syötteenä on opiskelijan nimi ja nimen jokaiselle kirjaimelle etsitään binaarinen vastine kooditaulusta. Tämän jälkeen ohjelma tulostaa binaarivastineen, joka on hänen nimensä koodattuna Huffman-koodilla. Ohjelman tulee myös dekoodata Huffman-koodattu viesti ja tähän

käytetään samaa kooditaulua. Koska Huffman-koodi on välitön eli mikään koodi ei ole toisen etuliite, niin binaarimuodossa oleva Huffman-koodattu viesti voidaan yksiselitteisesti dekoodata kirjaimiksi.

Esimerkkinä alla Huffman-puu (kuva 9) sekä taulukko, jossa Maija-Liisa Metso nimestä on laskettu kirjainten esiintymistodennäköisyys ja sen perusteella piirretty Huffman puu ja päätelty jokaiselle kirjaimelle Huffman koodi.



Kuva 9: Huffman-puu 'Maija-Liisa Metso' -nimessä esiintyville kirjaimille kirjainten esiintymistodennäköisyyden perusteella .

Seuraavan taulukon ensimmäisessä sarakkeessa on lueteltu viestissä esiintyvät kirjaimet ja toisessa sarakkeessa on kirjainten esiintymiskertojen lukumäärä. Kolmanteen sarakkeeseen on laskettu kirjainten esiintymistodennäköisyydet, kun kirjaimia on yhteensä viestissä 16 kappaletta. Neljännessä sarakkeessa on kuvan 9 puusta määritelty Huffman-koodi kullekin kirjaimelle. Viimeisessä sarakkeessa on $p_i l_i$ eli kirjaimen todennäköisyys kerrottuna koodisanan pituudella.

kirjain	esiintymisten lukumäärä	todennäköisyys	Huffman-koodi	$p_i l_i$
a	3	0,1875	000	0,5625
i	3	0,1875	001	0,5625
m	2	0,125	010	0,375
s	2	0,125	011	0,375
j	1	0,0625	1000	0,25
l	1	0,0625	1001	0,25
e	1	0,0625	1010	0,25
t	1	0,0625	1011	0,25
o	1	0,0625	1100	0,25
-	1	0,0625	1101	0,25

Yllä olevan taulukon viimeisestä sarakkeesta voidaan laskea keskimääräinen koodinpituus:

$$L = \sum_{i=1}^{10} p_i l_i = 2 * 0,5625 + 2 * 0,375 + 6 * 0,25 = 3,375 \tag{10}$$

Jos tätä keskimääräistä koodinpituutta verrataan Shannonin määrittelemään rajaan koodinpituudelle (kaava 6), joka on

$$\begin{aligned} L \geq \frac{H(X)}{\log D} &= \frac{\sum_{i=1}^{10} p_i \log \frac{1}{p_i}}{\log 2} = \sum_{i=1}^{10} p_i \frac{\ln \frac{1}{p_i}}{\ln 2} \\ &= 2 * 0,1875 \frac{\ln \frac{1}{0,1875}}{\ln 2} + 2 * 0,125 \frac{\ln \frac{1}{0,125}}{\ln 2} + 6 * 0,0625 \frac{\ln \frac{1}{0,0625}}{\ln 2} \\ &= 3,1556 \end{aligned}$$

niin tästä voidaan päätellä, että tämä Huffman-koodi ei saavuta Shannonin rajaa, mutta toteuttaa kuitenkin kaavan (6) epäyhtälön.

Nyt kun yllä olevaa kooditaulua käytetään koodaamaan nimi Maija-Liisa Metso (välilyönti jätetään huomiotta), niin saadaan binaarikoodi:

01000000110000001101100100100101100001010101011011110 0

Tämän binaarikoodin pituus on 56 bittiä ja sillä on koodattu 16 kirjainta, joten kirjaimen keskimääräiseksi pituudeksi saadaan $\frac{54}{16} = 3,375$, joka on siis täsmälleen keskimääräinen koodinpituus, joka määriteltiin kaavassa (10).

Koska Huffman-koodi on välitön, niin purkaminen on yksiselitteistä. Koska mikään koodi ei ole toisen etuliite, niin 0-alkuiset koodit ovat kolme bittisiä ja ykkösellä alkavat koodit neljän bitin mittaisia. Tämän perusteella

voidaan ohjelmoida dekoodaus. Dekoodauksessa huomataan, että sarja alkaa nollalla, joten ensimmäistä kirjainta vastaa siis kolme bittiä. Tällöin saadaan 010 bittijonon vastineeksi kirjain m. Seuraava bitti on myös 0, joten kolme seuraavaa bittiä 000 vastaavat kirjainta a. Myös seuraava bitti on nolla, joten otetaan kolme bittiä ja saadaan kirjain i. Tämän jälkeinen bitti on yksi, joten kyseessä on neljän bitin sarja, joten 1000 vastaa kirjainta j. Dekoodausta jatketaan tällä tavoin ja lopulta, kun koko bittijono on dekoodattu saadaan selville tätä Huffman-koodia vastaava teksti eli maija-liisametso.

5 Yhteenveto

Tässä työssä käytiin läpi yksi informaatioteorian peruslauseista eli Shannonin ensimmäinen lause ja sen matemaattinen todistaminen. Todistamisen pohjaksi esiteltiin Kraftin epäyhtälö ja McMillanin epäyhtälö. Työn soveltavassa osassa esiteltiin eräs optimaalinen koodaustapa eli Huffman-koodaus ja sen optimaalisuus todistettiin. Lopussa esiteltiin muutama käytännön esimerkki Huffman-koodauksen soveltamisesta.

Lähdeluettelo

[1] T.M. Cover & J.A. Thomas: *Elements of Information Theory.* John Wiley & Sons, Inc., New York, 1991.

[2] R.V.L. Hartley: *Transmission of Information.* Bell Sys. Tech. Journal, 7, 1928.

[3] D.A. Huffman: *A method for the construction of minimum redundancy codes.* Proc. IRE, 40: 1098-1101, 1952.

[4] F.M. Ingels: *Information and Coding Theory.* Intext Educational Publishers, Scranton, 1971.

[5] David J.C. MacKay: *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, Cambridge, 2003.

[6] B. McMillan: *Two inequalities implied by unique decipherability.* IEE Trans. Inform. Theory, IT-2, 1956.

[7] H. Nyquist: *Certain Factors Affecting Telegraph Speed.* Bell Sys. Tech. Journal, 3, 1924.

[8] C.E. Shannon: *A mathematical theory of communication.* Bell Sys. Tech. Journal, 27: 379-423, 623-656, 1948.

[9] N. Wiener: *Cybernetics.* New York: Wiley, 1948.