Muodolliset koneet ja laskettavuus

Juuso Valli

3. 9. 2017

Tiivistelmä

Tässä tutkielmassa käsitellään laskettavuuden peruskäsitteitä, tutustutaan rekisteri- ja Turingin koneisiin sekä osoitetaan näiden olevan laskentakyvyltään identtisiä.

Sisältö

1 Algoritmit ja muodolliset koneet

Algoritmilla tarkoitetaan äärellistä luetteloa ohjeita tai komentoja, joita noudattamalla voidaan ratkaista jokin tietty ongelma. Ratkaisulla tarkoitetaan yleensä vastausta kyllä-ei kysymykseen tai jonkinlaista numeerista tulosta annetun syötteen perusteella. Eukleideen algoritmi kahden luvun suurimman yhteisen tekijän selvittämiseen on eräs vanhimmista tunnetuista algoritmeista. Algoritmit ovat historiallisesti olleet merkittävä osa käytännön matematiikkaa; tavallisimpien algoritmien tuntemus on mahdollistanut monimutkaisten ongelmien matemaattisen ratkaisemisen ilman syvällistä ymmärrystä. Mikäli oppilas pystyy oppimaan algoritmin, hän pystyy yksinkertaisia ohjeita noudattamalla ratkaisemaan kokonaisen luokan ongelmia. Tässä merkityksessä algoritmit eivät ole menettäneet arvoaan, matematiikan alkeisopetus nojaa edelleen vahvasti yksinkertaisten algoritmien opetteluun. Algoritmien toinen vahvuus seuraa siitä tosiasta, että algoritmin suorittajan ei tarvitse olla ajatteleva ihminen. Sopivalla tavalla suunniteltu mekaaninen tai sähköinen kone pystyy myös noudattamaan algoritmin ohjeita. Tämä tarjoaa ihmissuorittajaan verrattuna lukuisia etuja. Koneen suorittamat algoritmit ovat vähemmän alttiita virheille ja useimmissa tapauksissa huomattavasti ihmistä nopeampia. Toisaalta ihmiset pystyvät käsittelemään paljon vapaamuotoisempia algoritmeja. Koneet yleensä pystyvät käsittelemään vain hyvin rajallista komentokantaa.

Algoritmin komentokannan rajoittamisesta on etua. Täysin vapaamuotoisia algoritmeja on vaikea tutkia; algoritmin oikeellisuus on vaikea todistaa täsmällisesti, algoritmin tulos voi vaihdella komentojen tulkinnan mukaan, ja kahden algoritmin vertailu on lähes mahdotonta. Muodollisen määrittelyn kautta päästään eroon monista epävarmuuksista. Komennot ovat yksiselitteisiä, ja kahden algoritmin suoritusaikaa voidaan verrata laskemalla suoritettujen komentojen lukumäärä. Algoritmin oikeellisuus on usein helpompi todistaa ja joskus voidaan osoittaa, että ratkaisualgoritmia ei ole olemassa tämän komentokannan puitteissa.

Komentokannan käsite on vahvasti sidoksissa siihen, millainen algoritmin suorittajan ajatellaan olevan. Vapaamuotoisen algoritmin suorittajan ajatellaan olevan ihminen. Muissa tapauksissa algoritmin suorittaja on jollakin tavalla määritelty formaalinen kone. Formaaleilla koneilla ei ole kaikenkattavaa määritelmää, vaan jokainen määritellään eri tavoin. Formaalista koneesta käytetään tässä merkintää $M : S \to R$, missä S on syöte ja R on algoritmin tulos. On tärkeä huomata että tässä M ei ole funktio $S \to R$, sillä formaalinen kone ei välttämättä koskaan lopeta suoritusta annetulla syötteellä. Kone M voidaan täydentää funktioksi $S \to R \bigcup \{\dot\infty\}$, missä $\dot\infty \notin R$ tarkoittaa, että M ei koskaan lopeta suoritusta.

Tietyllä tavalla määriteltyjen formaalien koneiden avaruudesta käytetään tässä merkintää $\hat{M}$.

2 Laskettavuus

Eräs algoritmitutkimuksen peruskysymys on selvittää, mitkä ongelmat ovat ratkaistavissa algoritmisesti. Ongelmalla tarkoitetaan tässä funktiota $P : S \to R$, jonka arvoja halutaan laskea jollakin algoritmilla. Koneen M sanotaan ratkaisevan ongelman P mikäli $\forall s \in S : M(s) = P(s)$. Tyypillisiä syöteavaruuksia S ovat äärellisien merkkijononen avaruus Σ^* tai $\mathbb{N}^n$. Tyypilliä tulosavaruuksia R ovat {'kyllä', 'ei'}, $\mathbb{N}$ tai Σ^*

Ongelman P sanotaan olevan ratkaistavissa konetyypillä $\hat{M}$, mikäli on olemassa sellainen $M \in \hat{M}$ joka ratkaisee P:n. Mikäli ongelma P on ratkaistavissa jollakin konetyypillä, kutsutaan sitä laskettavissa olevaksi. On tärkeää huomata, että ongelma on ratkaistavissa mikäli voidaan osoittaa, että sille on olemassa ratkaisu, vaikka ratkaisua ei suoraan tunnettaisikaan. Kaikille ongelmille ei luonnollisesti löydy ratkaisualgoritmia.

3 Muodolliset kielet

Muodolliset kielet liittyvät läheisesti sellaisiin muodollisiin koneisiin, joiden syöte on merkkijono ja jotka joko hyväksyvät tai hylkäävät saamansa syötteen. Tällaisia ovat esimerkiksi äärelliset automaatit ja Turingin koneet. Olkoon Σ äärellinen aakkosto. Σ:n alkioiden äärellisiä jonoja kutsuun sanoiksi. Tyhjästä sanastaa käytetään merkintää ϵ. Kaikkien sanojen joukosta käytetään termiä Σ^*. Muodollinen kieli L on Σ^*:n osajoukko. L:n alkoita kutsutaan sanoiksi. L:n ajatellaan usein määräytyvän joistakin säännöistä, mutta tämä ei ole välttämätöntä.

Olkoon $M : \Sigma^* \to \{'\text{kyllä}','\text{ei}'\}$ muodollinen kone ja olkoon $w \in L$. $M(w)$ merkitsee koneen M antamaa tulosta syötteellä w. Kone M tunnistaa kielen $L(M) = \{w \in \Sigma^* | M(w) =' \text{kyllä}'\}$.

4 Automaatit

Automaatit ovat yksinkertaisia muodollisia koneita. Ne joko hyväksyvät tai hylkäävät annetun äärillisen merkkijonon. Muodollisesti määriteltynä automaatti M on viisikko $(Q, \Sigma, \delta, q_0, F)$, jonka osat ovat

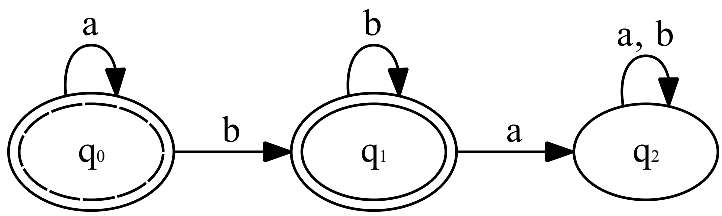
- äärellinen tilajoukko Q

- äärellinen aakkosto Σ
- siirtymäfunktio $\delta : Q \times \Sigma \to Q$
- alkutila $q_0 \in Q$
- hyväksyvien tilojen joukko $F \subset Q$.

Automaatti on aina jossakin tilassa $q \in Q$, alkutilanteessa $q = q_0$. Se lukee syötteestä yhden symbolin kerrallaan, olkoon tämä symboli $\sigma \in \Sigma$. Koneen nykyisen tilan ja luetun merkin perusteella se siirtyy tilaan $\delta(q, \sigma)$, ja siirtyy lukemaan seuraavaa symbolia. Kun koko syöte on käyty läpi tarkistetaan, onko automaatin sen hetkinen tila joukossa F. Jos näin on, palautetaan vastaukseksi 'kyllä', muutoin 'ei'.

Tällä tavoin määritelty muodollinen kone on monilta ominaisuuksiltaan käyttökelpoinen; sen suoritus kestää aina lineaarisen ajan, ja sen suorittamiseen tarvittavan muistin määrä on vakio.

Esimerkki 1. $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_0, q_1\})$ jossa siirtymäfunktio δ kuvataan seuraavalla graafilla:



Kuva 1: Automaatin tilakaavio

Siirtymäfunktio voidaan myös kuvata taulukkona:

	q_0	q_1	q_2
a	q_0	q_2	q_2
b	q_1	q_1	q_2

Tämä kone hyväksyy vain merkkijonot, jotka ovat muotoa $a^n b^m$, $n, m \in \mathbb{N}$.

Jos syötteeksi annetaan esimerkiksi merkkijono aaabb, suoritus etenisi seuraavasti:

Tila	Merkki	Jäljellä oleva syöte
q_0	a	aabb
q_0	a	abb
q_0	a	bb
q_0	b	b
q_1	b	
q_1		

Koska $q_1 \in F$, kone antaa tulokseksi 'kyllä'.
Toisaalta jos syötteenä olisi aba, suoritus etenisikin seuraavasti:

Tila	Merkki	Jäljellä oleva syöte
q_0	a	ba
q_0	b	a
q_1	a	
q_2		

Koska $q_2 \notin F$, kone antaisi tulokseksi 'ei'.

Lause 2. *Olkoon M esimerkissä 3 määritelty automaatti.* $L(M) = \{a^n b^m | n, m \in \mathbb{N}\}$.

Todistus. Olkoon sana $w \in \Sigma^*$. Jos $w \in L(M)$, niin automaatti selvästi palauttaa tuloksen 'kyllä'. Jos $w \notin L(M)$, niin sen täytyy sisältää ainakin yksi *ba*-pari. Tämä riittää siirtämään automaatin mistä hyvänsä tilasta tilaan q_2. Mikään merkki ei siirrä automaattia tilasta q_2 mihinkään muuhun tilaan, joten suorituksen päätyttyä kone on edelleen tilassa q_2, ja palauttaa siis tulokseksi 'ei'. $\square$

Äärellisillä automaateilla on kuitenkin monia rajoitteita. On runsaasti esimerkkejä ongelmista, jotka intuitiivisesti olisivat ratkaistavissa jollakin algoritmilla, mutta joita ei voi laskea äärellisen automaatin avulla. Tällaisia ongelmia ovat esimerkiksi 'Onko annettu suljelauseke oikein muodostettu?' tai 'Onko merkkijono muotoa $a^n b^n$?'. Onkin selvää, että äärelliset automaatit eivät ole riittävän yleisiä toteuttaakseen kaikkia algoritmeja.

Lause 3. *Ei ole olemassa äärellistä automaattia M, joka tunnistaa kielen* $L = \{a^n b^n | n \geq 0\}$.

Todistus. Olkoon M äärellinen automaatti, jolla on n tilaa. Mikä tahansa merkkijono, jonka pituus on vähintään n aiheuttaa sen, että jokin koneen tiloista toistetaan. Merkitään tätä tilaa S. Siirtymä tämän tilan ensimmäisestä esiintymästä toiseen vastaa jotakin käsitellyn merkkijonon osaa. Olkoon tämä osa y. Jos kone on tilassa S ja se saa syötteekseen y:n, päätyy se taas tilaan S. Tästä seuraa, että jos automaatti hyväksyy sanan $w = xyz$, täytyy sen hyväksyä sana $xy^iz, i \geq 0$.

Sovelletaan tätä nyt kielen L tapaukseen. Olkoon $w = a^m b^m, m > n$. Tällöin on olemassa sellainen jako $i+j+k = m, j > 0$, että kone M tunnistaa sanat $a^i a^{j*p} a^k b^m, p \in \mathbb{N}$. Näistä kuitenkin vain yksi kuuluu kieleen L, joten $L(M) \neq L$. □

5 Rekisterikoneet

Shepherdsonin ja Sturgisin rekisterikoneet ovat rakenteellisesti hyvin yksinkertaisia. Koneella on R rekisteriä, joista kukin sisältää yhden luonnollisen luvun. Suorituksen alussa rekistereissä ovat luonnolliset luvut $x_1, x_2, ..., x_m, m \leq R$, muualla nolla. Onnistuneen laskun lopussa tulos y on rekisterissä 1 ja muut rekisterit ovat nollia. Kone laskee siis funktion $f : \mathbb{N}^m \rightarrow \mathbb{N}$ jossa $f(x_1, ..., x_m) = y$.

Rekisterin n arvoon viitataan merkinnällä $\bar{n}$. Suoritettava algoritmi esitetään numeroituna luettelona, joka koostuu hyvin yksinkertaisista komennoista. Merkitään näitä komentoja $\hat{1}$, $\hat{2}$, $\ldots$, $\hat{M}$.

Rekisterikone tunnistaa kolme erilaista komentoa: (n, j), (n, j, k) sekä lopetuskomennon $\hat{M}$. Kahden ensimmäisen komennon merkitykset ovat:

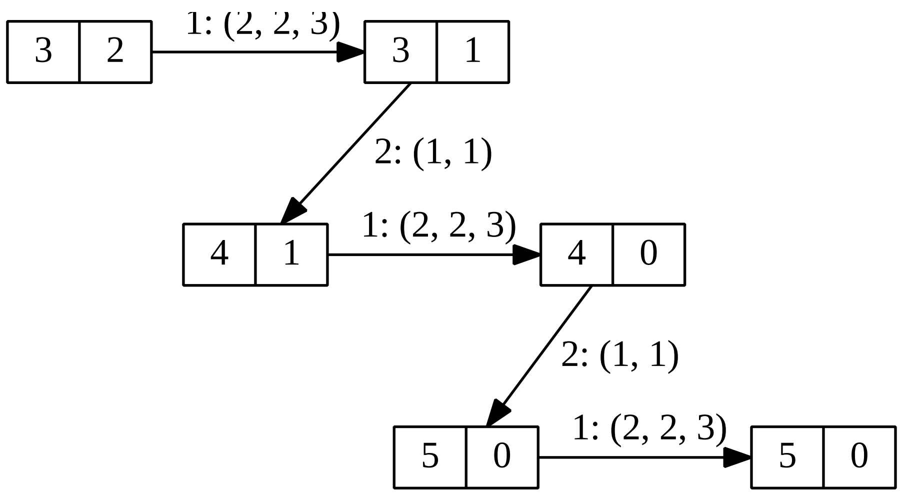
(n, j)	Lisää 1 rekisteriin n ja siirry komentoon j.
(n, j, k)	Jos $\bar{n} > 0$, vähennä 1 rekisteristä n ja siirry komentoon j; muulloin siirry komentoon k.

Näillä sinänsä yksinkertaisilla säännöillä on mahdollista suorittaa monimutkaisiakin laskutoimituksia.

Esimerkki 4. Määritellään rekisterikone joka suorittaa yhteenlaskun.

$R = 2, M = 3$
$\hat{1}$ $(2, 2, 3)$
$\hat{2}$ $(1, 1)$
$\hat{3}$ Seis

Olkoon rekisterien arvot alkutilanteessa $(3, 2)$. Ohjelman suoritus etenisi seuraavalla tavalla:



Kuva 2: Esimerkki rekisterikoneen toiminnasta

6 Turingin koneet

Turingin kone on ensimmäinen tunnettu muodollinen kone. Alan Turing esitteli sen vuonna 1937, kun käytti sitä apuvälineenä ratkaistakseen David Hilbertin vuonna 1928 esittämän päätösongelman. Turingin koneet poikkeavat rakenteellisesti rekisterikoneista huomattavasti ja muistuttavat enemmänkin äärellisiä tilakoneita. Muodollisesti määriteltynä Turingin koneella on seitsemän komponenttia $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, jossa

- äärellinen tilajoukko Q
- nauha-aakkosto Γ
- syöteaakkosto $\Sigma \subset \Gamma$
- siirtymäfunktio $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
- alkutila q_0
- 'tyhjä' nauhasymboli $B \in \Gamma - \Sigma$
- hyväksyvien tilojen joukko $F \subseteq Q$.

Turingin koneiden ajatellaan käsittelevän kaksisuuntaisesti ääretöntä nauhaa, jonka jollekin äärelliselle alueelle on kirjoitettu symboleja. Muu nauha sisältää vain symbolia B ('blank' eli tyhjä). Turingin koneet toimivat kuten äärelliset automaatit lukuun ottamatta kahta merkittävää eroa. Ne voivat siirtyä nauhaa sekä eteen- että taaksepäin, ja ne voivat muuttaa nauhan sisältöä.

Alkutilanteessa kone aloittaa syötteen vasemmasta reunasta, tilassa $q_0 \in Q$. Kone lukee yhden merkin ja päättää sen perusteella mihin tilaan siirtyä, minkä merkin kirjoittaa nauhalle luetun merkin tilalle ja siirtyäkö oikealle vai vasemmalle. Suoritus päättyy vasta, kun siirtymäfunktio δ palauttaa arvon $\emptyset$. Sen jälkeen verrataan senhetkistä tilaa joukkoon F kuten äärellisten automaattien tapauksessa.

Turingin konetta voi käyttää kielten tunnistamiseen kuten esimerkissä 5, mutta sillä voi myös laskea funktion. Syöte $x_1, ..., x_m$ koodataan nauhalle vaikka muodossa $B0^{x_1}B0^{x_2}B...B0^{x_m}B$. Onnistuneen laskun lopussa tulos y on nauhalla muodossa $B0^yB$. Turinginkin kone laskee siis tämän funktion $f : \mathbb{N}^m \to \mathbb{N}$, jossa $f(x_1, ..., x_m) = y$.

Esimerkki 5. Olkoon M Turingin kone siten, että
$M = (\{q_0, q_a, q_b, q_r, q_e\}, \{a, b\}, \{a, b, B\}, \delta, q_0, B, \{q_0\})$ missä δ toimii seuraavan taulukon mukaisesti:

	a	b	B
q_0	(q_a, B, R)	(q_e, b, R)	$\emptyset$
q_a	(q_a, a, R)	(q_a, b, R)	(q_b, B, L)
q_b	(q_e, a, R)	(q_r, B, L)	(q_e, B, R)
q_r	(q_r, a, L)	(q_r, b, L)	(q_0, B, R)
q_e	$\emptyset$	$\emptyset$	$\emptyset$

Tällä tavoin määritelty Turingin kone tunnistaa kielen $\{a^n b^n | n \geq 0\}$, jota äärelliset automaatit eivät pysty tunnistamaan.

7 Funktioiden esittäminen muodollisella koneella

Useimmissa tapauksissa ei riitä, että algoritmi tuottaa kyllä/ei -vastauksen, vaan algoritmin toivotaan tuottavan esimerkiksi laskutoimituksen tuloksen. Näissä tapauksissa ei ole tärkeää se, missä tilassa kone on lopettaessaan suorituksen, vaan se, mitä rekistereissä tai nauhalla on. Lopetustilan tarkastelu ei tällaisissa tapauksissa ole tarpeellista, vaan voimme pitää kaikkia lopetustiloja hyväksyttävinä.

Se, miten algoritmi palauttaa vastauksen, on sopimuskysymys. Turingin koneen tapauksessa vastaus luetaan nauhalta suorituksen päätyttyä, ja rekisterikoneen tapauksessa tulos tallennetaan yleensä rekisteriin yksi.

8 Rekisterikoneiden ja Turingin koneiden välinen yhteys

Vaikka Turingin koneet ja rekisterikoneet ovat rakenteellisesti hyvin erilaisia, ovat ne kuitenkin osoittautuneet laskukyvyltään ekvivalenteiksi. Jos ongelma on ratkaistavissa rekisterikoneella, se on myös ratkaistavissa Turingin koneella ja päinvastoin. Se, että näinkin erilaiset koneet ovat yhtä ilmaisuvoimaisia, viittaa siihen, että olemme lähestymässä käyttökelpoista tapaa määritellä laskettavuuden käsite. Laskettavuus samaistetaankin usein laskettavuuteen Turingin koneilla.

Osoitetaan ensin, että annetun Turingin koneen M toimintaa voidaan simuloida sopivasti rakennetulla rekisterikoneella. Jos tämä pitää paikkansa, mikä tahansa Turingin koneella ratkaistavissa oleva ongelma voidaan ratkaista myös rekisterikoneella tarvittaessa simuloimalla ko. Turingin koneen toimintaa. Tämän jälkeen osoitetaan, että sama pätee myös toiseen suuntaan.

Koska Turingin koneet ja rekisterikoneet ovat rakenteellisesti erilaisia, otetaan käyttöön hieman aputermistöä.

Määritelmä 6. Valitaan bijektio $f_\Gamma : \Gamma \to \{0, 1, 2, \ldots, |\Gamma| - 1\}$ siten että $f(B) = 0$. Merkin $\gamma \in \Gamma$ *kokonaislukuesitys* on $f_\Gamma(\gamma)$.

Kuvauksen f valinnalla ei muuten ole väliä, kunhan samassa yhteydessä käytetään aina samaa kuvausta. Mikäli kuvausta ei erikseen määritellä, oletetaan yleensä sen kuvaavan aakkoston positiivisiin kokonaislukuihin samassa järjestyksessä missä merkit lueteltiin.

Määritelmä 7. Valitaan bijektio $f_Q : Q \to \{0, 1, 2, \ldots, |Q|-1\}$. Tilan $q \in Q$ *kokonaislukuesitys* on $f_Q(q)$.

Määritelmä 8. *Puolinauha* $T_{n,R|L}$ on Turingin koneen nauhan puolikas jostakin indeksistä n alkaen joko vasemmalle tai oikealle. Indeksillä n tarkoitetaan ensimmäistä indeksiä mikä sisältyy puolinauhaan. $T_{3,R}$ tarkoittaa puolinauhaa, joka sisältää kaikki nauhan merkit indeksistä kolme alkaen oikealle. $T_{n,R|L}(i)$ tarkoittaa puolinauhan i:ttä merkkiä leikkauspäästä alkaen.

Määritelmä 9. Puolinauhan $T_{n,R|L}$ *kokonaislukuesitys* on

$$f_T(T_{n,R|L}) = \Sigma_{i=0}^{\infty} f_\Gamma(T_{n,R|L}(i)) \times |\Gamma|^i$$

Esimerkki 10. Olkoon $\Gamma = \{B, a, b, c\}$ ja $T_{0,R} = abac$. T:n kokonaislukuesitys olisi tässä tapauksessa $f_\Gamma(a) + f_\Gamma(b) \times 4 + f_\Gamma(a) \times 4^2 + f_\Gamma(c) \times 4^3 = 1 + 2 \times 4 + 1 \times 16 + 3 \times 64 = 217$.

Seuraavat identiteetit seuraavat suoraan puolinauhan kokonaislukuesityksen määritelmästä:

$$
\begin{aligned}
f_T(T_{i,R}) =& f_T(T_{i+1,R}) \times |\Gamma| + f_\Gamma(T(i)) \\
f_T(T_{i,R}) =& \left\lfloor \frac{f_T(T_{i-1,R})}{|\Gamma|} \right\rfloor \\
f_T(T_{i,L}) =& f(T_{i-1,L}) \times |\Gamma| + f_\Gamma(T(i)) \\
f_T(T_{i,L}) =& \left\lfloor \frac{f_T(T_{i+1,L})}{|\Gamma|} \right\rfloor
\end{aligned}
$$

Lause 11. *Olkoon P ongelma joka on ratkaistavissa Turingin koneella. Ongelma P on ratkaistavissa rekisterikoneella.*

Todistus. Olkoon M Turingin kone joka ratkaisee ongelman . Lauseen todistamiseksi rakennetaan sellainen rekisterikone M' joka simuloi M:n toimintaa.

M:n suoritusaikainen kokonaistila koostuu nauhan sisällöstä, lukupään sijainnista ja koneen tilasta. Numeroidaan ensin M:n nauha-aakkosto määritelmän 6 mukaan, ja tilajoukko Q määritelmän 7 mukaan. Nauhan sisältö ja lukupään sijainti voidaan yhdessä mallintaa kolmella rekisterillä, joista yksi sisältää lukupään vasemmalla puolella olevan puolinauhan kokonaislukuesityksenä, toinen sisältää lukupään kohdalla olevan merkin kokonaislukuesityksen ja kolmas sisältää lukupään oikealla puolella olevan puolinauhan kokonaislukuesityksen. Koneen tila esitetään neljäntenä rekisterinä joka sisältää tilan kokonaislukuesityksen. Lisäksi tarvitaan rajallinen määrä laskurekistereitä kerto- ja jakolaskuja varten.

Koneen aloitustila on

Rekisteri	Tila
1	$f_T(T_{-1,L}) = 0$
2	$f_\Gamma(T(0))$
3	$f_T(T_{1,R})$
4	$f_Q(q_0)$

Muissa rekistereissä aloitusarvo on 0.

Komentosarjan alussa on $|\Gamma|$ kappaletta muotoa $\hat{\imath}(2, i+1, p_i)$ olevia komentoja, jotka valitsevat luettua merkkiä vastaavan komentosarjan. Koska luettuja merkkejä on vain $|\Gamma|$ kappaletta, tiedetään että jokin sarjan komennoista johtaa komentoon p_i. Jokainen p_i-komento aloittaa vastaavanlaisen $|Q|$ komentoa pitkän sarjan muotoa $\hat{\jmath}(4, j+1, q_{i,k})$ olevia komentoja. Jokainen

$q_{i,j}$ komento vastaa siis Turingin koneen merkki-tila-yhdistelmää. Nämä komennot joko lopettavat ohjelman suorituksen tai aloittavat sarjan komentoja jotka asettavat uuden merkin rekisteriin 2 ja uuden tilan rekisteriin 4. Tällä tavoin olemme simuloineet M:n tilatransitiofunktion toiminnan.

Viimeisenä alkuperäinen kone M siirtyisi joko vasemmalle tai oikealle. Nämä operaatiot ovat lähes identtisiä, joten tarkastellaan tässä vain oikealle siirtymistä.

Mikäli M:n lukupää on asemassa n, M':n kolmessa ensimmäisessä rekisterissä ovat arvot

Rekisteri	Tila
1	$f_T(T_{n-1,L})$
2	$f_\Gamma(T(n))$
3	$f_T(T_{n+1,R})$

M:n siirtyessä oikealle M':n rekisterien tulisi siirtyä tilaan

Rekisteri	Tila
1	$f_T(T_{n,L}) = f_T(T_{n-1,L}) \times \lvert\Gamma\rvert + f_\Gamma(T(n))$
2	$f_\Gamma T(n+1)$
3	$f_T(T_{n+2,R}) = \left\lfloor \dfrac{f_T(T_{n+1,R})}{\lvert\Gamma\rvert} \right\rfloor$

Rekisterien tilan siirtymä voidaan toteuttaa kertomalla 1:n sisältö $|\Gamma|$:lla ja lisäämällä siihen rekisterin 2 sisältö. Tämän jälkeen jaetaan rekisterin 3 sisältö $|\Gamma|$:lla ja asetetaan jakojäännös rekisteriin 2, ja palataan ohjelman alkuun. Tällä tavoin olemme simuloineet M:n nauhan siirtämisen.

Suorituksen päätyttyä M:n lopetustila saadaan syöttämällä rekisterin neljä arvo kuvauksen f_Q käänteisfunktioon $f_Q^{-1}(\bar{4})$. Mikäli näin saatu tila sisältyy M:n hyväksyvien tilojen joukkoon F M' palauttaa arvon 'kyllä', muulloin 'ei'.

Mikäli konetta M käytetään kyllä/ei-kysymyksen ratkaisun sijaan funktion $f : \mathbb{N}^m \to \mathbb{N}$ laskemiseen, voidaan simuloitu nauha rekonstruoida M':n nauharekistereistä $f_T^{-1}(\bar{1})f_\Gamma^{-1}(\bar{2})f_T^{-1}(\bar{3})$. Funktion f laskun tulos voidaan lukea nauhalta samalla tavalla kuin jos alkuperäistä Turingin konetta M olisi käytetty laskun suorittamiseen.

□

Esimerkki 12. Olkoon M esimerkin 7 mukainen Turingin kone joka tunnistaa kielen $\{a^n b^n | n \geq 0\}$.

M:n nauhamerkistö Γ sisältää kolme merkkiä. Numeroidaan ne seuraavasti: $f_\Gamma(B) = 0, f_\Gamma(a) = 1, f_\Gamma(b) = 2$. M sisältää viisi eri tilaa, numeroidaan ne seuraavasti: $f_Q(q_0) = 0, f_Q(q_a) = 1, f_Q(q_b) = 2, f_Q(q_r) = 3, f_Q(q_e) = 4$.

Olkoon koneessa M' 6 rekisteriä:

1	Vasen puolinauha
2	Lukupään kohdalla oleva merkki
3	Oikea puolinauha
4	Tila
$5, 6$	Laskurekistereitä

Suorituksen päätyttyä M' palauttaa tulokseksi arvon tosi jos ja vain jos rekisterissä 4 on oleva arvo kuuluu M:n hyväksyvien tilojen joukkoon F. Tässä koneessa hyväksyvien tilojen joukko sisältää vain q_0:n jonka kokonaislukuesitys on 0.

Seuraava komentolistaus sisältää jonkin verran optimointia toistamisen välttämiseksi, mutta noudattaa pääpiirteissään yllä esitettyä mallia.

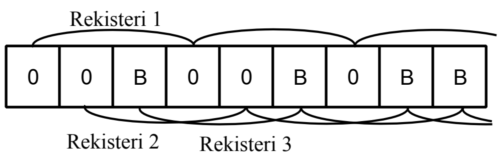
		Selitys
$\hat{1}$	(2, 2, 4)	Luettiin merkki B
$\hat{2}$	(2, 3, 9)	Luettiin merkki a
$\hat{3}$	(2, 4, 14)	Luettiin merkki b
$\hat{4}$	(4, 5, 81)	Luettiin merkki B ja oltiin tilassa q_0
$\hat{5}$	(4, 6, 20)	Luettiin merkki B ja oltiin tilassa q_a
$\hat{6}$	(4, 7, 22)	Luettiin merkki B ja oltiin tilassa q_b
$\hat{7}$	(4, 8, 49)	Luettiin merkki B ja oltiin tilassa q_r
$\hat{8}$	(4, 9, 22)	Luettiin merkki B ja oltiin tilassa q_e
$\hat{9}$	(4, 10, 26)	Luettiin merkki a ja oltiin tilassa q_0
$\hat{10}$	(4, 11, 27)	Luettiin merkki a ja oltiin tilassa q_a
$\hat{11}$	(4, 12, 22)	Luettiin merkki a ja oltiin tilassa q_b
$\hat{12}$	(4, 13, 29)	Luettiin merkki a ja oltiin tilassa q_r
$\hat{13}$	(4, 14, 22)	Luettiin merkki a ja oltiin tilassa q_e
$\hat{14}$	(4, 15, 22)	Luettiin merkki b ja oltiin tilassa q_0
$\hat{15}$	(4, 16, 33)	Luettiin merkki b ja oltiin tilassa q_a
$\hat{16}$	(4, 17, 36)	Luettiin merkki b ja oltiin tilassa q_b
$\hat{17}$	(4, 18, 39)	Luettiin merkki b ja oltiin tilassa q_r
$\hat{18}$	(4, 81, 22)	Luettiin merkki b ja oltiin tilassa q_e
$\hat{19}$	(4, 49)	Siirrytään tilaan q_a, kirjoitetaan B ja siirrytään oikealle
$\hat{20}$	(4, 21)	Siirrytään tilaan q_b, kirjoitetaan B ja siirrytään vasemmalle
$\hat{21}$	(4, 65)	
$\hat{22}$	(4, 23)	Siirrytään tilaan q_e ja lopetetaan suoritus
$\hat{23}$	(4, 24)	
$\hat{24}$	(4, 25)	
$\hat{25}$	(4, 81)	
$\hat{26}$	(4, 49)	Siirrytään tilaan q_a, kirjoitetaan B ja siirrytään oikealle
$\hat{27}$	(4, 28)	Pysytään tilassa q_a, kirjoitetaan a ja siirrytään oikealle
$\hat{28}$	(2, 49)	
$\hat{29}$	(4, 30)	Pysytään tilassa q_r, kirjoitetaan a ja siirrytään vasemmalle
$\hat{30}$	(4, 31)	
$\hat{31}$	(4, 32)	
$\hat{32}$	(2, 65)	
$\hat{33}$	(4, 34)	Pysytään tilassa q_a, kirjoitetaan b ja siirrytään oikealle
$\hat{34}$	(2, 35)	
$\hat{35}$	(2, 49)	
$\hat{36}$	(4, 37)	Siirrytään tilaan q_r, kirjoitetaan B ja siirrytään vasemmalle

		Selitys
$\hat{37}$	(4, 38)	
$\hat{38}$	(4, 65)	
$\hat{39}$	(4, 40)	Pysytään tilassa q_r, kirjoitetaan b ja siirrytään vasemmalle
$\hat{40}$	(4, 41)	
$\hat{41}$	(4, 42)	
$\hat{42}$	(2, 43)	
$\hat{43}$	(2, 65)	
$\hat{44}$	(4, 40)	Pysytään tilassa q_r, kirjoitetaan b ja siirrytään vasemmalle
$\hat{45}$	(4, 41)	
$\hat{46}$	(4, 42)	
$\hat{47}$	(2, 43)	
$\hat{48}$	(2, 65)	
$\hat{49}$	(1, 50, 51)	Siirrytään oikealle. Siirretään rekisterin 1 arvo rekisteriin 5
$\hat{50}$	(5, 49)	
$\hat{51}$	(5, 52, 55)	Lisätään rekisterin 5 sisältö kolme kertaa rekisteriin 1
$\hat{52}$	(1, 53)	
$\hat{53}$	(1, 54)	
$\hat{54}$	(1, 51)	
$\hat{55}$	(2, 56, 57)	Lisätään rekisterin 2 arvo rekisteriin 1
$\hat{56}$	(1, 55)	
$\hat{57}$	(3, 58, 63)	Jaetaan rekisteri 3 kolmella, asetetaan jakojäännös rekisteriin 2
$\hat{58}$	(3, 59, 62)	
$\hat{59}$	(3, 60, 61)	
$\hat{60}$	(5, 57)	
$\hat{61}$	(2, 62)	Jakojäännös 2
$\hat{62}$	(2, 63)	Jakojäännös 1
$\hat{63}$	(5, 64, 1)	Siirrä rekisterin 5 sisältö rekisteriin 3, paluu alkuun
$\hat{64}$	(3, 63)	
$\hat{65}$	(3, 66, 67)	Siirrytään vasemmalle. Siirretään rekisterin 3 arvo rekisteriin 5
$\hat{66}$	(5, 65)	
$\hat{67}$	(5, 68, 71)	Lisätään rekisterin 5 arvo rekisteriin 3 kolme kertaa
$\hat{68}$	(3, 69)	
$\hat{69}$	(3, 70)	
$\hat{70}$	(3, 67)	
$\hat{71}$	(2, 72, 73)	Lisätään rekisterin 2 arvo rekisteriin 3
$\hat{72}$	(3, 71)	

		Selitys
$\hat{73}$	(1, 74, 79)	Jaetaan rekisteri 1 kolmella, asetetaan jakojäännös rekisteriin 2
$\hat{74}$	(1, 75, 78)	
$\hat{75}$	(1, 76, 77)	
$\hat{76}$	(5, 73)	
$\hat{77}$	(2, 78)	Jakojäännös 2
$\hat{78}$	(2, 79)	Jakojäännös 1
$\hat{79}$	(5, 80, 1)	Siirrä rekisterin 5 sisältö rekisteriin 1, paluu alkuun
$\hat{80}$	(1, 79)	
$\hat{81}$	halt	

Lause 13. *Olkoon P ongelma joka on ratkaistavissa rekisterikoneella. Ongelma P on ratkaistavissa Turingin koneella.*

Todistus. Olkoon M rekisterikone joka ratkaisee ongelman P. Lauseen todistamiseksi rakennetaan sellainen Turingin kone M', joka pystyy simuloimaan rekisterikoneen M toimintaa. Olkoon rekisterikoneella R rekisteriä. Tällöin tallennetaan yhden rekisterin sisälltö koodattuna joka R:nteen muistipaikkaan valitusta rajasta eteenpäin. Esimerkiksi jos $R = 2$, joka toinen nauhan muistipaikka kuuluisi rekisteriin 1 ja joka toinen rekisteriin 2. Rekisterien sisältö voidaan koodata nauhalle monin eri tavoin. Yksinkertaisuuden vuoksi käytetään tässä unaarista lukujärjestelmää.



Kuva 3: Rekisterien sijoittelu nauhalle tapauksessa R = 3. Rekistereissä ovat kuvattuna arvot $(3, 2, 0)$, nauha on muualla tyhjä.

Ensimmäiseksi annettu syöte tulee muuntaa edellä kuvattuun muotoon. Tarvittavan aliohjelman teko on helppoa.

Sen jälkeen kirjoitetaan aliohjelmat, jotka suorittavat samat toiminnot kuin (n, j) ja (n, j, k). Kannattanee pitää tarkoitukseen varattua merkkiä rekisterialueen vasemmalla puolen, jotta se löytyy helpommin. Tässä tarkoitukseen käytetään merkkiä 1. Olkoon Turingin koneen lukupää tällä merkillä seuraavia aliohjelmia varten, ja olkoon se komentoa (n, j) vastaavassa tilassa. Aluksi kone siirtyy $n - 1$ merkkiä oikealle, jolloin se saavutta rekisteriä n vastaavan alueen. Tämän jälkeen se siirtyy oikealle R:n pituisin askelin, kunnes se löytää tyhjän merkin. Kone asettaa sen 0:ksi, palaa takaisin vasemmalle merkin 1 luokse ja muuttaa tilansa rekisterikoneen komentoa $\hat{\text{j}}$ vastaavaksi. Tällä tavoin rekisterin n arvoa on kasvatettu yhdellä. Olkoon kone nyt komentoa (n, j, k) vastaavassa tilassa. Kone siirtyy ensin $n-1$ merkkiä oikealle. Jos sillä kohdalla oleva merkki on tyhjä, palataan kohtaan 1 ja siirrytään rekisterikoneen komentoa $\hat{\text{k}}$ vastaavaan tilaan. Jos merkki ei ollut tyhjä, siirrymme oikealle R:n pituisin askelin kunnes löydämme tyhjän merkin. Siirrymme R merkkiä vasemmalle, merkitsemme sen tyhjäksi, siirrymme takaisin merkin 1 luo ja siirrymme komentoa $\hat{\text{j}}$ vastaavaan tilaan.

Kun komento 'Seis' saavutetaan, ensimmäisen rekisterin sisältö talletetaan ja muu nauha tyhjennetään.

□

Esimerkki 14. Olkoon M esimerkin 6 mukainen rekisterikone joka suorittaa kahden luvun yhteenlaskun. Rakennamme Turingin koneen joka simuloi koneen M toimintaa lauseen 9 kuvailemalla tavalla.

Muodollisesti määriteltynä olkoon M' Turingin kone $(\{q_a, ...q_k\}, \{0, 1\}, \{0, 1, B\}, \delta, q_j, B, \{q_k\})$ jossa δ on määritelty seuraavan taulukon kautta:

		0	1	B	Selitys
$\hat{1}$	q_a	$(q_b, 0, R)$	$\emptyset$	(q_b, B, R)	Siirry rekisterin 2 alkuun
$\hat{1}$	q_b	$(q_c, 0, R)$	$\emptyset$	q_k	Tarkista ehto
$\hat{1}$	q_c	$(q_d, 0, R)$	$\emptyset$	(q_d, B, R)	Ohitettiinko rekisterin 2 viimein merkki?
$\hat{1}$	q_d	$(q_c, 0, R)$	$\emptyset$	(q_e, B, L)	Siirry rekisterin 2 seuraavaan merkkiin
$\hat{1}$	q_e	$(q_f, 0, L)$	$\emptyset$	(q_f, B, L)	Palaa rekisterin 2 viimeiseen merkkiin
$\hat{1}$	q_f	(q_g, B, L)	$\emptyset$	$\emptyset$	Poista merkki rekisteristä 2
$\hat{2}$	q_g	$(q_g, 0, L)$	$(q_h, 1, R)$	(q_g, B, L)	Palaa alkuun
$\hat{2}$	q_h	$(q_i, 0, R)$	$\emptyset$	$(q_j, 0, L)$	Lisää merkki rekisterin 1 loppuun jos ollaan lopussa
$\hat{2}$	q_i	$(q_h, 0, R)$	$\emptyset$	(q_i, B, R)	Siirry rekisterin 1 seuraavaan merkkiin
$\hat{2}$	q_j	$(q_j, 0, L)$	$(q_a, 1, R)$	(q_j, B, L)	Palaa alkuun
$\hat{3}$	q_k	$\emptyset$	$\emptyset$	$\emptyset$	Suoritus päättyy

Alkutilanne on koodattu nauhalle siten, että ensimmäinen merkki on 1 ja sen oikealla puolen on rekisterialue. Rekisterialue on jaettu limittäin kahteen osaan; joka toinen alueen merkki kuuluu rekisteriin yksi ja joka toinen rekisteriin kaksi. Rekisterin sisältämä luonnollinen luku on ilmaistu unaariesityksellä, missä 0-merkkien lukumäärä on suoraan rekisterin sisältämä arvo.

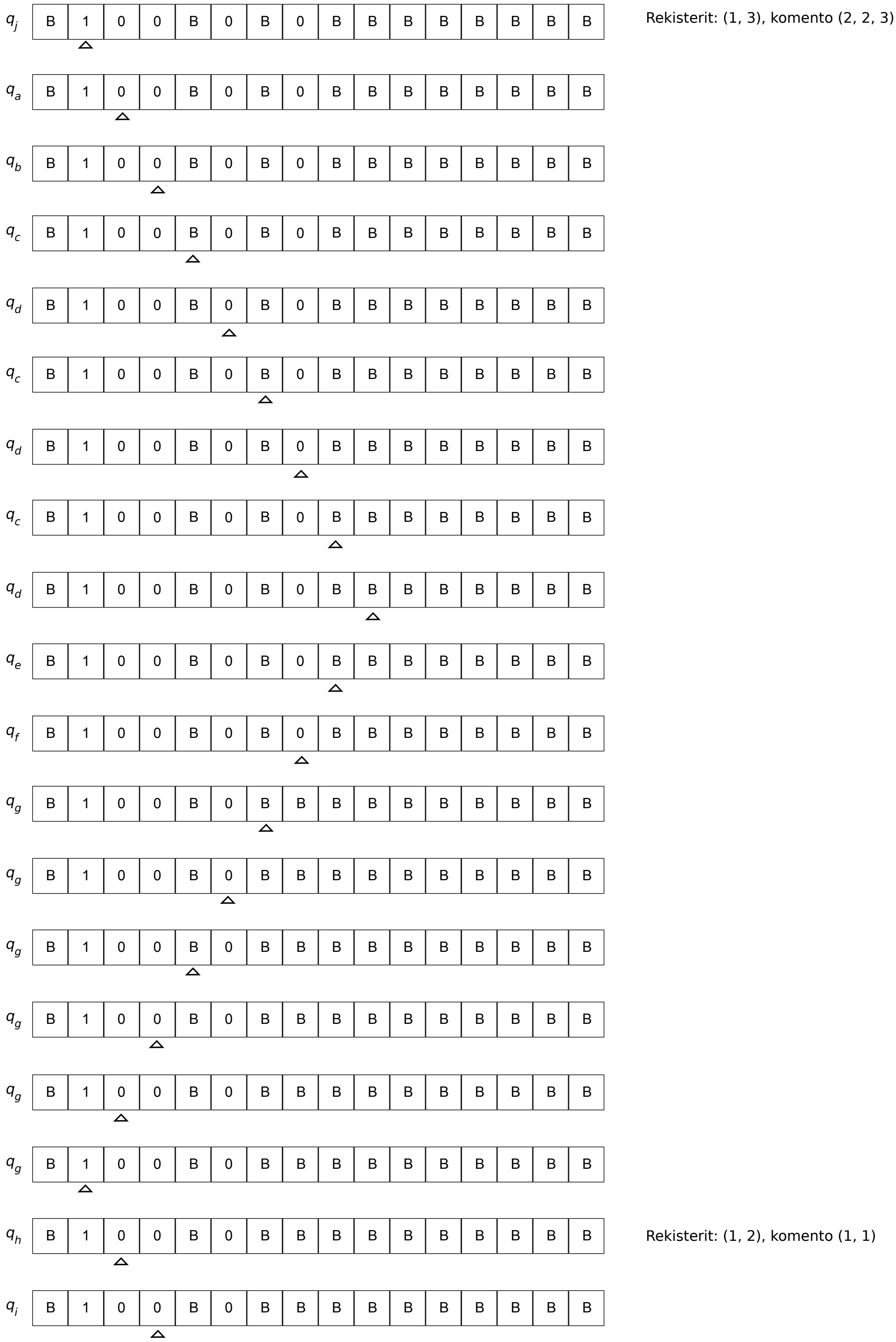
Simulaatio syötteelle $(1, 3)$ on esitetty kuvissa 5-8.

Lause 15. *Olkoon P ongelma. P on ratkaistavissa rekisterikoneella jos ja vain jos se on ratkaistavissa Turingin koneella.*

Todistus. Todistus on selvä lauseiden 11 ja 13 nojalla. Lauseiden nojalla havaitaan myös, että mikäli algoritmin suoritus yhdellä koneella ei koskaan pääty, myöskään toisella koneella simuloitu suoritus ei koskaan pääty. □

A Lähteet

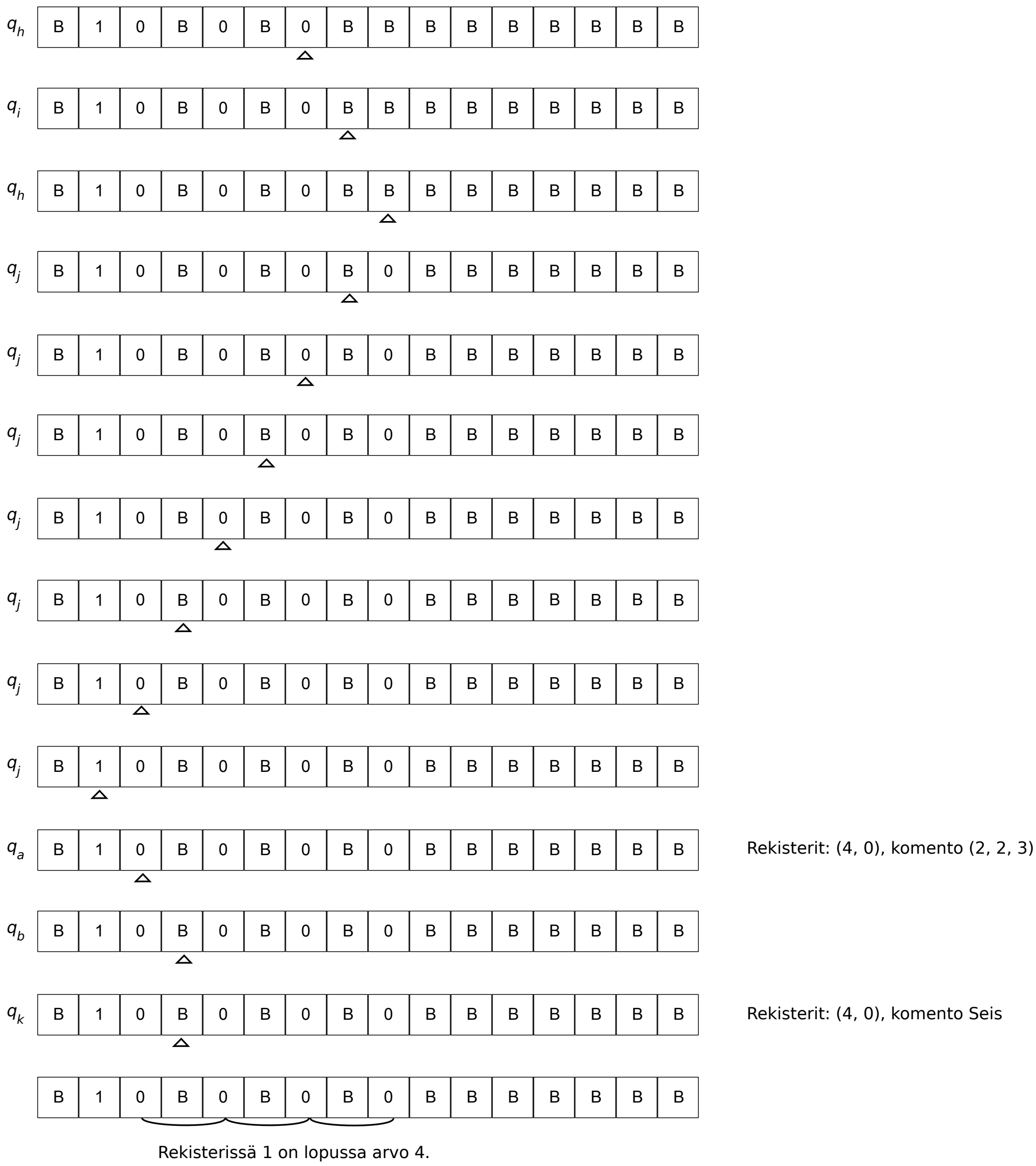
J. Truss: Discrete Mathematics For Computer Scientists, Addison-Wesley, 1999.



Kuva 4: Rekisterikoneen simulointi Turingin koneella

Kuva 5: Rekisterikoneen simulointi Turingin koneella

Kuva 6: Rekisterikoneen simulointi Turingin koneella



Kuva 7: Rekisterikoneen simulointi Turingin koneella