Ratkeavuuden ja ratkeamattomuuden välinen raja ja Postin vastaavuusongelma

Vesa Halava
Turun yliopisto
Matematiikan laitos
TUCS - Turun tietotekniikan tutkimuskeskus
vesa.halava@utu.fi

1 Johdanto

Algoritmisen ratkeamattomuuden käsitettä pidetään yhtenä 1900-luvun suurimmista filosofisista keksinnöistä. Vuosisadan alussa David Hilbert esitti otaksuman/toiveen, että kaikki matemaattiset päätösongelmat ovat ratkeavia. *Päätösongelmalla* tarkoitetaan tässä ongelmaa, joka koostuu joukosta *tapauksia*, joilla kaikilla on tai ei ole jokin tunnettu ominaisuus. Tehtävänä on päätellä, onko syötteeksi saadulla tapauksella kyseinen ominaisuus vai ei.

Päätösongelman ratkaisee algoritmi, joka jokaiselle tapaukselle vastaa oikein ja toimii äärellisessä ajassa. Päätösongelmaa sanotaan *ratkeavaksi*, jos tällainen algoritmi on olemassa. Hilbert siis toivoi, että kaikille hyvinmääritellyille päätösongelmille on olemassa algoritmi.

Esimerkki 1. *Tarkastellaan ongelmaa, jossa tapausten joukko on* $\mathbb{Z}_+ = \{1,2,\dots\}$*, ja kysytään, onko syötteenä saatu luku alkuluku vai ei. Tämä ongelma on tietysti ratkeava. Esimerkiksi algoritmi, joka kokeilee syötteelle n jakaako jokin luvuista* $2,3,4,\dots,\lfloor\sqrt{n}\rfloor$ *luvun n, ratkaisee tämän ongelman. Jos n on jaollinen jollakin näistä luvuista, se ei ole alkuluku, muulloin taas on.*

Algoritmin voidaan ajatella olevan äärellinen prosessi laskutoimituksia, laskuaskelia. Algoritmin teoreettisena mallina pidetään ns. *Turingin konetta*. Turingin kone koostuu äärettömästä luku- ja kirjoitus*nauhasta*, äärellisestä muistista, jota kutsutaan *tilajoukoksi* ja äärellisestä joukosta *sääntöjä*, joilla tiloja ja nauhan sisältöä muutetaan (ks. esimerkiksi [17]).

Jo 1900-luvun alkupuoliskolla huomattiin, että Hilbertin otaksuma oli väärä, kun löydettiin ns. *ratkeamattomia* päätösongelmia, joille ei ole olemassa ratkaisualgoritmia. Huomattiin, että on olemas-

sa jopa hyvin yksinkertaisesti määriteltyjä ratkeamattomia ongelmia. Yksi näistä on ns. *Postin vastaavuusongelma* (engl. Post Correspondence Problem), lyhyesti PCP, johon tässä työssä syvennytään.

PCP:n määritteli ja todisti ratkeamattomaksi Emil Post vuonna 1946 (ks. [15]). PCP:ssä tapaus koostuu kahdesta yhtäpitkästä listasta sanoja ja kysytään, voidaanko listojen sanoista muodostaa sama sana yhdistämällä aina vastaavilla paikoilla listoissa olevat sanat.

Esimerkki 2. *Olkoot nyt ensimmäinen lista* (abb, b, a) *ja toinen lista* (a, abb, bb). *Nyt valitsemalla listoista sanat 1, 3, 1, 1, 3, 2 ja 2 saadaan sana*

$$\begin{aligned} & abb \cdot a \cdot abb \cdot abb \cdot a \cdot b \cdot b \\ = {} & abbaabbabbabb \\ = {} & a \cdot bb \cdot a \cdot a \cdot bb \cdot abb \cdot abb, \end{aligned}$$

missä $\cdot$ *on listan sanojen välimerkkinä osoittamassa sanojen jakoa listan sanoiksi. Ensimmäisellä rivillä on ensimmäisen listan ja kolmannella rivillä toisen listan mukainen jako. Siis tässä tapauksella vastaus PCP:n kysymykseen on kyllä.*

Muista kuuluisista ratkeamattomista ongelmista mainittakoon *Turingin koneiden pysähtymisongelma* ja ns. *Hilbertin kymmenes ongelma*. Turingin koneiden pysähtymisongelma voidaan ajatella seuraavasti: Onko olemassa algoritmia, joka saadessaan syötteeksi minkä tahansa algoritmin ja jonkin sen syötteen, päättäisi pysähtyykö syötealgoritmi kyseisellä syötteellä vai ei? Tällaista algoritmia ei siis ole olemassa.

Kun Hilbert esitti toiveen, että kaikki matemaattiset päätösongelmat olisivat ratkeavia, hän listasi myös mielestään tärkeimmät matematiikan ongelmat, jotka pitäisi ratkaista. Listassa ei siis ollut pelkästään päätösongelmia, vaan yleisesti kaikenlaisia matematiikan ongelmia. Yhteensä listassa oli 23 ongelmaa, mukana mm. Riemannin hypoteesi. Listan kymmenes ongelma on hyvin kuuluisa päätösongelma. Hilbertin kymmenennessä ongelmassa kysyttiin, onko olemassa ratkaisualgoritmia *Diofantoksen yhtälöille*. Diofantoksen yhtälöt ovat kokonaislukukertoimisia yhtälöitä, muuttujia voi olla useita, ja ratkaisuiksi hyväksytään vain kokonaislukuja. Tämä ongelma tiedetään ratkemattomaksi (ks. [12]).

Matriisien teoriasta löytyy useita yksinkertaisesti määriteltyjä ratkeamattomia ongelmia. Kuuluisin niistä on ns. *mortality–ongelma*, jota tarkastellaan seuraavaksi.

Esimerkki 3. *Mortality–ongelman tapaus koostuu* 3×3 *-kokonaislukumatriiseista* $A_1, A_2, \ldots, A_n$ *ja kysytään, onko vai eikö ole olemassa sellaista jonoa* $i_1, i_2, \ldots, i_k$, *että*

$$A_{i_1} A_{i_2} \cdots A_{i_k} = 0.$$

Tämä ongelma on siis ratkeamaton (ks. [14] tai [3]). Mortality–ongelma voidaan ajatella myös lineaarikuvausten ongelmaksi, ts. on ratkeamatonta saadaanko annettujen kolmiulotteisen reaaliavaruuden lineaarikuvausten kompositiona nollakuvaus.

On syytä huomata, että jos ongelmassa mahdollisia syötteitä on äärellinen määrä, ongelma on ratkeava. Tämä seuraa siitä, että kaiki tapaukset voidaan käydä läpi. Jos tarkastellaan ratkeamattoman ongelman tapausten äärellistä osajoukkoa, on tämä ongelma ratkeava. Esimerkiksi yksittäisille sanalistapareille PCP on ratkeava tai jollekin yhdelle algoritmille ja sen syötteelle pysähtymisongelma on helppo ratkaista. Toisaalta, jos ongelmalla on ääretön määrä tapauksia, se voi silti olla ratkeava.

Ratkeamattomien ongelmien tapausten joukot ovat hyvin laajoja, esimerkiksi PCP:ssä syötteenä voi olla mikä tahansa, jopa minkä tahansa pituiset, sanalistat ja pysähtymisongelmassa syöttenä on mikä tahansa algoritmi ja sen mikä tahansa syöte. Huomaa, että pysähtymisongelma on ratkeamaton vaikka syötteeksi kiinnitettäisiin tyhjä merkkijono.

On selvää, että jokainen hyvinmääritelty päätösongelma on joko ratkeava tai ratkeamaton. Valitettavasti vain on olemassa monia tärkeitä ongelmia, joista ei tiedetä, ovatko ne ratkeavia vai ratkeamattomia.

Esimerkki 4. *Ns.* Skolemin ongelman *ratkeavuus/ratkeamattomuus on yhä avoin ongelma. Ongelmassa tapaukset ovat* 3×3 *-kokonaislukumatriiseja ja kysytään, onko vai ei ole olemassa sellaista luonnollista lukua n, että syötteenä saadun matriisin n:nnen potenssin oikean yläkulman alkio on nolla.*

Tämä ongelma on ratkeava, jos tapaukset ovat 2×2*- matriiseja.*

Laskettavuuden teoriassa ratkeamattomuuden tutkimuksessa on kaksi pääsuuntaa. Tärkein on tietysti uusien ongelmien ratkeamattomuus- tai ratkeavuustodistukset. Toinen pääsuunta on ratkeavuuden ja ratkeamattomuuden välisen rajan etsintä. Tätä rajaa etsitään tarkastelemalla ratkeamatonta ongelmaa ja etsimällä sen tapausten joukon osajoukkoja, joihin rajoituttaessa ongelma onkin ratkeava. Yleensä tällaisen osajoukon tapauksilla on jokin rajoittava ominaisuus, joka helpottaa ongelman ratkaisua.

Esimerkki 5. *PCP:n tapauksissa kummassakin listassa sanoja voi olla n kappaletta, missä n voi olla mikä tahansa ei-negatiivinen kokonaisluku. Ongelma on siis hyvin laaja, koska listojen pituutta tai sanojen pituutta ei ole millään lailla rajattu.*

Jos listojen pituuksia rajoitetaan, niin tiedetään, että jos listoissa on korkeintaan kaksi sanaa, eli $n \leq 2$*, niin PCP on ratkeava (ks. [1, 6]). Jos* $n = 1$*, niin tapaukset ovat triviaalisti ratkeavia, mutta ns.* binaarisen *PCP:n, eli tapauksien, missä* $n = 2$*, ratkaisualgoritmi on hyvin tekninen.*

Toisaalta jos $n \geq 7$*, niin PCP on ratkeamaton (ks. [13]). Jos tarkastellaan listojen pituuksia, ratkeavuuden ja ratkeamattomuuden välinen raja on siis jossain kahden ja seitsemän välissä. Missä se tarkalleen on, sitä ei tiedetä.*

Tutkimalla ratkeavuuden ja ratkeamattomuuden välistä rajaa tutkitaan siis niitä ongelman tapausten ominaisuuksia, jotka vaikuttavat ratkeavuuteen/ratkeamattomuuteen.

Tässä työssä tarkastellaan ratkeavuuden ja ratkeamattomuuden välistä rajaa Postin vastaavuusongelmassa, kun listoissa olevia sanoja rajoitetaan. Aloitetaan sanojen peruskäsitteiden määritelmillä.

2 Sanat

Olkoon $B = \{b_1, b_2, \ldots, b_m\}$ mikä tahansa äärellinen joukko. Sitä kutsutaan *aakkostoksi* ja sen alkioita kirjaimiksi. Aakkoston B *sana* on jono B:n kirjaimia. Sanan pituus on siinä esiintyvien kirjainten lukumäärä. Joukko B^* on kaikkien aakkoston B äärellisten sanojen joukko. Huomaa, että ns. *tyhjä sana* (merkitään ε) kuuluu joukkoon B^* (ja sen pituus on nolla). Lisäksi merkitään $B^+ = B^* \setminus \{\varepsilon\}$.

Esimerkki 6. *Joukon B^* sanojen u ja v voidaan* yhdiste *on sana uv, joka määritellään luonnollisella tavalla. Esimerkiksi (binaarisessä) aakkostossa $\{a,b\}$, jos $u = ab$ ja $v = baabba$, niin $uv = abbaabba$ ja $vu = baabbaab$.*

Seuraavaksi esitellään sanojen välisiä relaatioita. Sanotaan, että sana u on sanan v *prefiksi*, jos $v = uw$ jollekin sanalle w. Tätä merkitään $u \leq v$. Vastaavasti u on sanan v *suffiksi*, jos $v = wu$. Esimerkiksi sana ab on sanan $abcb$ prefiksi ja sana acb on sanan $abbcbacb$ suffiksi.

Lisäksi, jos $u \leq v$ tai $v \leq u$, niin sanotaan, että u ja v ovat *vertailtavissa*, merkitään $u \bowtie v$. Esimerkiksi sanat ab ja $abcb$ ovat vertailtavissa, kun taas sanat aa ja $abba$ eivät ole vertailtavissa.

Aakkoston B *ääretön sana* on (oikealle) ääretön jono aakkoston B kirjaimia. Esimerkiksi $\omega = ababab\cdots$ on aakkoston $\{a,b\}$ ääretön sana.

3 Postin vastaavuusongelma

Seuraavaksi esitetään PCP:n tarkka määritelmä. Sen tapaukset koostuvat kahdesta n-pituisesta listasta $\mathbf{u}$ ja $\mathbf{v}$ sanoja, siis

$$\begin{aligned} \mathbf{u} &= (u_1, u_2, \ldots, u_n) \quad \text{ja} \\ \mathbf{v} &= (v_1, v_2, \ldots, v_n), \end{aligned} \tag{1}$$

missä u_i ja v_i ovat sanoja kaikilla $1 \leq i \leq n$. PCP:ssä kysytään, onko vai eikö ole olemassa sellaista jonoa indeksejä $i_1, i_2, \ldots, i_k$, että $k \geq 1$ ja

$$u_{i_1} u_{i_2} \cdots u_{i_k} = v_{i_1} v_{i_2} \cdots v_{i_k}.$$

Listojen $\mathbf{u}$ ja $\mathbf{v}$ sanoista muodostetaan uusi sana yhdistämällä, ja PCP:ssä kysytään siis, että voidaanko listojen $\mathbf{u}$ ja $\mathbf{v}$ sanoista muodostaa yhdistämällä sama sana valitsemalla aina vastaavilla paikoilla listoissa olevat sanat.

Lukua n kutsutaan tapauksen *kooksi* ja jonoa $i_1, i_2, \ldots, i_k$ tapauksen *ratkaisuksi*.

Esimerkki 7. *Tapauksen* $\mathbf{u} = (abab, aaa, ab, a)$ *ja* $\mathbf{v} = (a, ba, aaab, aaa)$*, lyhin ratkaisu on*

$$12241422443.$$

Vaikka tapaus näyttää helpolta, voi sen lyhin ratkaisu olla pitkä. Esimerkiksi tapauksen, missä $\mathbf{u} = (bab, b, aba)$ *ja* $\mathbf{v} =$

$(b, ab, bbab)$ lyhimmän ratkaisun pituus on 216. Lisää pieniä PCP:n tapauksia, joiden minimiratkaisu on pitkä, löytyy osoitteesta [18].

Nykyään PCP formuloidaan yleensä *sanamorfismien* avulla. Olkoot A ja B kaksi aakkostoa. Kuvaus $h\colon A^* \to B^*$ on (sana)morfismi, jos se toteuttaa ehdon $h(uv) = h(u)h(v)$ kaikilla $u, v \in A^*$. Sanotaan, että sana $h(u)$ on sanan *u kuva* morfismissa h.

Määritellään nyt PCP uudelleen. Olkoon $A = \{1, 2, \ldots, n\}$ ja määritellään morfismit $h, g\colon A^* \to B^*$ siten, että kaikilla $i \in A$,

$$h(i) = u_i \quad \text{ja} \quad g(i) = v_i,$$

missä u_i ja v_i ovat sanoja listoista (1). PCP:n kysymys saadaan nyt muotoon, onko vai eikö ole olemassa sellaista sanaa $w \in A^+$, että

$$h(w) = g(w).$$

PCP:n tapaus määritellään siis morfismiparina (h, g), missä $h, g\colon A^* \to B^*$ ja tapauksen (h, g) koko on aakkoston A alkioiden lukumäärä $|A|$. Sana w, jolle $h(w) = g(w)$, on tapauksen ratkaisu. Huomaa, että ratkaisun w pitää olla epätyhjä.

Esimerkki 8. *Esimerkin 7 tapaus voidaan esittää morfismien avulla seuraavasti: $h, g\colon \{1,2,3,4\}^* \to \{a,b\}^*$, missä*

	1	2	3	4
h	*abab*	*aaa*	*ab*	*a*
g	*a*	*ba*	*aaab*	*aaa*

PCP on hyvin käyttökelpoinen todistettaessa muita ratkeamattomuustuloksia, esimerkiksi sanojen kombinatoriikassa, automaattien ja formaalisten kielten teoriassa tai matriisiteoriassa. PCP:n avulla voidaan esimerkiksi osoittaa, että mortality–ongelma on ratkeamaton, ks. esimerkki 3.

Postin vastaavuusongelman tapauksia voidaan rajoittaa monella tapaa tutkittaessa ratkeavuuden ja ratkeamattomuuden rajaa. Edellä esimerkissä 5 nähtiin jo, missä raja suunnilleen on, jos rajoitetaan tapausten kokoa. Vastaavasti, jos rajoitetaan listoissa olevien sanojen pituuksia (tai oikeastaan kirjainten kuvien pituuksia morfismeissa), niin tiedetään, että PCP on ratkeava, kun sanojen pituus on yksi, mutta jos sanojen pituus on korkeintaan kaksi, niin PCP on ratkeamaton (ks. [7]).

4 Merkitty PCP

Tässä luvussa tarkastellaan tapauksia, joissa morfismien kuvat toteuttavat lisäehdon. Aloitetaan *injektioista*. Morfismi $h\colon A^* \to B^*$ on injektio, jos kaikilla $u, v \in A^*$

$$h(u) = h(v) \implies u = v.$$

Injektio on siis tietyssä mielessä yksikäsitteinen kuvaus joukkoon B^*, nimittäin jos $w \in B^*$ ja $h(u) = w$ jollekin $u \in A^*$, niin kaikille muille sanoille v, $h(v) \neq w$. Huomaa, että jos käytetään PCP:n sanalistamääritelmää (1), niin injektiotapauksissa jokainen $w \in B^*$ voidaan muodostaa lis-

tan sanojen yhdisteenä korkeintaan yhdellä tavalla.

Lause 1. *[11] PCP on ratkeamaton tapauksissa, joissa morfismit ovat injektioita.*

Tämä tulos oli ensimmäinen todellinen parannus Postin alkuperäiseen tulokseen. Lauseen tapaukset ovat siis selvästi rajoitettuja, mutta injektiivisyys ei vielä riitä ratkeavuuteen. Rajoitetaan morfismeja lisää, ja oletetaan, että morfismit ovat *prefiksejä* (tai oikeastaan *prefiksivapaita*) eli minkään kirjaimen kuva ei ole toisen kirjaimen kuvan prefiksi. Tarkemmin, h on prefiksi, jos kaikilla $a,b \in A$, $z \in B^*$,

$$h(a) = h(b)z \implies a = b.$$

Morfismi h on *biprefiksi*, jos se on prefiksi ja *suffiksi*, eli kaikilla $a,b \in A$, $z \in B^*$,

$$h(a) = zh(b) \implies a = b.$$

Huomaa, että jos morfismi on biprefiksi tai vaikka prefiksi, se on myös injektio. Jos käytetään PCP:n sanalista määritelmää (1), niin tapaus on prefiksi, jos kummankaan listan sanat eivät ole keskenään vertailtavissa.

Lause 2. *[16] PCP on ratkeamaton tapauksissa, joissa morfismit ovat biprefiksejä. Erityisesti siis PCP on ratkeamaton tapauksissa, joissa morfismit ovat prefiksejä.*

Tämä tulos on siinä mielessä yllättävä, että prefiksitapausten ratkaisujen joukko on säännöllinen kieli. Tiedetään siis, että ratkaisujen muodostama kieli voidaan hyväksyä äärellisellä automaatilla, mutta on edelleen ratkeamatonta, sisältääkö kyseinen kieli epätyhjiä sanoja. Huomaa, että kyseistä ratkaisujen joukon hyväksyvää äärellistä automaattia ei voida muodostaa pelkän syötteenä saadun PCP:n tapauksen pohjalta. Jos voitaisiin, niin prefiksi PCP olisi ratkeava, sillä äärellisille automaateille ns. *tyhjyysongelma*, missä kysytään hyväksyy annettu automaatti yhtään epätyhjää sanaa, on ratkeava.

Rajoitetaan morfismeja lisää ja tarkastellaan tapauksia, joissa morfismit ovat *merkittyjä*. Morfismi on merkitty, jos jokaisen kirjaimen kuvan ensimmäinen kirjain on eri. Merkityt morfismit ovat siis prefiksejä, koska jokaisen kirjaimen kuva eroaa muiden kirjainten kuvista jo ensimmäisellä merkillä. Jos tarkastellaan PCP:n alkuperäistä määritelmää, niin tapauksessa, joka ovat merkitty, listan **u** kaikki sanat alkavat eri kirjaimella, kuten myös listan **v** sanat.

Esimerkki 9. *Morfismi* $h\colon \{a,b,c\} \to \{a,b,c\}$*, missä*

	a	b	c
h	abb	bb	$cbbb$

on merkitty.

Lause 3. *[9], [2] PCP on ratkeava tapauksissa, joissa morfismit ovat merkittyjä.*

Seuraus 1. *Ratkeavuuden ja ratkeamattomuuden raja on jossain prefiksi ja merkittyjen morfismien välissä.*

Tarkastellaan nyt lyhyesti edellisen lauseen todistusta, ts. tarkastellaan merkityt tapaukset ratkaisevaa algoritmia. Tarkat todistukset ja algoritmi kokonaisuudessaan löytyvät lähteestä [2].

Jos tapauksessa (h,g) morfismit ovat merkittyjä, niin lyhin kirjaimella $a \in A$ alkava ratkaisu on yksikäsitteinen.

Esimerkki 10. *Olkoot morfismit* $h,g\colon \{a,b,c\} \to \{a,b,c\}$*, missä*

	a	b	c
h	abb	bb	$cbbb$
g	a	bbb	c

Kirjaimelle a löydetään ratkaisu, kun tarkastellaan sanojen $h(a)$ *ja* $g(a)$ *eroa,* $h(a) = abb = g(a)bb$*. Siis seuraavan kirjaimen kuvan g:ssä täytyy alkaa b:llä. Nyt*

$$g(a)g(b)b = abbbb = h(a)h(b),$$

jne. Ratkaisuksi löydetään abb, sillä $h(abb) = abbbbbb = g(abb)$*.*

Entä jos valitulla kirjaimella alkavaa ratkaisua ei ole olemassa, mistä tiedetään pysähtyä ja lopettaa etsintä? Ei mistään. Pitää siis tarkastella ongelmaa toisella tavalla.

Muodostetaan ns. *palikoita*, joista mahdollinen ratkaisu muodostuu. Palikat ovat minimaalisia sanapareja (u,v), joille $h(u) = g(v)$. Jokaiselle aakkoston B kirjaimelle b muodostetaan oma palikka, jossa $h(u)$ alkaa b:llä. Huomaa, että sanoja u ja v ei rajoiteta millään tavalla. Niiden ei siis tarvitse olla esimerkiksi vertailtavissa.

Esimerkki 11. *Esimerkin 10 morfismeille palikat eri kirjaimille:*

$$a\colon (abb, abb),$$
$$b\colon (bbb, bb),$$
$$c\colon (c, cb).$$

Merkityn PCP:n tapauksen ratkaisuilla on yksikäsitteinen jako palikoihin, ts. jos $h(w) = g(w)$, niin

$$w = u_1u_2\cdots u_k = v_1v_2\cdots v_k, \qquad (2)$$

missä $k \geq 1$ ja (u_i, v_i) on palikka jollekin kirjaimelle $b_i \in B$ kaikilla $i = 1,\ldots,k$.

Algoritmin idea on redusoida annettu tapaus helpompaan tapaukseen, ja jatkaa tätä reduktiota kunnes saadaan tapaus, jossa ongelma on helppo ratkaista. Palikoiden avulla voidaan määritellään uusi merkityn PCP:n tapaus (h',g') seuraavasti. Kaikille kirjaimille $b \in B$, jos on olemassa palikka (u,v), missä $b \leq h(u)$, niin

$$h'(b) = u \quad \text{ja} \quad g'(b) = v.$$

Tapausta (h',g') kutsutaan tapauksen (h,g) *seuraajaksi*.

Esimerkki 12. *Esimerkin 10 seuraaja on siis edellisen esimerkin mukaan tapaus* (h',g')*, missä*

	a	b	c
h'	abb	bbb	c
g'	abb	bb	cb

Seuraaja on aina ekvivalentti alkuperäisen tapauksen kanssa eli sillä on ratkaisu, jos ja vain jos alkuperäisellä tapauksella on ratkaisu. Algoritmi perustuu

tämän seuraaja-reduktion iterointiin. Voidaan osoittaa, että tapaukset eivät muutu reduktiossa vaikeammiksi, ts. aakkostojen koko ei kasva, eikä tapauksen kuvien yhteenlaskettu pituus kasva. Huomaa, että tapausten aakkostot reduktioketjussa ovat kaikki aakkoston B osajoukkoja. Tapauksia, joissa aakkostot ovat B:n osajoukkoja ja kuvien pituus on ylhäältä rajoitettu, on äärellinen määrä. Näin ollen aikanaan saadaan reduktiosta siis tapaus, joka on esiintynyt reduktioketjussa jo aikaisemmin.

Lause 4. *Tapauksella, joka esiintyy reduktioketjussa useammin kuin kerran, on ratkaisu, jos ja vain jos sillä on ratkaisu, jonka pituus on yksi, ts. jokin kirjain on ratkaisu.*

Merkityn PCP:n ratkaiseva algoritmi toimii siis siten, että muodostetaan reduktioketju, ja pidetään kirjaa tapauksista, jotka ovat esiintyneet. Kun ensimmäisen kerran saadaan tapaus, joka on esiintynyt aikaisemmin, tarkistetaan, onko sillä yksikirjaiminen ratkaisu vai ei. Alkuperäisellä tapauksella on siis ratkaisu jos ja vain jos yksikirjaiminen ratkaisu löytyy.

Esimerkki 13. *Esimerkin 12 tapauksen (h', g') seuraaja on tapaus (h'', g''), missä*

	a	b	c
h''	a	bb	cb
g''	a	bbb	cb

jonka seuraaja taas on tapaus $(h^{(3)}, g^{(3)})$, missä

	a	b	c
$h^{(3)}$	a	bbb	c
$g^{(3)}$	a	bb	c

ja edelleen tämän tapauksen seuraaja on

	a	b	c
$h^{(4)}$	a	bb	c
$g^{(4)}$	a	bbb	c

Tämän tapauksen seuraaja taas on tapaus $(h^{(3)}, g^{(3)})$, joten lauseen 4 mukaan tapauksella on ratkaisu, koska kirjaimilla a ja c löytyy yksikirjaiminen ratkaisu.

Seuraaja–reduktio voidaan yhtälön (2) avulla ajatella myös ratkaisujen tiivistämiseksi. Lisäksi huomataan lauseen 4 avulla, että jos alkuperäiselle tapaukselle on olemassa ratkaisu, se tulee tiivistettyä yhden kirjaimen pituiseksi.

Lause 5. *Edellä käsitelty algoritmi toimii eksponentiaalisessa ajassa. Lisäksi voidaan osoittaa, että merkitty PCP on polynomitilassa ratkeava ongelma.*

5 Kahdella merkitty PCP

Voidaan myös osoittaa, että jos morfismien kaksikirjaimiset prefiksit ovat eri sanat, niin PCP on ratkeamaton (ks. [9]), joten ratkeavuuden ja ratkeamattomuuden välinen raja on yhdellä ja kahdella merkittyjen morfismien välissä. Huomaa, että kahdella merkitty ei takaa prefiksi-ominaisuutta tai edes injektiota, sillä myös yhden pituiset kuvat hyväksytään, esimerkiksi sanoilla a ja aa on eri kahden pituiset prefiksit. Esmerkiksi, jos $h(a) = a$, $h(b) = aa$ ja $h(c) = bb$, niin $h(aac) = aabb = h(bc)$, joten h on kahdella merkitty, mutta ei injektio.

6 Yleistetty PCP ja binäärinen PCP

Tarkastellaan seuraavaksi PCP:n muunnosta, ns. *yleistettyä* Postin vastaavuusongelmaa (engl. Generalized Post Correspondence Problem), lyhyesti GPCP. Erona alkuperäiseen ongelmaan ovat ns. *alku-* ja *loppusanat*.

Olkoot A ja B kaksi aakkostoa. GPCP:n tapaukset koostuvat morfismeista $h,g\colon A^* \to B^*$ ja sanoista $p_1,p_2,s_1,s_2 \in B^*$, ja siinä kysytään, onko vai eikö ole olemassa sellaista sanaa $w \in A^+$, että

$$p_1h(w)s_1 = p_2g(w)s_2.$$

Myös GPCP on ratkeamaton ongelma, koska kaikki PCP:n tapaukset ovat GPCP:n tapauksia.

GPCP:lle tunnetaan aakkoston koon suhteen samat ratkeavuus- ja ratkeamattomuustulokset kuin PCP:lle (ks. [1], [10]), ts. ratkeavuuden ja ratkeamattomuuden raja on jossain kahden ja seitsemän välillä. Sanojen pituuden suhteen raja on tarkalleen sama kuin PCP:lla (ks. [7]).

Merkityn PCP:n ratkaisualgoritmin idea toimii myös merkitylle GPCP:lle.

Lause 6. *[5] GPCP on ratkeava tapauksissa, joissa morfismit ovat merkittyjä.*

GPCP:ssa alkusanoille voidaan konstruoida palikat aivan kuten kirjainten kuvillekin. Loppusanojen käsittely on huomattavasti vaikeampaa, sillä reduktiossa muodostuu monta uutta loppusanaparia, mahdollisesti jopa äärettömän monta. Voidaan kuitenkin osoittaa, että riittää tarkastella äärellistä määrää uusia tapauksia, vieläpä tapauksia, joissa loppusanat ovat lyhyitä.

GPCP:n ratkeavuus merkityssä tapauksessa antaa myös uuden todistuksen binäärisen PCP:n ratkeavuudelle. Voidaan nimittäin osoittaa, että PCP tapauksissa, joissa $n = 2$, on ratkeava, jos merkitty GPCP on ratkeava tapauksissa, joissa $n = 2$ (ks. [1], [6]).

7 Ääretön PCP

Äärettömässä PCP:ssa tapaukset ovat kuten tavallisessa PCP:ssa, mutta kysytäänkin, että onko vai eikö ole olemassa sellaista ääretöntä sanaa, jolle morfismit *yhtyvät*. Sanotaan, että morfismit h ja g yhtyvät äärettömällä sanalla $\omega = a_1a_2\cdots$, jos $h(w) \bowtie g(w)$ kaikilla ω:n prefikseillä w.

Lause 7. *[16] Ääretön PCP on ratkeamaton. Erityisesti se on ratkeamaton tapauksissa, joissa morfismit ovat (bi)prefiksejä.*

Merkityn PCP:n ratkaisualgoritmin avulla voidaan kuitenkin osoittaa, että

Lause 8. *[4] Ääretön PCP on ratkeava tapauksissa, joissa morfismit ovat merkittyjä.*

Näin ollen äärettömän PCP:n ratkeavuuden ja ratkeamattomuuden välinen raja on myös prefiksien ja merkittyjen morfismien välillä.

Edellisen lauseen avulla voidaan lisäksi osoittaa, että

Lause 9. *[8] Ääretön PCP on ratkeava binäärisessa tapauksessa* $(n = 2)$.

Vaikka tapausten koko on pieni, tämä ongelma ei ole triviaali, sen osoittaa seuraava esimerkki.

Esimerkki 14. *Tarkastellaan tapausta*

	a	b
h	a	baa
g	aab	aa

Tällä tapauksella ei ole PCP:n ratkaisuja, ja voidaan osoittaa, että ääretön sana $\omega = a^2b^2a^4b^4a^8b^8\ldots$ *on sen ainoa ääretön ratkaisu. Huomaa, että* ω *laskee* 2*:n potensseja, joten se ei ole säännöllinen eli sitä ei voida määritellä äärellisellä automaatilla.*

8 Yhteenveto

Esitetään lopuksi tässä työssä mainitut PCP:n ratkeavuus- ja ratkeamattomuustulosten antamat ratkeavuuden ja ratkeamattomuuden väliset rajat yhtenä taulukkona:

RATKEAVA	RATKEAMATON
Koko $n \leq 2$	Koko $n \geq 7$
Pituus 1	Pituus ≥ 2
Merkitty	Prefiksi
Ääretön merkitty	Ääretön prefiksi

Kiitokset

Haluan kiittää Tietotekniikan Tutkimussäätiötä saamastani vuoden 2002 väitöskirjapalkinnosta ja väitöskirjatyöni ohjaajia, FT Tero Harjua ja professori Juhani Karhumäkeä tuesta ja yhteistyöstä. Lisäksi haluan kiittää professori Antti Valmaria monista hyödyllisistä ja selventävistä kommenteista tähän artikkeliin.

Viitteet

[1] A. Ehrenfeucht, J. Karhumäki ja G. Rozenberg, *The (generalized) Post Correspondence Problem with lists consisting of two words is decidable*, Theoret. Comput. Sci. **21** (1982), 119–144.

[2] V. Halava, *The Post Correspondence Problem for Marked Morphisms*, Väitöskirja, Matematiikan laitos, Turun yliopisto. TUCS Dissertations no. 37, 2002.

[3] V. Halava ja T. Harju, *Mortality in matrix semigroups*, Amer. Math. Monthly **108** (2001), no. 7, 649–653.

[4] V. Halava ja T. Harju, *Infinite solutions of the marked Post Correspondence Problem*, In *Formal and Natural Computing*, Lecture Notes in Comput. Sci., vol. 2300, Springer-Verlag, 2002, pp. 57–68.

[5] V. Halava, T. Harju ja M. Hirvensalo, *Generalized Post Correspondence Problem for marked morphisms*, Internat. J. Algebra Comput. **10** (2000), no. 6, 757–772.

[6] V. Halava, T. Harju ja M. Hirvensalo, *Binary (generalized) Post Correspondence Problem*, Theoret. Comput. Sci. **276** (2002), no. 1–2, 183–204.

[7] V. Halava, T. Harju, M. Hirvensalo ja J. Karhumäki, *(G)PCP for words of length at most two*, Tech. Report 463, TUCS, 2002.

[8] V. Halava, T. Harju ja J. Karhumäki, *Decidability of the binary infinite Post Correspondence Problem*, Discrete Appl. Math. **130** (2003), 521–526.

[9] V. Halava, M. Hirvensalo ja R. de Wolf, *Marked PCP is decidable*, Theoret. Comput. Sci. **255** (2001), no. 1-2, 193–204.

[10] T. Harju, J. Karhumäki ja D. Krob, *Remarks on generalized Post Correspondence Problem*, Proceedings of the 13th Annual Symp. on Theoretical Aspects of Computer Science, STACS'96, Lecture Notes in Comput. Sci., vol. 1046, Springer-Verlag, 1996, pp. 39–48.

[11] Y. Lecerf, *Récursive insolubilité de l'équation générale de diagonalisation de deux monomorphisms de monoïdes libres*, Computes Rendus Acad. Sci. Paris **257** (1963), 2940–2943.

[12] Y. Matiyasevich, *Hilbert's Tenth Problem*, Foundations of Computing Series, MIT Press, Cambridge, MA, 1993.

[13] Y. Matiyasevich ja G. Sénizergues, *Decision problems for semi-Thue systems with a few rules*, Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science (New Brunswick, New Jersey), IEEE Computer Society Press, 27–30 July 1996, pp. 523–531.

[14] M. S. Paterson, *Unsolvability in* 3×3 *matrices*, Studies in Applied Mathematics **49** (1970), 105–107.

[15] E. Post, *A variant of a recursively unsolvable problem*, Bull. of Amer. Math. Soc. **52** (1946), 264–268.

[16] K. Ruohonen, *Reversible machines and Post's Correspondence Problem for biprefix morphisms*, Elektron. Informationsverarb. Kybernet. (EIK) **21** (1985), no. 12, 579–595.

[17] A. Salomaa, *Formal Languages*, Academic Press, 1973.

[18] H. Stamer, *PCP @ Home*, `http://www.informatik.uni-leipzig.de/~pcp/pcpcontest_en.html` (haettu 6.11. 2003).