Churchin lause RA-kielelle

Juho Viitasalo

Matematiikan pro-gradu -tutkielma

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
Syksy 2007

Sisältö

1 Johdanto

Tämä pro gradu tutkielma käsittelee yhdysvaltalaisen loogikko Alonzo Churchin vuonna 1936 todistamaa metamatemaattista teoreemaa, joka tunnetaan Churchin lauseena. Lause kuuluu matemaattisen logiikan alaan. Sillä on myös yhtymäkohtia yleiseen formaaleita kieliä käsittelevään teoriaan sekä tietotekniikan teoriaan [10]. Lukijan pohjatiedoiksi oletetaan matematiikan ja joukko-opin peruskäsitteistön sekä merkintöjen tunteminen, vaikkakin asia on pyritty tietoisesti esittämään mahdollisimman kansantajuisesti.

Churchin lause RA-kielelle sanoo: Jos RA-kieli on ω-ristiriidaton, niin RA-kielen teoreemojen Gödel-lukujen joukko ($\mathcal{TR}$) ei ole rekursiivinen. Tämä on Robbinin muotoilu [19, s.126] ja se poikkeaa Churchin ja esimerkiksi Kleenen [13, s.301] muotoiluista. Robbinin muotoilu on parempi siitä syystä, että se pitää erillään funktion rekursiivisuuden ja sen laskettavuuden mikä on tärkeä seikka. Kyseinen ero ei korostu kaikissa käytetyissä lähteissä. Käsittelen tätä eroa kappaleessa 8.

Rajoitan tarkastelun vain Churchin lauseeseen RA-kielelle (*recursive arithmetic*). RA-kieli on lukuteoriaa imitoiva formaali kieli, jonka ensimmäinen auki kirjoitettu muotoilu on kenties Hilbertin ja Ackermannin teoksessa *Grundzüge der theoretischen Logik* 1928, johon Church itsekin viittaa lausetta todistaessaan. Hän käyttää RA-kielestä nimitystä *engerer Funktionenkalkül* [2].

Mitään yhtenäistä esitystä tai viitettä RA-kielen historiasta en kyennyt löytämään, joten on vaikea sanoa, onko Robbinin [19, s.66-79] ja Kuritun [16, s.26-69] käyttämässä RA-kielessä eroja Churchin käyttämään. RA-kielen historia olisi hyvä erillisen tutkimuksen aihe. Karkea jäsentely RA-kielen vaiheista löytyy Churchin kirjasta *Introduction to Mathematical Logic* [4, s.62-63].

Churchin lausetta ei tule sekoittaa Churchin *teesiin*, joka tunnetaan myös Church-Turing teesinä. Se voidaan ilmaista seuraavasti: Funktio on tehokkaasti laskettavissa, jos ja vain jos se on rekursiivinen. Churchin teesi ei ole matemaattislooginen teoreema vaan hypoteesi laskettavuuden luonteesta. Tosin sitä voidaan tukea useilla argumenteilla. Laskettavuudelle voidaan esittää myös useita muita määritelmiä, jotka ovat yhteneviä toistensa kanssa. Esimerkiksi Turingin koneilla laskettavat funktiot yhtenevät rekursiivisten funktioiden kanssa. Paneudun asiaan enemmän kappaleessa 7.

Churchin esitti todistuksen lauseelleen kahdessa osassa. Artikkelin *An Unsolvable Problem of Number Theory* [3] lopussa hän toteaa, että *Principia Mathematican* [23] mukaiselle systeemille[1], joka on ω-ristiriidaton, ei ole Hilbertin etsimää yleistä ratkaisumenetelmää, jonka avulla voidaan selvittää onko jokin kaava teoreema vai ei (*Entscheidungsproblem*).

> *"... if the system of Principia Mathematica be ω-consistent, its Entscheidungsproblem is unsolvable."*

Tulos on oikeammin johtopäätös, joka nojautuu kahteen asiaan. Ensinnäkin siihen, ettei ole olemassa etsittyä ratkaisumenetelmää, joka olisi rekursiivinen[2] ja toiseksi siihen, että Churchin teesi pätee.

Toisessa artikkelissa *A Note on the Entscheidungsproblem* [2] ja sen korjauksessa *Correction to a Note on the Entscheidungsproblem* [1] hän osoitti RA-kielen (*Funktionenkalkül*) olevan erikoistapaus niistä loogisista systeemeistä, joilla väite pätee.

> *"The general case of the Entscheidungsproblem of the engere Funktionenkalkül is unsolvable."*

Tämänkin muotoilun kohdalla on hyvä huomata, että se soveltaa Churchin teesiä. Jos siis Church-Turing teesi pitää paikkansa, niin siitä ja Churchin lauseesta seuraa, että ei ole olemassa laskennallista menetelmää, jonka avulla voidaan ratkaista kaikki lukuteorian ongelmat.

[1]Systeemin, joka on tarpeeksi voimakas mahdollistaakseen tietyt verrattain yksinkertaiset määrittelemis- ja todistusmenetelmät [3, s.363]. RA-kieli ei siis ole ainoa formaali systeemi, jossa Churchin lause pätee [19, s.127][22, s.126].

[2]Tämä on "yleinen" Churchin lause. Se ilmaistaan artikkelissa [3] epäsuorasti, mikä johtuu siitä, että Church käsittelee Kleenen kanssa kehittelemänsä λ-laskennan ja rekursiivisten funktioiden yhteyttä, mikä ei olisi tavoitteen kannalta välttämätöntä [3, s.346 alaviite 3]. Tämä yhteys kuitenkin tukee Churchin teesiä ja on siitä syystä merkittävä.

2 Esittely

2.1 Taustaa

Mihin matemaatikkoja enää tarvitaan, kun käytettävissämme on jo tehokkaita tietokoneita? Tietokoneet osaavat tunnetusti suorittaa kaikki laskutoimitukset, jotka matemaatikkokin osaisi ja vielä monimutkaisempiakin. Lisäksi tietokone tekee laskutoimitukset nopeammin ja tarkemmin. Tämän kaltaisia kysymyksiä kysyttäessä ei ymmärretä matemaatikon toimenkuvaa oikein, vaan nähdään se pelkkänä laskentona, jota on harjoiteltu peruskoulussa. Tällaista laskentoa koulussa suorittavalle oppilaalle voikin tulla vähän väliä mieleen kysymys: Miksi ei saa käyttää laskinta?

Jotta voisi ymmärtää, mitä laskin tekee, on kuitenkin ensin ymmärrettävä mitä luvut ovat ja miten ne toimivat. Toiseksi laskin ei osaa kertoa mitä laskea ja miten, kun matematiikkaa sovelletaan käytäntöön. Tässä tarvitaan ihmisen hahmotus- ja ongelmanratkaisukykyä, sekä ennen kaikkea kykyä muotoilla ongelmia matemaattisesti käsiteltävään muotoon.

Tosin tällainen käytännön soveltaminenkaan ei ole matemaatikon varsinaista alaa. Matematiikassa kyse on ensisijaisesti matemaattisten tai loogisten lainalaisuuksien tutkimisesta ja matemaattisen teorian muodostamisesta. Yksi yleinen matematiikan tutkimisen kohde on esimerkiksi, mitä tulee olettaa, että jotkin yhtälöt pitävät paikkansa ja miten näistä oletuksista voidaan päätellä näiden yhtälöiden pätevyys. Otetaan yksinkertaisena esimerkkinä Goldbachin hypoteesi, joka esitettiin jo vuonna 1742, mutta jolle ei ole vieläkään löydetty ratkaisua useista yrityksistä huolimatta.

Jokainen kahta suurempi parillinen luku voidaan esittää kahden alkuluvun[3] summana.

Nyt voidaan laskea. Alkulukuja ovat $2, 3, 5, 7, 11, 13$ jne. Kokeillaan ensin lukua 4, joka on pienin kahta suurempi parillinen luku. $4 = 2 + 2$, joten

[3] Alkuluku on luonnollinen luku > 1, joka on jaollinen vain 1:llä tai itsellään.

hypoteesi pitää paikkansa luvulla neljä. Jos jatketaan laskemista seuraavilla luvuilla, voidaan huomata, että hypoteesi saa tukea. $6 = 3 + 3$, $8 = 3 + 5$, $10 = 5 + 5$ ja $12 = 5 + 7$.

Laskemista ei kannata jatkaa käsin, sillä voidaan kirjoittaa tietokoneohjelma, joka etsii parillisille luvuille näitä alkulukupareja. Vuoteen 2007 mennessä tietokoneet ovat laskeneet näitä alkulukupareja lukuun $5 \cdot 10^{17}$ asti (`http://www.ieeta.pt/~tos/goldbach.html`, Goldbach conjecture verification). Vastaesimerkkiä ei ole löytynyt.

Vastaesimerkin puute ei kuitenkaan riitä todistamaan hypoteesin paikkansapitävyyttä *kaikilla* luonnollisilla luvuilla. Tämä on lukuteoreettinen ongelma. Ei ole oikein sanoa väitteen olevan tosi sillä perusteella, että sitä on jo testattu suurella määrällä lukuja. Lukuja jää kuitenkin aina äärettömän monta testaamatta. Kysymykseen, ”Miksi ei saa käyttää laskinta?”, voi siis vastata: ”Saahan sitä käyttää, mutta se ei kykene ainakaan todistamaan lukuteoreettisia väitteitä.”

Merkittävä ongelma on myös, että kysymys ei ole pelkästään lukuteoreettisen lauseen ”totuudesta”, vaan myös siitä, mitä tarkoitetaan luonnollisilla luvuilla. Tämä ei ole niin selvää kuin voisi luulla. Normaalisti laskutoimituksia tehdessä sovelletaan paperilla näkyviin symbolijonoihin (esim. ”$1+1 = 2$” tai ”$2 - 3 = 0$”) jonkinlaista intuitiivista semantiikkaa. Toisin sanoen niille annetaan jonkinlainen merkitys eli semantiikka, jonka perusteella sanotaan, onko jokin lause tosi vai ei. Toisaalta ei voida suoraan sanoa, onko tämä intuitio ristiriidaton. Voisiko esimerkiksi kahdesta intuitiivisesti todesta lauseesta suoraan päätellä ristiriitaisen tuloksen.

Toinen lähestymistapa on käsitellä lukuteorian lauseita ainoastaan syntaksin tasolla. Toisin sanoen tutkitaan ainoastaan sitä onko symbolijono oikein muodostettu. Tätä varten lukuteoria täytyy formalisoida.

Lukuteorian formalisointi edellyttää seuraavia asioita. Muodostetaan formaali eli muodollinen kieli, jossa on vain äärellinen määrä erilaisia symboleita ja tarkat symbolijonojen muodostussäännöt. Hyvin muodostettua symbolijonoa kutsutaan kaavaksi. Kieleen tulee liittää myös päättelysäännöt ja aksioomat. Aksioomat ovat joukko kaavoja, jotka täyttävät seuraavat ehdot:

1. Aksioomaa ei voi päätellä toisista aksioomasta käsin päättelysääntöjen avulla.

2. Aksioomista ei voi päätellä keskenään ristiriitaisia kaavoja (A ja $\neg A$).

Aksioomat ovat päättelyn lähtökohtia, joita ei voi johtaa muista kaavoista. Ne ovat ikäänkuin siemeniä, josta koko looginen järjestelmä kasvaa ja haaroittuu päättelysääntöjen kautta. Tämän järjestelmän ristiriidattomuus riippuu aksioomien valinnasta. Jos aksioomat valittaisiin siten, että niistä voitaisiin päätellä ristiriita, niin tässä aksioomajärjestelmässä voitaisiin todistaa mikä tahansa väite.

Aksioomista johdettuja kaavoja kutsutaan teoreemoiksi. Uusien teoreemojen päättely on pelkkä jono kaavoja, jotka on johdettu toisistaan päättelysääntöjen avulla. Missään vaiheessa ei tarvitse kysyä, mitä nämä kaavat merkitsevät. Formaalissa kielessä tehtävän päättelyn säännöt ja lähtökohdat ovat niin selvät, että jokaisesta päättelyvaiheesta voidaan tunnistaa, mitä päättelysääntöä ja mitä aksioomia on käytetty. Päättely on niin yksiselitteistä, että sen pätevyyden voi tarkistaa mekaanisesti. Näin intuitiolle, joka voi vaihdella ihmisestä riippuen, ei anneta tilaa.

Jotta lukuteoria voidaan katsoa formalisoiduksi, tulee siis valita äärellinen joukko symboleita ja kaavojen muodostussääntöjä siten, että niiden avulla voi ilmaista minkä tahansa lukuteorian väitteen, mutta myös siten, että mikään intuitiivisesti ”mieletön” symbolijono ei ole kaava. Esimerkiksi merkkijonoa $6 = 4 + - + 1x = 2$ vastaavaa formaalin kielen symbolijonoa ei tulisi kelpuuttaa kaavaksi, koska se ei merkitse mitään. Sen lisäksi tulee löytää sopivat päättelysäännöt, sekä aksioomat, joista voi päätellä ainoastaan intuitiivisesti tosia lukuteorian väitteitä. Näiden aksioomien tulisi tietysti olla ristiriidattomat, sillä kieli, jossa voi päätellä intuitiivisesti epätosia väitteitä (esim. $0 * 1 = 2$), ei ole käyttökelpoinen. Jos formaali kieli täyttää nämä tavoitteet sanotaan, että se *imitoi* intuitiivista lukuteoriaa. Sanomme tällaista formalismia *lukuteorian formalismiksi.*

Jos halutaan siis tarkasti tietää, mitä oletuksia ja päättelyaskelia tarvitaan Goldbachin hypoteesin kaltaisen ongelman ratkaisemiseen tarvitaan, niin se tulisi esittää jossakin lukuteorian formalismissa. Sitten tulee löytää päättelyjono, joka lähtee liikkeelle aksioomista ja päätyy haluttuun kaavaan, joka vastaa hypoteesia. Jos päättelyjono voidaan muodostaa, niin kyseessä ei ole enää hypoteesi vaan teoreema, mutta tätä siis emme tiedä, koska päättelyjonoa ei ole vielä löytynyt.

Kysymys lukuteorian formalisoinnista on merkittävä siitäkin syystä, että se tekee mahdolliseksi matematiikan itsensä ja siihen liittyvän päättelyn tutkimisen eksaktin matemaattisesti. Tätä kutsutaan metamatematiikaksi [13, s.59]. Lisäksi, jos lukuteorian formalisointi on mahdollista, niin tämä formalisointi voidaan laajentaa koskemaan muitakin matematiikan tutkimushaaroja, sillä luonnollisten lukujen voidaan ajatella olevan pohjana kaikelle muulle matematiikalle [5, s.63]. Lukuteorian formalismin esikuvana ja lähtökohtana on käytetty predikaattilogiikkaa (ks. [15]), jossa syntaksi ja semantiikka ovat irrotettavissa toisistaan.

Eräs ehdokas lukuteorian formalisoinniksi on RA-kieli. Hyvä esitys siitä on Joel Robbinin kirjassa Mathematical Logic: First Course [19, Luku 3] ja vielä perusteellisempi Lassi Kuritun monisteessa Matemaattinen logiikka [16, Luku 2]. RA-kielessä on 17 aksioomaa ja predikaattilogiikan tavalliset päättelysäännöt (modus ponens ja kvantifiointi). Toinen yleisesti käytetty aksioomajoukko on Peanon aksioomat [14, s.284][22, s.116].

RA-kielen on siis tarkoitus imitoida luonnollisten lukujen ominaisuuksia. Tosin se pystyy imitoimaan ainoastaan primitiivirekursiivisia[4] laskutoimituksia. Primitiivirekursiiviset funktiot ovat vain osa kaikista luonnollisten lukujen funktioista [16, s.iii][22, s.43]. Tämä on yksi syy, miksi kyseinen teoria ei kykene täysin sisältämään käsitystämme luvuista ja niiden käytöstä.

On toinenkin syy, miksi RA-kieli ei vastaa intuitiivista käsitystä luonnollisista luvuista eikä edes niiden primitiivirekursiivisista ominaisuuksista. Tämä liittyy Gödelin ensimmäiseen epätäydellisyyslauseeseen. Sen nojalla on olemassa validi eli paikkansa pitävä lukuja koskeva väite, joka ei kuitenkaan ole teoreema RA-kielessä [16, Lause 3.30][19, s.115]. Tämä pitää paikkansa kaikille formaaleille systeemeille, jotka ovat riittävän monimutkaisia imitoimaan luonnollisten lukujen primitiivirekursiivisia laskutoimituksia. Ei siis ole mahdollista muodostaa lukuteoriaa täydellisesti imitoivaa kieltä. Mer-

[4]Primitiivirekursiivisuuden määritelmä ks. luku 3

kittävimmät yritykset ovat RA-kielen lisäksi olleet Peanon aritmetiikka ja *Principia Mathematican* formaali kieli.

Gödelin todistus perustuu valehtelijan paradoksia muistuttavaan ideaan (”tämä lause on epätosi”)[19, s.111]. Hän rakentaa RA-kielen kaavan, joka väittää, ettei kyseinen kaava ole teoreema. Gödelin mullistava havainto olikin juuri siinä, että RA-kieli on riittävän ilmaisuvoimainen viitatakseen itseensä. Tämä tapahtuu Gödel-numeroinnin kautta, jossa jokaista kaavaa vastaa yksiselitteinen luonnollinen luku. Churchin lauseen todistus RA-kielelle perustuu myös Gödel-numeroinnille ja samantapaisen kaavan muotoilemiseen kuin Gödelin ensimmäisessä epätäydellisyyslausekin.

2.2 Esityksen rakenne

Tässä pro gradu-tutkielmassa esitetään Churchin lauseen todistus kokonaisuudessaan sekä kaikki sen todistamisessa käytetyt käsitteet ja lauseet. Tämän jälkeen esitetään läpileikkaus lauseen historiallisista yhtymäkohdista. Churchin lausetta tarkastellaan myös Church-Turing teesin valossa, mikä taas edellyttää katsausta laskemisen ja tietotekniikan teoriaan Turingin koneiden kautta. Tästä löytyy selvyyttä mainittuihin mekaanisten laskimien rajoitteisiin ratkaistaessa lukuteorian ongelmia.

2.3 Keskeiset käsitteet

Hyvin keskeisessä asemassa Churchin lauseen todistusta on tietysti RA-kieli, jota todistus koskee. RA-kieleen liittyvät olennaisesti primitiivirekursiiviset funktiot. Primitiivirekursiivisuus on alakäsite, joka on erityistapaus rekursiivisuudesta. Primitiivirekursiivisia laskutoimituksia ovat esim. yhteen ja kertolasku sekä niiden johdannaiset. Tarkemmin primitiivirekursiivisiin funktioihin on paneuduttu esim. Kuritun monisteessa [16], Robbinin kirjassa [19, s.66] tai Kleenen kirjassa *Introduction to Metamathematics* [13, s.217].

Tämän jälkeen esitellään μ-rekursiiviset funktiot ja vertaillaan niitä primitiivirekursiivisten funktioiden kanssa. μ-rekursiivisuus on väljempi ehto kuin primitiivirekursiivisuus, joten jokainen primitiivirekursiivinen funktio on μ-rekursiivinen. Primitiivirekursiivisen ja μ-rekursiivisen funktion ratkaiseva ero on siinä, että μ-rekursiivisen funktion määritelyssä sallitaan rajoittamaton minimalisaatio. Rajoitetun ja rajoittamattoman minimalisaation eroa käsitellään myös. μ-rekursiivisuutta käsittelevät Kleene [13], Odifreddi[17] ja Väänänen [22].

Church käytti itse funktion rekursiivisuuden määritelmää, joka perustuu funktion johdettavuuteen joukosta yhtälöitä [3][12]. Määritelmä on liitetty (ks. liite 9.2) mukaan, sillä se antaa täydellisemmän kuvan siitä, millaisia rekursiiviset funktiot voivat olla sekä hiukan historiallista perspektiiviä. On todistettu, että rekursiivisten ja μ-rekursiivisten funktioiden määritelmät muodostavat saman funktioluokan (ks. lause 3.22).

Ongelmana oli tutkimusta tehtäessä, ettei Robbin määrittele lainkaan käsitettä rekursiivinen funktio. Hänen todistuksessaan Churchin lauseelle viitataan käsitteeseen rekursiivinen joukko, jossa ei tarvita lainkaan rekursiivisen funktion käsitettä. Jos kuitenkin halutaan tulkita lausetta Churchin teesin valossa, niin Robbinin rekursiivisten joukkojen ja Churchin rekursiivisten funktioiden välille tulee löytää yhteys. Tämä yhteys esitellään lauseessa 3.34.

Todistuksessa keskeinen työkalu on myös Gödel-numerointi, joka esitellään kappaleessa 4.3.

2.4 Viitteet

Churchin lauseen todistamisessa seurataan ensisijaisesti Robbinin kirjaa [19], mutta tukena käytetään myös Kuritun monistetta [16] etenkin RA-kielen osalta. Tämä pro gradu täydentää osaltaan Kuritun monistetta, sillä siinä Churchin lause mainitaan, mutta todistus sivuutetaan. Kleenen teksteihin[13] ja [12] viitataan usein, sillä hän on kirjoittanut kattavasti funktioiden rekursiivisudesta ja metamatemaattisesta tutkimuksesta. Kleenen rinnalla viitataan myös Piergiorgio Odifreddin teoksiin[17] ja [18], koska ne ovat uusinta rekursiivisuutta kattavasti käsittelevää kirjallisuutta ja nykyaikaisen lukijan voi olla helpompi ymmärtää hänen käyttämiään merkintätapoja.

2.5 Merkintätavat

Tutkielmassa on käytetty paljon melko vanhoja lähteitä vuosilta 1910-1969. Noihin aikoihin matemaattisen logiikan ja erityisesti metamatematiikan teoria oli uutta ja merkintätavat ovat voineet vaihdella paljonkin kirjoittajasta riippuen.

Tämän tutkielman tekijä on pyrkinyt käyttämään johdonmukaisesti moderneja logiikan ja joukko-opin merkintöjä. RA-kielen syntaksin ja semantiikan merkinnät on omaksuttu yhdistellen Kuritulta ja Robbinilta. Turingin koneita koskevat merkinnät vaihtelevat myös lähteittäin. Tässä tutkielmassa käytettävä Turingin koneiden formalismi on itse muotoiltu.

3 Rekursiivisuus

3.1 Rekursio

Rekursio merkitsee palautumista tai samanlaisena toistumista[5]. Käsitettä käytetään muun muassa matemaattisten ilmiöiden ja funktioiden määrittelemisessä[6], tietynlaisten ongelmien kuvailemisessa ja ratkaisemisessa sekä tietokoneohjelmoinnissa itseään kutsuvien funktioiden yhteydessä. Rekursiivisuuden käsite on myös keskeinen yleisessä formaaleja kieliä ja automaatiota käsittelevässä teorissa, joka on pohjana tietotekniikan teorialle [10].

Ongelmien ratkaisemisessa rekursiolla tarkoitetaan ongelman palautumista toiseen samanlaiseen ongelmaan (rekursioaskel). Jos ongelma on periaatteessa ratkaistavissa äärellisellä määrällä rekursioaskelia, niin se on rekursiivinen.

Arkipäiväisen esimerkin voi ottaa vaikka sukupuusta. Jos halutaan selvittää, onko joku henkilö X Sippolan Joosepin jälkeläinen, niin tulee selvittää, onko toinen hänen vanhemmistaan Sippolan Jooseppi. Jos on, niin lopetetaan. Jos ei ole, niin selvitetään onko kumpikaan vanhemmista Joosepin jälkeläinen. Tämä tehdään samalla tavalla kuin ensimmäisenkin henkilön tapauksessa[7]. Jos oletetaan, että voidaan aina saada selville kunkin ihmisen vanhemmat rajallisessa ajassa, niin päädytään lopulta vastaukseen, koska sukupolvia on ollut vain äärellinen määrä. Vastaus kysymykseen voi olla joko kyllä tai ei.

Mitä useampi sukupolvi taaksepäin joudutaan tutkimaan, sitä suuremmaksi kunkin rekursioaskeleen työmäärä kasvaa. Jos tutkittavana on n:s sukupolvi taaksepäin, niin tehtävien jälkeläisyystutkimusten määrä on 2^n. Jos X ei ole Joosepin jälkeläinen, joudutaan tutkimaan jokaisen esivanhemman jälkeläisyys ensimmäiseen nimiä kantaneeseen ihmissukupolveen asti. Tämä on tietysti hyvin työlästä.

---

[5] Englanniksi "*recur*" = toistua, uusiutua, palata

[6] rekursioperiaate [15, s.9-10]

[7] Eli selvitetään onko kumpikaan vanhemman vanhempi Sippolan Jooseppi. Jos on, niin lopetetaan. Jos ei ole, niin selvitetään on onko yksikään vanhempien vanhemmista Joosepin jälkeläinen, joka tehdään samalla tavalle jne...

Jos tiedetään Sippolan Joosepin eläneen korkeintaan k:n sukupolven päässä X:stä, niin tutkimus helpottuu. Jos nimittäin Jooseppi ei löydy k:n rekursioaskelen jälkeen, niin tutkimus voidaan lopettaa. Tällöin voitaisiin esittää jo etukäteen kattoarvio tutkimukseen kuluvasta ajasta.

Voidaan kuvitella tilanne, jossa sukupolvia olisikin ääretön määrä ja oletetaan ettemme tietäisi mitään kattoarviota sille, kuinka monen sukupolven päässä Jooseppi korkeintaan on. Jos X ei olisi Joosepin jälkeläinen, niin selvitysprosessi olisi tässä tapauksessa päättymätön. Jos taas X olisi Joosepin jälkeläinen, niin prosessi päättyisi, mutta sen kestoa olisi mahdotonta arvioida etukäteen.

Nämä erimerkit tarjoavat karkeita analogioita erilaisille rekursiivisille ilmiöille matemaattisten funktioiden maailmassa.

3.2 Funktion primitiivirekursiivisuus

Luonnollisia lukuja koskevat merkinnät määritellään seuraavasti.

Määritelmä 3.1
Olkoon $\mathbb{N}$ *luonnollisten lukujen joukko eli* $\mathbb{N} = \{0, 1, 2, \ldots\}$. *Merkitään*

$$\mathbb{N}^k = \{(n_1, \ldots, n_k) \mid n_i \in \mathbb{N} \textit{ kaikille } i = 1, \ldots, k\},$$

missä $k \in \mathbb{N}$, $k \geq 1$. *Erityisesti* $\mathbb{N}^1 = \mathbb{N}$ *ja sovitaan, että* $\mathbb{N}^0 = \{0\}$. $\boxed{\mathbb{N}}$

Ennen primitiivirekursiivisuuden määritelmää meillä tulee olla joitakin funktiotyyppejä, jotka toimivat rakenneosina primitiivirekursiivisissa (*p.r.*) funktioissa. $\boxed{\textit{p.r.}}$

Määritelmä 3.2 (Primitiivikuvaukset)
Primitiivikuvaksia ovat:

1. *Nollakuvaus* $\mathrm{Z} : \mathbb{N} \to \mathbb{N}$, *missä* $\mathrm{Z}(x) = 0$ *kaikille* $x \in \mathbb{N}$.

2. *Seuraajakuvaus* $\mathrm{S} : \mathbb{N} \to \mathbb{N}$, *missä* $\mathbf{s}(x) = x + 1$ *kaikille* $x \in \mathbb{N}$.

3. *Projektiokuvaus* $\mathrm{P}_i^k : \mathbb{N}^k \to \mathbb{N}$, *missä* $\mathrm{P}_i^k(x_1, \ldots, x_k) = x_i$ *kaikille* $(x_1, \ldots, x_k) \in \mathbb{N}^k$ *ja* $1 \leq i \leq k$.

Lisäksi määritellään kaksi kuvaustyyppiä, jotka koostuvat toisista kuvauksista.

4. *Yhdistekuvaus* $\text{Y}(f; g_1, \ldots, g_m) : \mathbb{N}^k \to \mathbb{N}$ *muodostuu kuvauksista* $f : \mathbb{N}^m \to \mathbb{N}$ *sekä* $g_i : \mathbb{N}^k \to \mathbb{N}$*, missä* $k \geq 0$*,* $m \geq 1$ *ja* $i = 1, \ldots, m$ *seuraavasti.*

   $$\text{Y}(f; g_1, \ldots, g_m)(x_1, \ldots, x_k) = f(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k))$$

   kaikille $(x_1, \ldots, x_k) \in \mathbb{N}^k$*.*

5. *Primitiivirekursiolla muodostettu kuvaus: Olkoon* $k \in \mathbb{N}$*,* $k \geq 0$ *ja* $g : \mathbb{N}^k \to \mathbb{N}$ *sekä* $h : \mathbb{N}^{k+2} \to \mathbb{N}$ *kuvauksia. Kuvaus* $f : \mathbb{N}^{k+1} \to \mathbb{N}$ *on saatu* ***primitiivirekursiolla*** *kuvauksista* g *ja* h*, jos*

   (a) $f(x_1, \ldots, x_k, 0) = g(x_1, \ldots, x_k)$ *kaikille* $(x_1, \ldots, x_k) \in \mathbb{N}^k$ *ja*

   (b) $f(x_1, \ldots, x_k, y + 1) = h(x_1, \ldots, x_k, y, f(x_1, \ldots, x_k, y))$
   kaikille $(x_1, \ldots, x_k, y) \in \mathbb{N}^{k+1}$*.*

   Tällöin merkitsemme $f = \text{R}(g, h)$*.* $\boxed{\text{R}(g, h)}$

Huomautus 3.3 *Huomattavaa on, että näiden funktioiden määrittelyjoukko on koko* $\mathbb{N}^k$ *jollakin* $k \geq 0$ *ja arvojoukko on* $\mathbb{N}$*. Tällaisia funktioita kutsutaan numeerisiksi [19, s.66]. Erityisesti tulee huomata, että funktion* $g : \mathbb{N}^0 \to \mathbb{N}$ *lähtöjoukko on* $\{0\}$*, joten se voidaan määritellä ainoastaan vakiofunktioksi.*

Määritelmä 3.4 (Primitiirekursiiviset funktiot)

1. Z *on p.r.*

2. S *on p.r.*

3. *Funktiot* P_i^k*, missä* $k \in \mathbb{N}$ *ja* $1 \leq i \leq k$*, ovat p.r.*

4. *Jos* $h : \mathbb{N}^m \to \mathbb{N}$ *ja* $g_i : \mathbb{N}^k \to \mathbb{N}$*, missä* $i = 1, \ldots, m$*, ovat p.r. , niin* $f : \mathbb{N}^k \to \mathbb{N}$*,* $f = \text{Y}(h; g_1, \ldots, g_m)$ *on p.r.*

5. *Jos* $g : \mathbb{N}^k \to \mathbb{N}$ *ja* $h : \mathbb{N}^{k+2} \to \mathbb{N}$*, missä* $k \geq 0$*, ovat p.r. , niin* $f : \mathbb{N}^{k+1} \to \mathbb{N}$*,* $f = \text{R}(g, h)$ *on p.r.*

6. *Sanotaan, että funktio on primitiivirekursiivinen vain, jos se on sitä ehtojen 1-5 nojalla.*

Huomautus 3.5 *Yllä esitetty määritelmä on epätarkka, koska se on tietyssä mielessä kehämäinen. Tarkan määritelmän mukaan primitiivirekursiivisten funktioiden joukko on pienin funktiojoukko, joka sisältää primitiivikuvaukset 1-3 ja on suljettu yhdistämisen ja primitiivirekursion suhteen. Tarkka määritelmä löytyy Kuritun monisteesta [16, s.3].*

Huomautus 3.6 *Kaikki primitiivirekursiiviset funktiot ovat numeerisia.*

Kolme ensimmäistä kuvausta ovat siis sellaisia, joihin lopulta tulisi päätyä, kun selvitetään onko funktio *p.r.* . Funktion primitiivirekursiivisuuden selvittäminen on rekursiivinen toimitus. Tämä vertautuu sukupuuesimerkkiin, jossa tiedämme kuinka kuinka monen sukupolven päässä Jooseppi korkeintaan on. Ero on siinä, että auki kirjoitetusta *p.r.* funktiosta voi nähdä suoraan kuinka monta rekursioaskelta on korkeintaan otettava, jotta saataisiin selville onko se *p.r.* .

Esimerkiksi summafunktio $f : \mathbb{N}^2 \to \mathbb{N}$, $f(x, y) = x + y$ voidaan kirjoittaa muotoon $f(x, y) = \text{R}(\text{P}^1_1(x), \text{Y}(\text{S}; \text{P}^3_3(x, y, z)))$ [16, s.4]. Jos halutaan selvittää, onko funktio *p.r.* , niin tarvittavien rekursioaskelien määrä saadaan symbolien Z, S, P, Y ja R yhteismäärästä. Tarkastus etenee summafunktion tapauksessa seuraavasti:

1. Onko $\text{R}(\text{P}^1_1(x), \text{Y}(\text{S}; \text{P}^3_3(x, y, z)))^{(1)}$ *p.r.* ? On, jos $\text{P}^1_1(x)^{(2)}$ ja $\text{Y}(\text{S}; \text{P}^3_3(x, y, z))^{(3)}$ ovat *p.r.* .

2. Onko (2) *p.r.* ? On.

3. Onko (3) *p.r.* ? On, jos S ja P^3_3 ovat *p.r.* .

4. Onko S *p.r.* ? On.

5. Onko P^3_3 *p.r.* ? On.

Siis (1) on *p.r.* . Tietysti täytyy myös tarkastaa, että kuvausten lähtöjoukkojen dimensiot ovat määritelmän 3.4 mukaisia. Tässä tapauksessa näin on.

Primitiivirekursiiviset funktiot ovat Väänäsen mukaan ”karkeasti” sellaisia, joille voidaan tehdä arvio niiden laskemiseen kuluvasta ajasta [22, s.43,47]. Tämä laskemiseen kuluva aika on riippuvainen itse funktiosta, mutta myös siitä pisteestä, missä funktion arvo lasketaan. Tätä käsittelee tarkemmin Odifreddi kirjassaan *Classical Recursion Theory Volume II* [18, s.297]. Tosin hänkään ei kerro mitä tarkoittaa sana ”karkeasti” tässä yhteydessä.

Määritelmä 3.7 (Primitiivirekursiiviset joukot)
Sanotaan, että joukko $A \subset \mathbb{N}^k$ on primitiivirekursiivinen (merkitään: $A \in \mathcal{PR}$), jos on olemassa primitiivirekursiivinen funktio $\chi_A : \mathbb{N}^k \to \{0,1\}$ siten, että $\boxed{\mathcal{PR}}$

$$\chi_A(x) = \begin{cases} 1 & \text{kun } x \in A \\ 0 & \text{kun } x \notin A \end{cases},$$

eli jos A:n karakteristinen funktio on primitiivirekursiivinen.

3.3 Primitiivirekursiivisuuteen liittyviä lauseita

Lause 3.8 *Olkoon $A, B \subset \mathbb{N}^k$ primitiivirekursiivisia joukkoja, missä $k \in \mathbb{N}$ ja $k \geq 1$. Tällöin myös $\mathbb{N}^k \backslash A$, $A \cap B$, $A \cup B$ ja $A \backslash B$ ovat primitiivirekursiivisia joukkoja.*

Todistus:
Ks. [16, Lause 1.10]. $\square$

Määritelmä 3.9 (Rajoitettu vähennyslasku)
Olkoon $f : \mathbb{N}^2 \to \mathbb{N}$ funktio siten, että

$$f(x,y) = \begin{cases} x - y & \text{jos } x \geq y \\ 0 & \text{jos } x < y \end{cases}.$$

Sanotaan, että f on rajoitettu vähennyslasku. Merkitään sitä seuraavasti $\boxed{\dot{-}}$

$$f(x,y) = x \dot{-} y.$$

Lause 3.10 *Seuraavat funktiot ovat primitiivirekursiivisia:*

- $f : \mathbb{N}^2 \to \mathbb{N}, f(x,y) = x + y$ *Yhteenlasku*

- $f : \mathbb{N}^2 \to \mathbb{N}, f(x, y) = x \cdot y$ *Kertolasku*
- $f : \mathbb{N}^2 \to \mathbb{N}, f(x, y) = x \dot{-} y$ *Rajoitettu vähennyslasku*

Todistus:
Ks. Kurittu [16, Lause 1.3], [16, Lause 1.6] ja [16, Lause 1.9]. □

Huomautus 3.11 *Kaikkien näiden kuvausten ”johdannaiset” ovat myös primitiivirekursiivisia. Tämä seuraa primitiivirekursiivisuuden määritelmästä.*

3.4 Rajoitettu minimalisaatio

Määritelmä 3.12 (Rajoitettu minimalisaatio)
Olkoon $A \subset \mathbb{N}^{k+1}$ siten, että $A \in \mathcal{PR}$, missä $k \geq 1$. Määritellään kuvaus $\bar{\mu} : \mathbb{N}^{k+1} \to \mathbb{N}$ siten, että kaikille $\vec{x} \in \mathbb{N}^k$ ja $y \in \mathbb{N}$

$$\bar{\mu}_A(\vec{x}, y) = \begin{cases} \min\{z \in \mathbb{N} \mid (\vec{x}, z) \in A\}) & \textit{kun } \exists z(z < y \wedge (\vec{x}, z) \in A) \\ 0 & \textit{kun } \neg\exists z(z < y \wedge (\vec{x}, z) \in A) \end{cases}.$$

Sanotaan, että $\bar{\mu}_A$ on saatu rajoitetulla minimalisaatiolla joukosta A.

Rajoitettu minimalisaatio vastaa toimitusta, jossa etsitään pienintä luonnollista lukua joukosta A siten, että jos lukua ei ole löytynyt lukuun $y \in \mathbb{N}$ mennessä, niin merkitään tulokseksi nolla.

Tämä vertautuu sukupuuesimerkin tapaukseen, jossa tiedetään kuinka monen sukupolven päässä Jooseppi korkeintaan on ja selvitetään X:n jälkeläisyyttä. Jos jälkeläisyys selviää ennen y:ttä rekursioaskelta, niin ilmoitetann sukupolvien lukumäärä. Jos jälkeläisyys ei selviä, ilmoitetaan nolla.

Lause 3.13 *Funktio $\bar{\mu}_A$ on primitiivirekursiivinen.*

Todistus:
Ks. Kurittu [16, s.12]. □

3.5 Rajoittamaton minimalisaatio

Määritelmä 3.14 (Rajoittamaton minimalisaatio)
Olemassaoloehto: *Olkoon* $A \subset \mathbb{N}^{k+1}$ $(k \geq 1)$ *siten, että kaikille* $\vec{x} \in \mathbb{N}^k$ *on olemassa* $y \in \mathbb{N}$ *siten, että* $(\vec{x}, y) \in A$.

Määritellään funktio $\mu_A : \mathbb{N}^k \to \mathbb{N}$ *seuraavasti.*

$$\mu_A(\vec{x}) = \min\{y \in \mathbb{N} \mid (\vec{x}, y) \in A\}.$$

Sanotaan, että μ_A *on saatu rajoittamattomalla minimalisaatiolla joukosta* A.

Rajoittamaton minimalisaatio vastaa toimitusta, jossa etsitään pienintä luonnollista lukua, joka kuuluu joukkoon A sellaisessa tapauksessa, että tällainen luku on olemassa.

Tämä vertautuu sukupuuesimerkkiin seuraavalla tavalla. Oletetaan, että tiedetään X:n olevan Joosepin jälkeläinen. Selvitetään, kuinka monen sukupolven päässä Jooseppi eli. Kun selvitysprosessissa törmätään Jooseppiin, niin ilmoitetaan sukupolvien lukumäärä.

Selvää on, että tässä tapauksessa emme voi etukäteen arvioida toimitukseen kuluvaa aikaa. Tämä pätee myös kaikkien rekursiivisten funktioiden kohdalla, jotka eivät ole primitiivirekursiivisia [22, s.43].

Entä, jos X ei olekaan Joosepin jälkeläinen? Tässä tapauksessa selvitys jatkuisi aina ensimmäiseen sukupolveen asti. Luonnollisten lukujen tapauksessa tällaista rajaa ei tunnetusti ole. Tästä syystä olemassaoloehto on olennainen. Jos nimittäin ei ole olemassa y:tä siten, että $(\vec{x}, y) \in A$, niin laskeminen ei koskaan pääty.

Huomautus 3.15 μ_A *voi olla myös primitiivirekursiivinen. Esimerkiksi, olkoon* $A = \{(x, y) \in \mathbb{N}^2 \mid y \geq 1\}$. *Nyt voidaan valita* $\mu_A(x) = \mathrm{Y}(\mathrm{S}; \mathrm{Z}(x)) = 1$, *mikä on selvästi* p.r. . *Merkittävä seikka on, että* μ_A *ei ole aina* p.r. . *Esimerkki tästä on Ackermannin funktio. Jokainen* μ_A *on kuitenkin* μ*-rekursiivinen määritelmän 3.18 nojalla ja myös rekursiivinen lauseen 3.22 nojalla.*

3.6 Funktion μ-rekursiivisuus

Määritelmä 3.16 (μ-rekursio)
Olemassaoloehto: *Olkoon* $g : \mathbb{N}^{k+1} \to \mathbb{N}$ *funktio siten, että kaikilla* $\vec{x} \in \mathbb{N}^k$ *on olemassa* $y \in \mathbb{N}$ *siten, että* $g(\vec{x}, y) = 0$.

Olkoon $A = \{(\vec{x}, y) \in \mathbb{N}^{k+1} \mid g(\vec{x}, y) = 0\}$. *Olkoon funktio* $f(\vec{x}) = \mu_A(\vec{x})$. *Sanotaan, että* f *on saatu funktiosta* g μ*-rekursiolla. Ks. Odifreddi [17, s.21].*

Huomautus 3.17 μ*-rekursion määritelmä voidaan kirjoittaa myös muotoon, joka muistuttaa enemmän primitiivirekursion määritelmää (3.2). Olkoon funktio* g *ja joukko* A*, kuten edeltävässä määritelmässä. Määritellään funktio* $f = \mu_A$ *seuraavalla tavalla käyttäen apufunktiota* h*. Olkoon* $\vec{x} \in \mathbb{N}^k$ *ja* $h : \mathbb{N}^{k+2} \to \mathbb{N}$ *siten, että*

1. $h(\vec{x}, y, 0) = y$

2. $h(\vec{x}, y, z+1) = h(\vec{x}, y+1, g(\vec{x}, y+1))$

Nyt voimme kirjoittaa funktion f *muotoon* $f(\vec{x}) = h(\vec{x}, 0, g(\vec{x}, 0))$ *ja sanoa, että* f *on saatu* g*:stä* μ*-rekursiolla.*

Tämä määritelmä tosiaankin vastaa rajoittamattoman minimalisaation kautta tehtyä määritelmää (ks. [12, s.45]).

Määritelmä 3.18 (μ-rekursiiviset funktiot)
μ*-rekursiivisten funktioiden joukko on pienin suljettu funktiojoukko, joka sisältää määritelmän 3.4 funktiot 1-3 ja on suljettu yhdistämisen, primitiivirekursion ja* μ*-rekursion suhteen. Ks. Odifreddi [17, s.22].*

Seuraus: *Jos funktio on primitiivirekursiivinen, niin se on* μ*-rekursiivinen.*

Huomautus 3.19 *Odifreddi laittaa määritelmän Kleenen nimiin, joka on ilmeisesti julkaissut määritelmän 1936 artikkelissa General Recursive Functions of Natural Numbers, Mathemathische Annalen 112, 727-742. Tätä artikkelia ei kuitenkaan ollut saatavilla.*

3.7 Funktion rekursiivisuus

Churchin ja Kleenen käyttämää rekursiivisen funktion määritelmää varten tarvitaan formaali kieli, jonka avulla se voidaan esittää. Koska RA-kielen merkinnät käyvät tähän tarkoitukseen hyvin, niin tässä esitetään vasta epätäsmällinen määritelmä. Tarkka määritelmä löytyy liitteestä 9.2, johon voi paneutua, kun RA-kieli on tullut tutuksi.

Määritelmä 3.20 (Rekursiiviset funktiot)
Sanotaan, että funktio $f : \mathbb{N}^k \to \mathbb{N}$ on rekursiivinen, jos on olemassa joukko yhtälöitä, jotka määrittelevät f:n tietyllä tavalla rekursiivisesti siten, että sen arvo voidaan laskea yksiselitteisesti jokaisessa pisteessä $\vec{x} \in \mathbb{N}^k$.

Huomautus 3.21 *Tämä määritelmä vastaa Churchin, artikkelissa An Unsolvable Problem of Elementary Number Theory, käyttämää rekursiivisten funktioiden määritelmää. Määritelmä löytyy myös Kleeneltä [13, s.274] [12, s.43] ja Odifreddiltä [17, s.34-36].*

Lause 3.22 *Jos funktio on μ-rekursiivinen, niin se on rekursiivinen.*

Todistus:
Tästä voi vakuuttua melko helposti tutkimalla rekursiivisten funktioiden määritelmää ja μ-rekursiivisten funktioiden määritelmää määritelmän 3.16 huomautuksen näkökulmasta. Tarkemmin ks. Kleene [13, s.279] ja Odifreddi [17, s.34-37]. $\square$

Lause 3.23 *Jos funktio on rekursiivinen, niin se on μ-rekursiivinen.*

Todistus:
Ks. Kleene [13, s.289] tai [12, s.51]. $\square$

Huomautus 3.24 *Kahden edeltävän lauseen nojalla funktio on rekursiivinen jos ja vain jos se on μ-rekursiivinen. Tästä syystä voimme käyttää niistä kummastakin nimitystä rekursiivinen.*

Määritelmä 3.25 (μ-rekursiiviset joukot)
Sanotaan, että joukko on μ-rekursiivinen, jos sen karakteristinen funktio on μ-rekursiivinen. (Odifreddi [17, s.22])

Seuraus: *Lauseen 3.22 nojalla voi suoraan nähdä, että joukko on μ-rekursiivinen myös, jos sen karakteristinen funktio on rekursiivinen.*

Määritelmä 3.26 (Numeroiva funktio)
Olkoon $A \in \mathbb{N}$ ja $f : \mathbb{N} \to \mathbb{N}$ siten, että $A = \{f(n) \mid n \in \mathbb{N}\}$. Sanotaan, että f numeroi A:n.

Huomautus 3.27 *f voi saada saman arvon useammassa pisteessä.*

Määritelmä 3.28 (Primitiivirekursiivisesti numeroituvat joukot)
Sanotaan, että joukko $A \subset \mathbb{N}$ on primitiivirekursiivisesti numeroituva (merkitään: $A \in \mathcal{PRN}$), jos $A = \emptyset$ tai on olemassa primitiivirekursiivinen funktio f, joka numeroi A:n. $\boxed{\mathcal{PRN}}$

Määritelmä 3.29 (Rekursiivisesti numeroituvat joukot)
Sanotaan, että joukko $A \subset \mathbb{N}$ on rekursiivisesti numeroituva (merkitään: $A \in \mathcal{RN}$), jos $A = \emptyset$ tai on olemassa rekursiivinen funktio f, joka numeroi A:n. $\boxed{\mathcal{RN}}$

Lause 3.30 *Jos $A \in \mathbb{N}$ on rekursiivisesti numeroituva, niin se on myös primitiivirekursiivisesti numeroituva.*

Todistus:
Ks. Rosser [20, s.88] ja Kleene [13, s.307]. $\square$

Määritelmä 3.31 (Rekursiiviset joukot 1)
Sanotaan, että joukko $A \subset \mathbb{N}$ on rekursiivinen (merkitään: $A \in \mathcal{R}_1$), jos A ja $\mathbb{N} \backslash A$ ovat primitiivirekursiivisesti numeroituvia. (Robbin [19, s.123])

Määritelmä 3.32 (Rekursiiviset joukot 2)
Sanotaan, että joukko $A \subset \mathbb{N}$ on rekursiivinen (merkitään: $A \in \mathcal{R}_2$), jos sen karakteristinen funktio χ_A on rekursiivinen. (Kleene [12, s.45] ja Väänänen [22, s.42])

Lause 3.33 *$A \in \mathcal{R}_2$ jos ja vain jos $A \in \mathcal{RN}$ ja $\mathbb{N} \backslash A \in \mathcal{RN}$.*

Todistus:
Ks. Väänänen [22, s.45]. $\square$

Lause 3.34 $\mathcal{R}_1 = \mathcal{R}_2$.

Todistus:
1. $\mathcal{R}_1 \subset \mathcal{R}_2$
Olkoon $A \in \mathcal{R}_1$. Olkoon nyt f *p.r.* funktio, joka numeroi A:n ja f_c *p.r.* funktio, joka numeroi $\mathbb{N} \backslash A$:n. Lauseen 3.18 seurauksen nojalla f ja f_c ovat rekursiivisia. Nyt lauseen 3.33 nojalla $A \in \mathcal{R}_2$.

2. $\mathcal{R}_2 \subset \mathcal{R}_1$
Olkoon $A \in \mathcal{R}_2$ ja χ_A sen rekursiivinen karakteristinen funktio. Lauseen 3.30 nojalla riittää osoittaa, että A ja $\mathbb{N} \backslash A$ ovat rekursiivisesti numeroituvia.
i. $A \in \mathcal{RN}$
Jos $A = \emptyset$, niin väite pätee selvästi. Olkoon $A \neq \emptyset$ ja $a \in A$. Nyt voidaan määritellä rekursiivinen funktio $f : \mathbb{N} \to \mathbb{N}$ siten, että $f(n) = \chi_A(n) \cdot n + a \cdot (1 \dot{-} \chi_A(n))$. Toisin sanoen

$$f(n) = \begin{cases} n & \text{kun } n \in A \\ a & \text{kun } n \notin A \end{cases}.$$

Funktio f numeroi A:n. Siis $A \in \mathcal{RN}$.
ii. $\mathbb{N} \backslash A \in \mathcal{RN}$
Jos $A = \mathbb{N}$, niin väite pätee jälleen selvästi. Oletetaan siis, että $A \neq \mathbb{N}$ ja valitaan $b \in \mathbb{N} \backslash A$. Määritellään seuraavaksi funktio $f_c : \mathbb{N} \to \mathbb{N}$ siten, että $f_c(n) = \chi_A(n) \cdot b + n \cdot (1 \dot{-} \chi_A(n))$. Funktio f_c numeroi $\mathbb{N} \backslash A$:n. Siis $\mathbb{N} \backslash A \in \mathcal{RN}$.
□

Huomautus 3.35 *Koska joukot $\mathcal{R}_1$ ja $\mathcal{R}_2$ ovat samat, merkitään niitä pelkästään symbolilla $\mathcal{R}$. Siis $\mathcal{R}_1 = \mathcal{R}_2 = \mathcal{R}$.* $\boxed{\mathcal{R}}$

Huomautus 3.36 *Vaikka rekursiivinen joukko on aina rekursiivisesti numeroituva, niin päinvastainen ei päde. Churchin lause on vastaesimerkki tästä (ks. lauseet 6.2 ja 6.3).*

3.8 $\mathcal{PR} \subset \mathcal{R}$

Lause 3.37 $\mathcal{PR} \subset \mathcal{PRN}$.

Todistus:
Olkoon joukko $A \in \mathcal{PR}$. Jos $A = \emptyset$, niin selvästi väite pätee. Olkoon siis $A \neq \emptyset$ ja $a \in A$. Koska $A \in \mathcal{PR}$, niin on olemassa primitiivirekursiivinen karakteristinen funktio χ_A. Olkoon funktio $f : \mathbb{N} \to \mathbb{N}$ siten, että $f(x) = \chi_A(x) \cdot x + a \cdot (1 \dot{-} \chi_A(x))$. Nyt $A = \{f(n) \mid n \in \mathbb{N}\}$ eli f numeroi A:n. Lisäksi f on primitiivirekursiivinen, sillä laskutoimitukset $+$, $\dot{-}$ ja $\cdot$ ovat kaikki primitiivirekursiivisia määritelmän 3.10 nojalla. A on siis primitiivirekursiivisesti numeroituva. $\square$

Lause 3.38 *$\mathcal{PR} \subset \mathcal{R}$.*

Todistus:
Olkoon $A \in \mathcal{PR}$. Nyt lauseen 3.8 nojalla myös $\mathbb{N} \backslash A \in \mathcal{PR}$. Lauseen 3.37 nojalla $A \in \mathcal{PRN}$ ja $\mathbb{N} \backslash A \in \mathcal{PRN}$, josta lauseen 3.30 nojalla $A \in \mathcal{RN}$ ja $\mathbb{N} \backslash A \in \mathcal{RN}$. Siis A on rekursiivinen joukko. $\square$

Lause 3.39 *$\mathcal{R} \not\subset \mathcal{PR}$.*

Todistus:
Ackermannin funktio [22, s.43] tai Cantorin diagonaalinen metodi [13, s.272].
$\square$

4 RA-kieli

4.1 Syntaksi

Määritelmä 4.1 (RA-aakkoset)
RA-kielen aakkosia ovat. Z, S, P, R, Y, $|$, $\mathbf{0}$, $\mathbf{x}$, $\cong$, $\rightarrow$, f, $\forall$, [,], (, *ja*).

Määritelmä 4.2 (RA-sanat)
RA-kielen sanoja ovat kaikki RA-kielen aakkosista muodostetut äärelliset symbolijonot. Käytetään lihavoituja isoja kirjaimia $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$,... *merkitsemään RA-kielen sanoja.*

Määritelmä 4.3 (RA-sanajonot)
Olkoon $\mathcal{A} = \mathbf{A}_1, \ldots, \mathbf{A}_n$, *missä* $n \geq 1$ *ja* $\mathbf{A}_i$ *on RA-kielen sana kaikilla* $i = 1, \ldots, n$. *Sanotaan, että* $\mathcal{A}$ *on RA-kielen sanajono.*

Määritelmä 4.4
Merkitään sanajonon $\mathcal{A}$ *i:ttä jäsentä seuraavasti :* $(\mathcal{A})_i$. $\boxed{\mathcal{A}_i}$

Määritelmä 4.5 (RA-lyhenteet)
RA-kielen lyhenteitä ovat:

$\neg$	*: negaatio*	$\neg\mathbf{A} = [\mathbf{A} \rightarrow f]$	*("ei ole niin, että")*
$\vee$	*: disjunktio*	$\mathbf{A} \vee \mathbf{B} = [\neg\mathbf{A} \rightarrow \mathbf{B}]$	*("tai")*
$\wedge$	*: konjunktio*	$\mathbf{A} \wedge \mathbf{B} = \neg[\neg\mathbf{A} \vee \neg\mathbf{B}]$	*("ja")*
$\leftrightarrow$	*: ekvivalenssi*	$\mathbf{A} \leftrightarrow \mathbf{B} = [\mathbf{A} \rightarrow \mathbf{B}] \wedge [\mathbf{B} \rightarrow \mathbf{A}]$	*("jos ja vain jos")*
$\mathbf{k}_n$	*: numeraali*	$\mathbf{k}_n = \underbrace{S(S(\ldots S}_{n}(\mathbf{0}))\ldots)$	*(* $\mathbf{k}_0 = \mathbf{0}$*)*
$\mathbf{x_n}$	*: muuttujasymboli*	$\mathbf{x_n} = (\mathbf{x}\underbrace{\|\|\ldots\|}_{n})$	*(* $\mathbf{x_0} = \mathbf{x}$*)*
$\exists$	*: eksistenssikvanttori*	$\exists\mathbf{x_n}\mathbf{A} = \neg\forall\mathbf{x_n}\neg\mathbf{A}$	*("jollakin")*
P^n_i	*: projektiosymboli*	$P^n_i = (\underbrace{\|\|\ldots\|}_{n}\, P\, \underbrace{\|\|\ldots\|}_{i})$	

Huomautus 4.6 *Numeraalien ja muuttujasymbolien suhde RA-kielessä tulee vastaamaan muuttujasymbolien ja vakioiden suhdetta predikaattilogiikassa. Kutsumme siis numeraaleja myös RA-kielen vakiosymboleiksi.*

Määritelmä 4.7 (RA-funktiosymbolit)
Olkoon $m, n \in \mathbb{N}$ *ja* $\mathcal{F}(m,n) \subset \{$*RA-kielen sanat*$\}$ *seuraavalla tavalla määriteltyjä joukkoja:*

- $\mathcal{F}(1,1) = \{Z, S, P_1^1\}$
- $\mathcal{F}(1,n) = \{P_i^n | i \in \{1,\ldots,n\}\}$, *kun* $n \geq 2$
- *Jos* $m \geq 2$, *niin tehdään rekursio-oletus, että joukot* $\mathcal{F}(k,n)$, $k \in \{1,\ldots,m-1\}$ *on jo määritelty kaikille* $n \geq 1$. *Kun* $k = m$, *niin sallitaan kaksi vaihtoehtoa.*
  1. $n \geq 2$ *ja on olemassa* $j, k \in \{1,\ldots,m-1\}$ *sekä* $\mathbf{G} \in \mathcal{F}(j, n-1)$ *ja* $\mathbf{H} \in \mathcal{F}(k, n+1)$. *Nyt*
  $$(R\mathbf{GH}) \in \mathcal{F}(m,n).$$
  2. *On olemassa* $q \geq 1$ *ja* $j, k_1, \ldots, k_q \in \{1,\ldots,m-1\}$ *siten, että on olemassa* $\mathbf{H} \in \mathcal{F}(j,q)$ *ja kaikille* $i \in \{1,\ldots,q\}$ *on olemassa* $\mathbf{G}_i \in \mathcal{F}(k_i, n)$. *Nyt*
  $$(Y\mathbf{H}\mathbf{G}_1 \ldots \mathbf{G}_q) \in \mathcal{F}(m,n).$$

Sanotaan, että $\mathbf{F}$ *on n-paikkainen funktiosymboli, jos* $\mathbf{F} \in \bigcup_{m=1}^{\infty} \mathcal{F}(m,n)$

Määritelmä 4.8 (RA-termit)
Olkoon $\mathbf{A}$ *RA-kielen sana.* $\mathbf{A}$*:n termirakennejono on RA-kielen sanajono* $\mathbf{A}_1, \ldots, \mathbf{A}_m$ *siten, että* $\mathbf{A}_m = \mathbf{A}$ *ja kaikille* $i \in \{1,\ldots,m\}$ *pätee joku seuraavista vaihtoehdoista.*

1. $\mathbf{A}_i = \mathbf{0}$.
2. $\mathbf{A}_i = \mathbf{x_n}$ *jollekin* $n \in \mathbb{N}$.
3. *On olemassa* $j_1, \ldots, j_n \in \{1,\ldots,i-1\}$ *ja n-paikkainen funktiosymboli* $\mathbf{F}$ *siten, että*
$$\mathbf{A}_i = \mathbf{F}(\mathbf{A}_{j1}\mathbf{A}_{j2}\ldots\mathbf{A}_{jn}).$$

Sanotaan, että sana $\mathbf{A}$ *on termi, jos sillä on termirakennejono.*

Huomautus 4.9 *Luettavuuden parantamiseksi voidaan kohdan 3 muotoiseen termiin lisätä pilkkuja seuraavalla tavalla*

$$\mathbf{A}_i = \mathbf{F}(\mathbf{A}_{j1}, \mathbf{A}_{j2}, \ldots, \mathbf{A}_{jn}).$$

Pilkut eivät kuitenkaan ole RA-kieltä.

Määritelmä 4.10 (RA-kaavat)
Olkoon $\mathbf{A}$ *RA-kielen sana.* $\mathbf{A}$*:n kaavarakennejono on äärellinen jono* $\mathbf{A}_1, \ldots, \mathbf{A}_m$ *RA-kielen sanoja siten, että* $\mathbf{A}_m = \mathbf{A}$ *ja kaikille* $i \in \{1, \ldots, m\}$ *pätee joku seuraavista vaihtoehdoista.*

1. $\mathbf{A}_i = \mathbf{t} \cong \mathbf{s}$, *missä* $\mathbf{t}$ *ja* $\mathbf{s}$ *ovat termejä.*

2. $\mathbf{A}_i = f$.

3. *On olemassa* $j, k \in \{1, \ldots, i-1\}$ *siten, että* $\mathbf{A}_i = [\mathbf{A}_j \to \mathbf{A}_k]$.

4. *On olemassa* $j \in \{1, \ldots, i-1\}$ *ja muuttujasymboli* $\mathbf{x_n}$ *siten, että* $\mathbf{A}_i = \forall \mathbf{x_n} \mathbf{A}_j$.

Sanotaan, että sana $\mathbf{A}$ *on kaava, jos sillä on kaavarakennejono.*

Määritelmä 4.11 (Sijoitus)
Merkinnällä $S_{\mathbf{t}}^{\mathbf{x_n}}(\mathbf{A})$ *tarkoittaa kaavaa, joka saadaan, kun korvataan kaikki* $\mathbf{x_n}$*:n vapaat*[8] *esiintymät kaavassa* $\mathbf{A}$ *termillä* $\mathbf{t}$.

Huomautus 4.12 *Tarkemmin RA-kielen syntaksista ks. [19, alkaen s.72] ja [16, alkaen s.27]. Sijoituksen tarkempi määritelmä ks. [16, s.33-34].*

Lause 4.13 (Funktiosymbolirakennejono) *Jokaisella RA-kielen funktiosymbolilla on funktiosymbolirakennejono. Ks. [16, s.29-30]*

RA-kieli on predikaattilogiikan laajennus samaan tapaan kuin predikaattilogiikka on propositiologiikan laajennus. RA-kieli sisältää kaikki predikaattilogiikan aksioomat ja päättelysäännöt [16, s.35]. Olkoon $\mathbf{A}$, $\mathbf{B}$ ja $\mathbf{C}$ mielivaltaisia kaavoja ja $n \in \mathbb{N}$ mielivaltainen. Olkoon $\mathbf{t}$ aksioomassa 4 mielivaltainen

[8]Muuttuja $\mathbf{x_n}$ on vapaa, kun se ei sijaitse sitä koskevan kvantifikaation $\forall \mathbf{x_n}$ vaikutusalueella.

vakio- tai muuttujasymboli, kun on kyse predikaattilogiikasta ja mielivaltainen termi, kun on kyse RA-kielestä.

Propositiologiikan aksioomat:

Aks 1: $\mathbf{A} \to [\mathbf{B} \to \mathbf{A}]$

Aks 2: $[\mathbf{A} \to [\mathbf{B} \to \mathbf{C}]] \to [[\mathbf{A} \to \mathbf{B}] \to [\mathbf{A} \to \mathbf{C}]]$

Aks 3: $\neg\neg\mathbf{A} \to \mathbf{A}$

Predikaattilogiikan lisäaksioomat:

Aks 4: $\forall \mathbf{x_n} \mathbf{A} \to S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A})$, missä sijoitus $S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A})$ ei sido[9] $\mathbf{x_n}$:ää.

Aks 5: $\forall \mathbf{x_n}[\mathbf{A} \to \mathbf{B}] \to [\mathbf{A} \to \forall \mathbf{x_n} \mathbf{B}]$, missä $\mathbf{x_n}$ ei esiinny vapaana $\mathbf{A}$:ssa.

RA-kielessä aksiooma 4 laajenee siis koskemaan myös termejä. Tätä voidaan kutsua laajennukseksi, sillä vakiot ja muuttujat ovat myös termejä.

RA-kielessä on myös 12 muuta aksioomaa. Kuten huomattiin, niin RA-kielen kaavat käsittelevät yhtälöitä. Yksitoista aksioomista koskevatkin yhtälöiden ominaisuuksia. Lisäksi yksi aksiooma imitoi induktioperiaatetta. Nämä aksioomat on valittu siten, että RA-kielen päättelyt eivät johda ristiriitaan luonnollisten lukujen semantiikan kanssa. [16, s.36-37]

Päättely RA-kielessä määritellään täsmälleen samalla tavalla kuin predikaattilogiikassakin. Seuraavassa määritelmässä kohdan 2. päättelysääntö on nimeltään *modus ponens* ja kohdan 3. päättelysääntö nimeltään *kvantifiointi*.

Määritelmä 4.14 (RA-päättelyjono)
Olkoon $\mathbf{A}$ *RA-kielen kaava. Sanotaan, että RA-kielen kaavat* $\mathbf{A}_1, \dots, \mathbf{A}_n$ *muodostavat* $\mathbf{A}$*:n päättelyjonon, jos* $\mathbf{A}_n = \mathbf{A}$ *ja kaikille* $i \in \{1, \dots, n\}$ *pätee joku seuraavista vaihtoehdoista.*

1. $\mathbf{A}_i$ *on aksiooma.*

2. *On olemassa* $j, k \in \{1, \dots, i-1\}$ *siten, että* $\mathbf{A}_k = \mathbf{A}_j \to \mathbf{A}_i$.

[9] $S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A})$ sitoo $\mathbf{x_n}$:n, jos $\mathbf{x_n}$ esiintyy vapaana $\mathbf{A}$:ssa kvantifikaation $\forall \mathbf{x_k}$ vaikutusalueella ja $\mathbf{x_k}$ esiintyy $\mathbf{t}$:ssä. $(n \neq k)$

3. *On olemassa* $j \in \{1, \dots, i-1\}$ *ja muuttujasymboli* $\mathbf{x_u}$ *siten, että* $\mathbf{A}_i = \forall \mathbf{x_u} \mathbf{A}_j$.

Määritelmä 4.15 (RA-teoreema)
Sanotaan, että RA-kielen kaava $\mathbf{A}$ *on teoreema, jos sillä on päättelyjono.*

Kaikki propositio- ja predikaattilogiikan teoreemat ovat siis myös RA-kielen teoreemoja, kun niissä olevat propositiokirjaimet ja predikaatit korvataan RA-kielen kaavoilla.

Seuraavaksi esitellään predikaattilogiikan teoreema, jota tarvitaan Churchin lauseen todistuksessa RA-kielelle.

Lause 4.16 $\vdash S_{\mathbf{t}}^{\mathbf{x_n}}(\mathbf{A}) \rightarrow \exists \mathbf{x_n} \mathbf{A}$, *jos sijoitus ei sido muuttujaa* $\mathbf{x_n}$.

Todistus:
Robbin esitteli teoreeman kirjassa [19, s. 49] ilman todistusta, joten todistus esitetään tämän gradun liitteessä. Ks. liite s. 50. □

Seuraava teoreema on puhtaasti syntaktinen ja hyvin merkittävä RA-kielen kannalta.

Lause 4.17 (Korvaavuusperiaate) *Olkoot* $\mathbf{s}$ *ja* $\mathbf{t}$ *termejä sekä* $\mathbf{A}$ *kaava siten, että* $\mathbf{s}$ *esiintyy* $\mathbf{A}$*:ssa ja jos muuttuja* $\mathbf{x_n}$ *esiintyy* $\mathbf{s}$*:ssä tai* $\mathbf{t}$*:ssa, niin* $\mathbf{s}$ *ei esiinny* $\mathbf{A}$*:ssa olevan kvantifikaation* $\forall \mathbf{x_n}$ *alueella. Olkoon* $S_{\mathbf{t}}^{\mathbf{s}}(\mathbf{A})$ *sana, joka syntyy, kun* $\mathbf{A}$*:ssa sijoitetaan* $\mathbf{s}$*:n paikalle* $\mathbf{t}$*. Tällöin pätee:*

1. $S_{\mathbf{t}}^{\mathbf{s}}(\mathbf{A})$ *on kaava.*

2. $\vdash \mathbf{s} \cong \mathbf{t} \rightarrow [\mathbf{A} \longleftrightarrow S_{\mathbf{t}}^{\mathbf{s}}(\mathbf{A})]$

Todistus:
[16, Lause 2.23] tai [19, Teoreema 25.6] □

4.2 Semantiikka

RA-kielen semantiikka määritellään niin, että se imitoi luonnollisten lukujen primitiivirekursiivisia laskutoimituksia.

Määritelmä 4.18 (Funktiosymbolin valuaatio)
Määritellään funktio $val : \{\text{RA-funktiosymbolit}\} \to \{\text{p.r. funktiot}\}$ seuraavasti.

1. $val(Z) = \mathrm{Z}$.
2. $val(S) = \mathrm{S}$.
3. $val(P_i^n) = \mathrm{P}_i^n$, *kaikille* $n \in \mathbb{N}$ *ja* $i = 1, \ldots, n$.
4. $val(R\mathbf{G}\mathbf{H}) = \mathrm{R}(val(\mathbf{G}), val(\mathbf{H}))$.
5. $val(Y\mathbf{H}\mathbf{G}_1 \ldots \mathbf{G}_q) = \mathrm{Y}(val(\mathbf{H}); val(\mathbf{G}_1) \ldots val(\mathbf{G}_q))$

Sanotaan, että $val(\mathbf{F})$ *on funktiosymbolin* $\mathbf{F}$ *valuaatio.*

Määritelmä 4.19 (Termin valuaatio)
Olkoon $T = \{\mathbf{t} \in \{\text{RA-termit}\} \mid \mathbf{t} \text{ ei sisällä muuttujasymboleja}\}$. *Määritellään funktio* $v : T \to \mathbb{N}$ *seuraavasti.*

1. $v(\mathbf{0}) = 0$.
2. $v(\mathbf{F}(\mathbf{t}_1, \ldots, \mathbf{t}_n)) = val(\mathbf{F})(v(\mathbf{t}_1), \ldots, v(\mathbf{t}_n))$, *missä* $\mathbf{F}$ *on* n*-paikkainen funktiosymboli ja* $\mathbf{t}_1, \ldots, \mathbf{t}_n \in T$.

Sanotaan, että $v(\mathbf{t})$ *on termin* $\mathbf{t}$ *valuaatio.*

Huomautus 4.20 *Funktio* $f : \mathbb{N}^k \to \mathbb{N}$ *on primitiivirekursiivinen jos ja vain jos on olemassa* n*-paikkainen funktiosymboli siten, että* $val(\mathbf{F}) = f$. *Toisin sanoen RA-kielen funktiosymbolit imitoivat täsmällisesti primitiivirekursiivisia funktioita. Ks. Kurittu [16, s.64].*

Määritelmä 4.21 (Suljetun kaavan validius)
Olkoon $\mathbf{A}$ *suljettu*[10] *RA-kielen kaava. Merkitsemme* $\mathbb{N} \models \mathbf{A}$, *jos* $\mathbf{A}$ *on validi eli* ***tosi luonnollisten lukujen mallissa*** *ja* $\mathbb{N} \not\models \mathbf{A}$ *jos se ei ole validi. Validius määritellään seuraavasti.*

1. *Olkoon* $\mathbf{s}$ *ja* $\mathbf{t}$ *muuttujattomia termejä. Sanotaan, että*

[10]Kaava on suljettu, jos se ei sisällä yhtään vapaata muuttujaa.

$$\mathbb{N} \models \mathbf{t} \cong \mathbf{s} \qquad \textit{jos } v(\mathbf{t}) = v(\mathbf{s})$$
$$\mathbb{N} \models \mathbf{t} \ncong \mathbf{s} \qquad \textit{jos } v(\mathbf{t}) \neq v(\mathbf{s}).$$

2. $\mathbb{N} \not\models f$.

3. *Olkoon* **A** *ja* **B** *RA-kielen suljettuja kaavoja. Sanotaan, että*

$$\mathbb{N} \not\models \mathbf{A} \to \mathbf{B} \qquad \textit{jos } \mathbb{N} \models \mathbf{A} \textit{ ja } \mathbb{N} \not\models \mathbf{B},$$
$$\mathbb{N} \models \mathbf{A} \to \mathbf{B} \qquad \textit{muuten.}$$

4. *Olkoon* $\forall x\mathbf{A}$ *on RA-kielen suljettu kaava. Sanotaan, että*

$$\mathbb{N} \models \forall x\mathbf{A} \qquad \textit{jos } \mathbb{N} \models S^{\mathbf{x}}_{\mathbf{k}_n}(\mathbf{A}) \textit{ kaikilla } n \in \mathbb{N},$$
$$\mathbb{N} \not\models \forall x\mathbf{A} \qquad \textit{muuten.}$$

4.3 Gödel-numerointi

Gödel keksi, että jokaiseen RA-kielen sanaan voidaan kiinnittää yksiselitteinen lukuarvo. Tätä lukuarvoa sanotaan kyseisen sanan Gödel-luvuksi. Toisin sanoen on olemassa injektio

$$\gamma_s : \{\text{RA-kielen sanat}\} \to \mathbb{N}.$$

Samanlainen koodaus tulee löytää myös sanajonoille eli injektio

$$\gamma_j : \{\text{RA-kielen sanajonot}\} \to \mathbb{N}.$$

Olkoon **A** jokin RA-kielen mielivaltainen sana tai sanajono. Merkitään sanojen ja sanajonojen Gödel-lukuja samalla tavalla. $\gamma_{s,j}(\mathbf{A}) = ``\mathbf{A}''$. Asiayhteydestä selviää, onko kyse sanan vai sanajonon Gödel-luvusta.

Gödel-numerointi mahdollistaa RA-kielen itseviittaavuuden. Tässä yhteydessä ei ole olennaista, miten numerointi tapahtuu vaan, että se voidaan tehdä. Hyvä esitys Gödel-luvuista löytyy Kuritun monisteesta [16, s.88]. Nykyisin tällainen symbolien koodaus numeroiksi on tuttua myös tietotekniikan kautta (esim. ASCII merkit). Yksinkertainen koodaustapa löytyy myös Davisin kirjasta [5, s.222].

5 Määritelmiä ja lauseita

Lause 5.1 (Eheyslause) *Jokainen RA-kielen teoreema on validi.*

Todistus:
Ks. [16, Lause 2.47] $\square$

Lause 5.2 (Gödelin heikko täydellisyyslause)
Olkoon $\mathbf{A}$ *RA-kielen kaava, jossa ei esiinny muuttujia ja* $\mathbb{N} \vDash \mathbf{A}$. *Tällöin* $\mathbf{A}$ *on teoreema.*

Todistus:
Ks. [16, Lause 2.55] ja [19, Teoreema 25.5]. $\square$

Määritelmä 5.3
Olkoon $\mathcal{PJ} = \{\text{“}\mathcal{A}\text{”} \in \mathbb{N} \mid \mathcal{A}$ *on RA-kielen päättelyjono.*$\}$ $\boxed{\mathcal{PJ}}$

Lause 5.4 $\mathcal{PJ}$ *on primitiivirekursiivinen.*

Todistus:
Ks. [19, Teoreema 36.2] tai [16, Lause 3.26]. $\square$

Määritelmä 5.5
Olkoon $sij : \mathbb{N}^3 \to \mathbb{N}$ *funktio siten, että* $sij(x, n, a) = \text{“}S^{\mathbf{x}}_{\mathbf{k}_n}(\mathbf{A})\text{”}$, *missä* $\mathbf{x}$ *on yksittäinen muuttuja ja* $x = \text{“}\mathbf{x}\text{”}$ *sekä* $\mathbf{A}$ *RA-kielen kaava siten, että* $a = \text{“}\mathbf{A}\text{”}$.

Lause 5.6 *Funktio* sij *on primitiivirekursiivinen.*

Todistus:
Ks. [19, Teoreema 35.4] ja [16, Lause 3.22]. $\square$

Huomautus 5.7 *Koska* sij *on primitiivirekursiivinen, on myös olemassa RA-kielen 3 paikkainen funktiosymboli* **sij**, *joka imitoi sitä. [16, Lause 2.27]*

5.1 Ristiriidattomuus

Määritelmä 5.8
Sanotaan, että formaali kieli K *on ristiriitainen, jos on olemassa* K*:n kaava* $\mathbf{A}$ *siten, että* $\vdash \mathbf{A}$ *ja* $\vdash \neg\mathbf{A}$.

Määritelmä 5.9

Sanotaan, että formaali kieli K on ristiriidaton, jos se ei ole ristiriitainen.

5.2 ω-ristiriidattomuus

Määritelmä 5.10

Sanotaan, että formaali kieli K on ω-ristiriitainen, jos on olemassa K:n kaava $\mathbf{A}$ ja yksittäinen muuttuja x siten, että

$$\vdash \exists x \mathbf{A} \text{ ja } \vdash \neg S^{x}_{k_n}(\mathbf{A}) \text{ kaikilla } n = 0, 1, 2, \ldots.$$

Määritelmä 5.11

Sanotaan, että formaali kieli K on ω-ristiriidaton, jos se ei ole ω-ristiriitainen.

6 Churchin lause RA-kielelle

6.1 Todistus

Tässä luvussa päästään tämän pro gradu - tutkielman varsinaiseen ydinasiaan eli Churchin lauseen todistukseen.

Määritelmä 6.1
Olkoon $\mathcal{TR}$ kaikkien RA-kielen teoreemojen Gödel-lukujen joukko:

$$\mathcal{TR} = \{\text{``}\mathbf{A}\text{''} \in \mathbb{N} \mid \vdash \mathbf{A}\}.$$

Joukosta $\mathcal{TR}$ on syytä pitää mielessä, että kyseessä on vain joukko luonnollisia lukuja.

Seuraava lause osoittaa, että on olemassa primitiivirekursiivinen menetelmä, jonka avulla joukko $\mathcal{TR}$ voidaan muodostaa. Sitä käytetään apuna varsinaisen Churchin lauseen todistamisessa.

Lause 6.2 *$\mathcal{TR}$ on primitiivirekursiivisesti numeroituva.*

Todistus:
Teoreeman 5.4 nojalla RA-päättelyjonojen Gödel-lukujen joukko $\mathcal{PJ}$ on primitiivirekursiivinen joukko. Voidaan siis valita primitiivirekursiivinen funktio f seuraavalla tavalla. Olkoon $\mathcal{A}$ mielivaltainen RA-kielen sanajono, nyt

$$f(\text{``}\mathcal{A}\text{''}) = \begin{cases} \text{``}(\mathcal{A})_n\text{''} & \text{kun } \mathcal{A} \in \mathcal{PJ} \\ \text{``}\mathbf{0} \cong \mathbf{0}\text{''} & \text{kun } \mathcal{A} \notin \mathcal{PJ} \end{cases},$$

missä n on $\mathcal{A}$:n pituus. "$(\mathcal{A})_n$" on siis $\mathcal{A}$:n viimeisen jäsenen Gödel-luku (ks. määritelmä 4.4. Huomataan, että "$(\mathcal{A})_n$" on aina teoreema ja käymällä läpi kaikki päättelyjonot saadaan kaikki RA-kielen teoreemat. Nyt f on sopiva funktio numeroimaan joukon $\mathcal{TR}$, joten $\mathcal{TR}$ on primitiivirekursiivisesti numeroituva. [19, s.126] □

Lause 6.3 (Churchin lause RA-kielelle) *Jos RA-kieli on ω-ristiriidaton, niin $\mathcal{TR}$ ei ole rekursiivinen.*

Todistus:
Tehdään antiteesi eli oletetaan, että $\mathcal{TR}$ on rekursiivinen. Siis toisin sanoen $\mathcal{TR}$ ja $\mathbb{N}\backslash\mathcal{TR}$ ovat primitiivirekursiivisesti numeroituvia.

Lauseen 6.2 mukaan $\mathcal{TR}$ on primitiivirekursiivisesti numeroituva, joten siitä ei löydy ristiriitaa. Antiteesin ja määritelmän 3.31 nojalla tiedetään, että on olemassa primitiivirekursiivinen funktio g joka numeroi joukon $\mathbb{N}\backslash\mathcal{TR}$. Olkoon siis $\mathbf{g}$ funktiosymboli, joka imitoi funktiota g ja $\mathbf{sij}$ funktiosymboli, joka imitoi funktiota sij (ks. huomautus 5.7).

Olkoon $\mathbf{I}$ RA-kielen kaava

$$\exists x_1(\mathbf{g}(x_1) \cong \mathbf{sij}(\mathbf{k}_{``x_0"}, x_0, x_0))$$

ja olkoon luku $i = ``\mathbf{I}"$. Olkoon lisäksi $\mathbf{J}$ RA-kielen suljettu kaava

$$\exists x_1(\mathbf{g}(x_1) \cong \mathbf{sij}(\mathbf{k}_{``x_0"}, \mathbf{k}_i, \mathbf{k}_i))$$

ja luku $j = ``\mathbf{J}"$.

Nyt $\mathbf{J}$ joko on tai ei ole teoreema. Oletetaan ensin, että $\mathbf{J}$ ei ole teoreema. Toisin sanoen $j \in \mathbb{N}\backslash\mathcal{TR}$. Nyt siis on olemassa jokin $n \in \mathbb{N}$, jolle

$$g(n) = j.$$

Teoreeman 5.2 nojalla

$$\vdash \mathbf{g}(\mathbf{k}_n) \cong \mathbf{k}_j \tag{1}$$

Toisaalta voidaan huomata, että

$$\begin{aligned} sij(``\mathbf{x_0}", i, i) &= ``S^{\mathbf{x_0}}_{\mathbf{k}_i}\mathbf{I}" \\ &= ``S^{\mathbf{x_0}}_{\mathbf{k}_i}(\exists \mathbf{x_1}(\mathbf{g}(\mathbf{x_1}) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}"}, \mathbf{x_0}, \mathbf{x_0}))" \\ &= ``\exists \mathbf{x_1}(\mathbf{g}(\mathbf{x_1}) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}"}, \mathbf{k}_i, \mathbf{k}_i))" \\ &= ``\mathbf{J}" = j. \end{aligned} \tag{2}$$

Koska $sij(``\mathbf{x_0}", i, i) = v(\mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}"}, \mathbf{k}_i, \mathbf{k}_i))$ ja $j = v(\mathbf{k}_j)$, niin määritelmän 4.21 ja yhtälön (2) nojalla kaava $\mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}"}, \mathbf{k}_i, \mathbf{k}_i) \cong \mathbf{k}_j$ on validi. Kyseisessä kaavassa ei ole muuttujia, joten teoreeman 5.2 nojalla kaava on myös teoreema. Siis

$$\vdash \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}"}, \mathbf{k}_i, \mathbf{k}_i) \cong \mathbf{k}_j.$$

Nyt teoreemojen (1) ja 4.17 nojalla

$$\vdash \mathbf{g}(\mathbf{k}_n) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i).$$

Koska $\mathbf{g}(\mathbf{k}_n) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i)$ on toisaalta $S^{\mathbf{x_1}}_{\mathbf{k}_n}(\mathbf{g}(\mathbf{x_1}) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i))$, niin teoreeman 4.16 nojalla

$$\vdash \exists \mathbf{x_1}(\mathbf{g}(\mathbf{x_1}) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i))$$

eli toisin sanoen

$$\vdash \mathbf{J},$$

mikä on ristiriidassa lähtöoletuksen kanssa.

Oletetaan seuraavaksi, että $\mathbf{J}$ on teoreema eli

$$\vdash \exists x_1(\mathbf{g}(x_1) \cong \mathbf{sij}(\mathbf{k}_{``x_0''}, \mathbf{k}_i, \mathbf{k}_i)). \tag{3}$$

Toisin sanoen $j \in \mathcal{TR}$ eli $j \notin \mathbb{N} \backslash \mathcal{TR}$. Kaikille $n \in \mathbb{N}$ pätee siis $g(n) \neq j$. Nyt teoreeman 5.2 nojalla

$$\vdash \neg(\mathbf{g}(\mathbf{k}_n) \cong \mathbf{k}_j) \text{ kaikilla } n \in \mathbb{N}.$$

Vastaavalla päättelyllä kuin edellistapauksessa saadaan

$$\vdash \neg(\mathbf{g}(\mathbf{k}_n) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i)) \text{ kaikilla } n \in \mathbb{N},$$

josta

$$\vdash \neg S^{\mathbf{x_1}}_{\mathbf{k}_n}(\mathbf{g}(\mathbf{x_1}) \cong \mathbf{sij}(\mathbf{k}_{``\mathbf{x_0}''}, \mathbf{k}_i, \mathbf{k}_i)) \text{ kaikilla } n \in \mathbb{N}. \tag{4}$$

Nyt teoreemat (3) ja (4) johtavat ristiriitaan oletuksen kanssa, jonka mukaan RA-kieli on ω-ristiriidaton.

Kumpikin oletus $\vdash \mathbf{J}$ ja $\vdash \neg\, \mathbf{J}$ johtivat ristiriitaan. Antiteesi on siis väärä, joten $\mathcal{TR}$ ei ole rekursiivinen, jos RA-kieli on ω-ristiriidaton. □

6.2 ω-ristiriidattomuuden merkitys

Onko RA-kieli sitten ω-ristiriidaton? Gödelin toisen epätäydellisyyslauseen nojalla tätä ei voi todistaa pelkästään syntaksin kautta, vaan tarvitaan avuksi luonnollisten lukujen semantiikkaa. Sama pätee myös RA-kielen ristiriidattomuuden tapauksessa [19, s.114-115]. Jos siis oletetaan luonnollisten lukujen intuitiivisen semantiikan olevan ristiriidaton, niin RA-kieli on ω-ristiriidaton. Jos oletusta ei tehdä, niin ristiriidattomuutta ei voida todistaa.

Määritelmä 6.4
Sanotaan, että luonnollisten lukujen semantiikka on ristiriidaton, jos ei ole olemassa $\mathbf{A}$ *siten, että* $\mathbb{N} \models \mathbf{A}$ *ja* $\mathbb{N} \models \neg\mathbf{A}$.

Lause 6.5 *RA-kieli on ω-ristiriidaton, jos oletamme luonnollisten lukujen semantiikan olevan ristiriidaton.*

Todistus:
Tehdään antiteesi, jonka mukaan RA-kieli on ω-ristiriitainen. On siis olemassa RA-kielen kaava $\mathbf{A}$ siten, että $\vdash \neg S^x_{\mathbf{k}_n}(\mathbf{A})$ kaikilla $n = 0, 1, 2, \ldots$ ja $\vdash \exists x(\mathbf{A})$. Nyt eheyslauseen (5.1) nojalla $\mathbb{N} \models \neg S^x_{\mathbf{k}_n}(\mathbf{A})$ kaikilla $n = 1, 2, \ldots$ ja RA-kielen semantiikan määritelmän (4.21) nojalla

$$\mathbb{N} \models \forall x(\neg\mathbf{A}). \tag{5}$$

Koska myös $\vdash \exists x(\mathbf{A})$ (eli $\vdash \neg\forall x(\neg\mathbf{A})$) pätee, niin eheyslauseen nojalla $\mathbb{N} \models \neg\forall x(\neg\mathbf{A})$, mikä johtaa ristiriitaan toteamuksen (5) ja oletuksen kanssa, jonka mukaan luonnollisten lukujen semantiikka on ristiriidaton. □

Tämä tarkoittaa, että jos haluamme epäillä Churchin lauseen pätevyyttä, niin meidän tulee asettaa intuitiivisen lukukäsityksemme ristiriidattomuus kyseenalaiseksi.

7 Churchin lauseen historiaa

7.1 Laskettavuus

Ajatus formaalista kielestä, jonka avulla voidaan ratkaista monimutkaisia ongelmia on jo melko vanha. G. W. Leibniz, 1600-luvulla elänyt matemaatikko ja filosofi, haaveili symbolikielestä, jonka avulla voisi muotoilla minkä tahansa todellisuutta koskevan kysymyksen. Hän oli saanut innoituksensa aritmetiikasta ja algebrasta, jotka olivat esimerkkejä tarkoin harkitun symbolikielen eduista. Hänen etsimänsä kieli kantoi nimeä *calculus ratiocinator*.

Tällaisen kielen kehittäminen edellyttäisi kaiken inhimillisen tiedon *ensyklopediaa*, jonka valmistuttua olisi mahdollista ilmaista kaikkein perustavimmat käsitteet merkein ja sisällyttää ne kieleen. Tämä merkkikieli voisi operoida näillä keskeisillä käsitteillä ja esimerkiksi johtaa muita käsitteitä perustavammista.

Leibnizin eräässä kirjoituksessa kuvaillaan pöydän ääressä istuvia vaikeaa ongelmaa ratkovia vakavia "viisauden ystäviä", jotka ensin muotoilevat käsillä olevan ongelman tällä symbolikielellä ja huudahtavat sitten: "Laskemaan!". Ongelma ratkeaisi kynän ja paperin avulla ja kaikkien olisi pakko hyväksyä ratkaisu oikeaksi. [5, s.26-27]

Leibnizin toiveista ja useista yrityksistä huolimatta *calculus ratiocinator* ei koskaan valmistunut. Mielenkiintoista on, että joillakin kielen luonnosteluilla on yhteyksiä esimerkiksi Boolen logiikkaan. [5, s.27-28]

Hypätään seuraavaksi 1800-luvulle, jolloin Gottlob Frege julkaisi *Begriffsschrift* eli Käsitekirjoitus nimisen kirjasen, joka oli hyvin merkittävä kehitysaskel logiikan alalla ja loi pohjan modernille logiikalle ja matemaattisille merkintätavoille. Kirjan alaotsikon mukaan se oli "lukuteorian mallin mukainen puhtaan ajattelun kaavakieli". [5, s.56-57]

Fregen logiikasta on tullut se peruslogiikka, jota opetetaan matematiikan, tietojenkäsittelyopin ja filosofian johdantokursseilla. Se pitää sisällään mm. kvanttorien ja muuttujien käytön. Frege muotoili siis keinotekoisen kielen jolla on armottoman tarkka syntaksi ja päättelysäännöt. Tämän ansiosta loogiset päättelyt voitiin esittää mekaaniseen tapaan, viittaamalla ainoastaan merkkien erilaisiin muodostelmiin. Kyseessä oli ensimmäinen esimerkki tiukan formaalista symbolikielestä. Fregen käsitekirjoitus onkin eräänlainen ohjelmointikielien esimuoto. [5, s.61]

On huomattava, että Frege ei esitellyt varsinaisesti niin sanottua ensimmäisen kertaluvun logiikkaa eli predikaattilogiikkaa vaan toisen kertaluvun logiikan, joka kylläkin sisältää predikaattilogiikan. Toisen kertaluvun logiikassa sallitaan kvantifikaatiot muuttujien lisäksi myös predikaattien (siis relaatioiden) yli. [8, s.636] [19, s.132]

Fregen tavoite oli muotoilla käsitekirjoituksen avulla puhtaasti looginen teoria luonnollisista luvuista ja osoittaa sitten, että kaikki matematiikka palautuisi logiikkaan. Tätä näkökantaa kutsutaan *logisismiksi*. Vaikeaksi ongelmaksi nousi luonnollisten lukujen määritteleminen pelkästään loogisin käsittein [5, s.63]. Ajan saatossa tälle on esitetty erilaisia vaihtoehtoja, esimerkiksi *Principia Mathematican* järjestelmä tai RA-kieli. Kuten Gödelin epätäydellisyyslause sanoo, niin Fregen unelmaa kaiken matematiikan palauttamisesta logiikkaan ei voida täysin saavuttaa.

Fregen käsitekirjoitus ei täyttänyt myöskään Leibnizin haavetta. Sen avulla voidaan kyllä pyrkiä selvittämään, onko jokin lause pääteltävissä lähtöoletuksista, mutta se ei tarjoa menetelmää, jonka avulla tämä kysymys voitaisiin ratkaista yksinkertaisesti laskemalla. Jos loogikko ei löydä päättelyjonoa, joka johtaa todistettavaan lauseeseen, niin ei voida tietää johtuuko se vain sinnikkyyden puutteesta, vai eikö päättelyjonoa todellakaan ole. [5, s.62]

7.2 Hilbertin ratkaisuongelma

Vuonna 1900 pidettiin Pariisissa kansainvälinen matemaatikkokongressi, jossa matemaatikko David Hilbert piti kuuluisan esitelmän. Se käsitteli kahtakymmentäkolmea matemaattista ongelmaa, jotka olivat hänen mielestään merkittäviä, mutta joiden ratkeaminen ei ollut vielä näköpiirissä. Nämä ongelmat ovat olleet suunnattoman hedelmällisiä matematiikan tieteenalan kehityksen kannalta.

Hilbert oli hyvin optimistinen ratkaisujen löytymisen suhteen, sillä hän ajatteli kaikkien tarkasti rajattujen matemaattisten ongelmien olevan ratkaistavissa. Suurin osa näistä ongelmista onkin ratkaistu, mutta muutamasta on voitu osoittaa, ettei niille ole olemassa ratkaisua. Ongelma numero kaksi koskikin juuri lukuteoreettisen formalismin ristiriidattomuutta, jota Gödelin lauseen mukaan ei voi todistaa formaalisti. [5, s.96-98]

Ongelma, johon Churchin lause itse asiassa liittyy on Hilbertin kymmenes ongelma, joka kysyy voidaanko löytää yleistä ratkaisumenetelmää Diofantoksen yhtälöille. Myöhemmin Hilbert laajensi tämän ajatuksen koskemaan koko lukuteorian formalismia teoksessa *Grundzüge der Theoretischen Logik* 1928. Se sai nimekseen *Entscheidungsproblem* eli ratkaisuongelma. [11, s.691] [21, Luku 11]

Ratkaisuongelman voisi muotoilla seuraavasti: Voidaanko kehittää joukko sääntöjä, joita tarkasti seuraamalla saadaan äärellisessä ajassa varma vastaus mihin tahansa matemaattiseen ongelmaan? Toisin sanoen, voidaanko matemaatikko korvata laskijalla? Tämä tuo mieleen Leibnizin *calculus ratiocinatorin*. Esimerkki aksioomajärjestelmästä, jolle tällainen ratkaisumenetelmä voidaan esittää on propositiologiikka. Sen jokaisen kaavan tapauksessa voidaan muodostaa totuustaulu, jonka avulla selviää onko kaava teoreema vai ei. Hilbert halusi siis löytää vastaavanlaisen ratkaisumenetelmän aksioomajärjestelmälle, joka pystyy imitoimaan lukuteoriaa.

7.3 Churchin teesi ja Churchin lause

Alonzo Church vastasi Hilbertin ratkaisuongelmaan kieltävästi 1936, kuten johdannossa esiteltiin. Tämä tulos saavutettiin pitkälti Churchin ja Kleenen yhteistyön tuloksena ja taustalla vaikuttivat Gödelin metamatemaattiset ideat. Church siis esitti tehokkaan laskettavuuden määritelmäksi rekursiivista laskettavuutta ja osoitti sen jälkeen, että ei ole olemassa rekursiivista menetelmää, jonka avulla mielivaltaisen matemaattisen lauseen todistettavuuden voi selvittää formalisoidussa lukuteoriassa.

Mielenkiintoista on, että mm. Gödel ja Emil Post, joka oli myös tutkinut laskettavuuteen liittyviä asioita, eivät vakuuttuneet Churchin todistuksesta tai tarkemmin ottaen hänen tehokkaan laskettavuuden määritelmästään. Tämä johtui siitä, että Churchin teesi ei sinänsä viittaa mitenkään varsinaiseen laskemiseen vaan abstraktiin formaaliin teoriaan rekursiivisista funktioista ja λ-laskennasta.

Miten voitaisiin vakuuttua siitä, että ei ole olemassa vielä toisenlaisia laskettavuuden muotoja, joita soveltamalla ratkaisuongelmaan voitaisiin antaa myönteinen vastaus? [8, s.652]

7.4 Turingin koneet

7.4.1 Mikä on kone?

" Voidaan näet kyllä kuvitella kone niin rakennetuksi, että se lausuu sanoja, vieläpä niin, että se lausuu joitakin ruumiinliikkeisiin soveltuvia sanoja, jotka saavat aikaan jotain muutosta sen elimissä: esimerkiksi siten, että kun kosketamme sen jotakin kohtaa, se kysyy mitä sille tahdotaan sanoa, ja jotakin toista kohtaa kosketettuamme huutaa tuntevansa kipua, ja muuta sen tapaista. Mutta se ei voi milloinkaan järjestää sanojansa eri tavoin vastatakseen järkevästi kaikkeen, mitä sen läsnäollessa lausutaan, niin kuin kaikkein tylsimmätkin ihmiset osaavat tehdä. "[6]

-Descartes-

Tämä on katkelma Descartesin 1600-luvulla kirjoittamasta teoksesta Metodin esitys. Niihin aikoihin erilaisten mekaanisten koneiden keksiminen ilmeisesti innoitti tällaisiin ajatuksiin. Descartes ymmärsi tätä kautta myös eläimet sekä ihmisruumiin monimutkaisiksi orgaanisiksi koneiksi. Ihmisellä on kuitenkin hänen mukaansa myös järki ja vapaa tahto, jonka avulla ihminen voi ohjata ruumistaan. Descartes kuvaa järjen ja mekaanisen toiminnan eroa seuraavasti.

" ...järki on yleinen väline, jota voidaan käyttää kaikenlaisissa ennalta arvaamattomissa tilanteissa, mainitut elimet tarvitsevat sitä vastoin erityistä rakenteen suunnittelua jokaista eri tehtävää varten. "[6]

-Descartes-

Tämä ajatus koneen ja ihmisjärjen erosta on osuva ja se johdattaa meidät 1900-luvulla eläneen Alan Turingin ideaan laskentakoneista. Tosin Turing pystyi osoittamaan, että voidaan valmistaa ns. universaalikone. Se voi suoriutua hyvin erilaisista tehtävistä vaihdettavan ohjelmiston avulla, vaikka sillä onkin vain yksi pysyvä rakenne.

7.4.2 Turing ja ratkaisuongelma

On tunnettua, että laskentatehtävät ovat sellaisia, joita voidaan oppia toteuttamaan rutiininomaisesti. Tällainen laskeminen ei ole kovin luovaa työtä vaan tarkkojen sääntöjen rutiininomaista noudattamista, jossa olennaista on välttää ainoastaan huolimattomuusvirheitä. Jos ongelman ratkaisemiseksi on olemassa tällainen joukko yksinkertaisia sääntöjä sanomme, että ratkaisu on laskettavissa ja kyseistä sääntöjoukkoa kutsumme "algoritmiksi".

Nuori englantilainen matematiikan tutkija Alan Turing tutustui Gödelin epätäydellisyyslauseeseen 1935 M. H. A. Newmanin pitämällä matematiikan perusteita käsittelevällä luentosarjalla Cambridgessa. Gödelin lause loi vahvoja epäilyksiä Hilbertin ratkaisuongelmaa kohtaan matemaatikkojen piirissä. Vaikutti siltä, että ei olisi mahdollista löytää ratkaisualgoritmia kaikille matemaattisille ongelmille. Luennon innoittamana Turing tarttui haasteeseen ja pyrki löytämään todistuksen, jonka mukaan tällaista algoritmia ei tosiaankaan ole.

Ongelmaksi nousi, että algoritmin käsite ei ollut tarkkarajainen. Millaisia sääntöjä algoritmi voisi sisältää? Entä miten niitä voi soveltaa laskennan edetessä? Turing alkoi lähestyä näitä kysymyksiä filosofisen työskentelyn kautta. Hän ajatteli tilannetta, jossa suoritetaan jotakin rutiininomaista laskutoimitusta. Toisin kuin Church ja Kleene hän ei keskittynyt niinkään sovellettaviin sääntöihin vaan siihen mitä laskija itse asiassa tekee laskiessaan. Hän karsi laskutoimituksen suorittamisesta pois kaiken epäolennaisen ja sai muotoiltua abstraktin ja hyvin yksinkertaisen laskennallisuuden mallin. Tätä laskennan mallia toteuttavaa laskijaa kutsutaan nykyään Turingin koneeksi. [5, s.150]

Tämä tapahtui samoihin aikoihin, kun Church ja Kleene pohtivat samaa ongelmaa Atlantin toisella puolella, mutta Turing ei tiennyt tästä. Hänkin onnistui vastaamaan kielteisesti Hilbertin ratkaisuongelmaan, tosin oman määritelmänsä kautta. Toisin sanoen, ei ole olemassa Turingin konetta, joka ratkaisisi minkä tahansa lukuteoreettisen ongelman.[5, s.168]

Turing julkaisi tuloksensa 1936 vain hiukan myöhässä, sillä Church oli jo ennättänyt julkaista oman artikkelinsa aiheesta. Kun Turingille selvisi, että Church oli saavuttanut samantapaisen tuloksen, hän pystyi nopeasti osoittamaan, että Churchin tehokkaan laskettavuuden määritelmä ja hänen määritelmänsä ovat ekvivalentit. [9, s.23-25] [13, s.363-376]

7.4.3 Mitkä ovat laskemisen olennaiset piirteet?

Järkeily etenee Turingia mukaillen seuraavasti. Ajatellaan laskijaa[11] suorittamassa jotakin annettua laskutoimitusta. Yleensä laskutoimitus suoritetaan paperin ja kynän avulla. Se millaista "kynää" ja millaista "paperia" käytetään ei tietenkään ole kovin olennaista. Usein paperille tehdään merkintöjä useammalle riville laskemisen edetessä, mutta Turing havaitsi, että tämäkään ei ole olennaista. Voidaan siis olettaa, että laskutoimitus suoritetaan yksiulotteisella ruudutetulla paperinauhalla, jossa on yksi merkki ruutua kohti.

Oletetaan myös, että niiden eri merkkien lukumäärä, joita voidaan paperille kirjoittaa, on äärellinen. Jos näin ei olisi, niitä ei voisi erottaa toisistaan. Tämä ei kuitenkaan rajoita laskemista, sillä uutta merkkiä tarvittaessa voidaan käyttää sen sijaan sopivaa merkkijonoa. Olennaista on, että *välittömästi tunnistettavia* merkkejä on vain äärellinen määrä.

Mitä laskija tekee laskemisen edetessä? Ainakin hänen täytyy pystyä lukemaan paperilla olevat merkit. Turing havaitsi, että laskemisen mahdollisuuksia ei rajoiteta, jos oletetaan, että laskija voi lukea (tai tarkastella) vain yhtä merkkiä kerrallaan. Lisäksi laskija voi kirjoittaa uusia merkkejä paperille ja pyyhkiä entisiä pois.

Laskemisen etenemistä määrää jokaisessa laskemisen vaiheessa laskijan senhetkinen "mielentila" ja nauhan tarkasteltavassa ruudussa oleva symboli. Näin ajateltuna laskeminen koostuu tarkoin määritellyistä erillisistä askelista[12]. Turing pystyi edelleen perustelemaan, että laskennalliset mahdolli-

[11]Sana *computer* tarkoitti noihin aikoihin laskutoimituksia suorittavaa ihmistä.

[12]Konetta, jonka toiminta perustuu tällaisiin "hyppäyksiin" diskreettien tilojen välillä,

suudet pysyvät samoina, vaikka laskemisessa sallitut toiminnot rajoitetaan muutamiin alkeistoimintoihin. Jokaisessa laskemisen vaiheessa laskija voi...

- ... poistaa tarkasteltavassa ruudussa olevan merkin.
- ... kirjoittaa tarkasteltavaan ruutuun uuden merkin.
- ... siirtää tarkastelupistettä yhden ruudun oikealle tai vasemmalle.
- ... muuttaa mielentilaansa.
- ... lopettaa laskemisen.

Mielentilat vastaavat toimintaohjeita, joiden mukaan laskija toimii. Turing kutsui niitä "m-konfiguraatioiksi". Ne määräävät mitä tehdään laskemisen seuraavalla askeleella. On hyvä huomata, että laskija ei tarvitse minkäänlaista kokonaiskuvaa siitä, mitä hän on tekemässä. Riittää, kun hän "sokeasti" seuraa sääntöjä. Turing kirjoittaa:

" *... 'mielentilan' käyttöönotolta vältytään tarkastelemalla sen fyysisempää ja täsmällisempää vastinparia. Laskijan on aina mahdollista keskeyttää työnsä, mennä pois ja unohtaa koko juttu, ja tulla myöhemmin takaisin ja jatkaa uudelleen. Näin tehdessään hänen tulee jättää jälkeensä muistilappu (joka on kirjoitettu jollakin tunnetulla menetelmällä), joka selittää, kuinka työtä tulee jatkaa. Tämä lappu on 'mielentilan' vastinpari. Oletamme, että laskija työskentelee niin katkonaisesti, ettei hän koskaan suorita yhdellä kertaa enempää kuin yhden työvaiheen. Muistilapun täytyy olla sellainen, että hän pystyy suorittamaan yhden työvaiheen ja kirjoittamaan seuraavan lapun. Siten laskennallisen prosessin määrää kullakin hetkellä ainoastaan muistilappu ja nauhalla olevat symbolit...* "[9, s.21]

Millaisia m-konfiguraatiot eli lapulla olevat ohjeet sitten voivat olla? Turingin ajatusta seuraten, ne eivät voi olla muuta kuin ohjeita siitä, mikä alkeistoiminto tehdään, kunkin tarkasteltavan symbolin tapauksessa. On tietysti olennaista, että lapulle koodattu konfiguraatio on kirjoitettu merkinnöillä, jotka laskija tuntee tarkasti ja osaa niitä noudattaa.

kutsutaan digitaaliseksi.

Kuhunkin laskentaprosessiin sallitaan vain äärellinen määrä konfiguraatioita, jotta ne voidaan selvästi erottaa toisistaan. Kutsutaan näitä konfiguraatioita jatkossa yksinkertaisesti laskijan tiloiksi. Käytetään tilan merkitsemiseen seuraavanlaista formalismia.

$$X : a \to b, \text{[siirto]}, Y$$

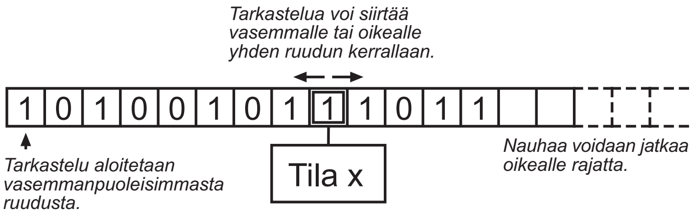
Tässä X on määriteltävän tilan indeksi. a on tarkasteltavassa ruudussa oleva symboli. b on symboli, joka piirretään a:n paikalle. Kohtaan [siirto] kirjoitetaan O, V tai $\star$, missä O tarkoittaa siirtymistä ruudun verran oikealle, V vastaavasti vasemmalle ja $\star$ tarkoittaa laskemisen lopettamista. Y on sen tilan indeksi, johon siirrytään näiden toimintojen jälkeen. Voidaan merkitä myös $a \to a$, jolloin merkkiä ei vaihdeta, sekä $a \to \square$, jolloin a pyyhitään ja ruutu jätetään tyhjäksi.

Olennaista koneen suoriutumisen kannalta on se, että nauhalla olevien ruutujen tarkastelu aloitetaan aina samasta paikasta. Koska merkkejä luetaan yleensä länsimaissa vasemmalta oikealle, niin aloitetaan koneen toiminta vasemmanpuoleisimmasta ruudusta. On myös tärkeää, että nauhan pituutta ei rajoiteta. Jos sen pituus olisi rajallinen, niin se vaikuttaa olennaisesti siihen, mitä laskutoimituksia voidaan suorittaa. Sovitaan siis, että nauhaa voidaan tarpeen vaatiessa jatkaa rajattomasti oikealle.

Turing näytti esimerkinomaisesti, miten voidaan konstruoida koneita, jotka tuottavat Neperin luvun, π:n ja eräiden muiden matematiikassa tyypillisesti esiintyvien reaalilukujen binäärisiä kuvauksia.

Turingin koneet voidaan myös toteuttaa fyysisinä koneina. Descartesin määritelmä koneelle osui naulan kantaan, sillä jokainen Turingin kone tosiaan vaatii erillistä suunnittelua kutakin tehtävää varten. Tämä suunnittelu ilmenee abstraktilla tasolla listana sen tiloista. Tosin voidaan rakentaa universaali Turingin kone, joka voi simuloida mitä tahansa Turingin konetta. Se ottaa vastaan syötteenä nauhan, jossa ensin on simuloitavan Turingin koneen koodi ja sen jälkeen tälle koneelle annettava syöte. [21] [5, s.150-167] [9, s.16-21]

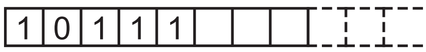
Alla on kuva Turingin koneesta, joka käsittelee binäärilukuja.



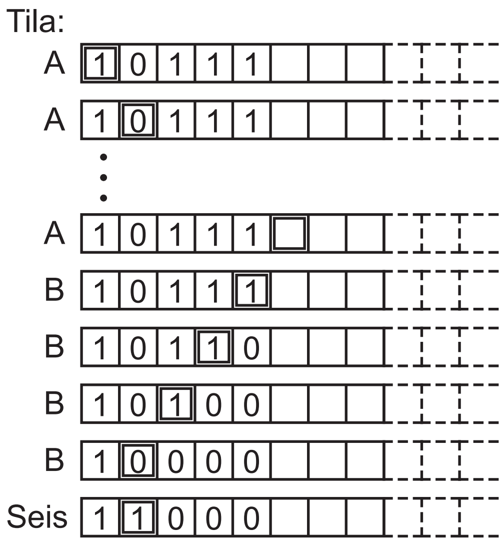
Esimerkki:
Annetaan laskijalle tehtäväksi kasvattaa annettua lukua yhdellä (vertaa seuraajafunktioon). Annetaan luku binäärimuodossa eli jonona ykkösiä ja nollia. Annetaan laskijalle seuraavat säännöt:

$A : 0 \to 0,\ O,\ A$	$B : 0 \to 1,\ \star$
$A : 1 \to 1,\ O,\ A$	$B : 1 \to 0,\ V,\ B$
$A : \square \to \square,\ V,\ B$	$B : \square \to 1,\ \star$

Annetaan luku ruudutetulla nauhalla *binäärimuodossa* seuraavasti.



Luku vastaa kymmenjärjestelmän lukua 23. Laskija aloittaa laskemisen normaalisti vasemmanpuoleisimmasta ruudusta. Sääntöjen seuraaminen etenee seuraavalla tavalla.



Saatu luku vastaa lukua 24, joten laskija kasvatti annettua lukua yhdellä.

7.4.4 Churchin teesi ja Turingin teesi

Lausetta: funktio on tehokkaasti laskettava jos ja vain jos se on laskettavissa Turingin koneella, voi hyvällä syyllä kutsua Turingin teesiksi.

Churchin ja Turingin teesit poikkeavat toisistaan siinä, miten ne lähestyvät laskemista. Churchin määritelmä kuuluu matemaattisen formalismin maailmaan, kun taas Turing ujuttaa nerokkaasti mukaan fysikaalisen todellisuuden. Gödel vakuuttuikin paremmin Turingin laskettavuutta koskevasta teesistä juuri tästä syystä. [8, s.652] [9, s.25]

Määritelmien erilaisuuden huomioon ottaen on kenties yllättävää, että ne määräävät saman funktiojoukon. Tämän lisäksi on myös muita formalismeja, jotka ovat ekvivalentteja rekursiivisten funktioiden kanssa. Näistä yksi on tietysti Churchin ja Kleenen λ-laskenta. Lisäksi näitä ovat Emil Postin kehittämä laskettavuuden malli sekä kombinatorinen logiikka [14, s.206]. Olisi erikoista jos kyseessä olisi pelkkä yhteensattuma. Erilaisten laskettavuuden määritelmien keskinäinen vastaavuus onkin yksi argumentti sen puolesta, että kyseiset määritelmät onnistuvat kuvaamaan kattavasti sen mitä voi kutsua mekaaniseksi tai tehokkaaksi laskemiseksi.

Churchin teesistä käytetään usein nimitystä Church-Turing teesi. Nykyään teesiin sisällytetään kaikki edellä mainitut ekvivalentit laskettavuuden mallit ja jos halutaan todeta jonkun funktion rekursiivisuus, niin riittää osoittaa, että sille on olemassa ratkaisualgoritmi missä tahansa formalismissa. Kun algoritmi on löydetty, niin saatetaan sanoa, että se on rekursiivinen Churchin teesin nojalla (”argument by Church’s thesis”).[8, s.653]

Tämä on jossain määrin harhaanjohtavaa, jos ajatellaan Churchin teesin tarkoittavan sitä, mitä se alunperin tarkoitti. Pitäisihän viitata tuloksiin, joiden mukaan eri laskettavuuden määritelmät ovat ekvivalentteja rekursiivisten funktioiden kanssa, eikä siihen, että mikä tahansa *laskettava* funktio on rekursiivinen.

On huomattava ero Church-Turing teesin ja sen välillä, että Turingin laskettavuus ja Churchin laskettavuus vastaavat toisiaan. Church-Turing teesiä ei voida todistaa formaalisti, koska se viittaa intuitiiviseen laskettavuuden käsitteeseen, kun taas ekvivalensille voidaan esittää formaali todistus.

Church-Turing teesiä, toisin kuin eri formalismien ekvivalenssia, voi siis helpommin epäillä. Onhan mahdollista, että keksittäisiin laskentamenetelmä, jolla rekursiivisen laskettavuuden rajat ylitettäisiin. Jotkut ovat esimerkiksi ehdottaneet, että kehitteillä olevat kvanttitietokoneet voisivat ylittää Turing-laskettavuuden rajat. Toiset taas näkevät, että kvanttitietokoneet lisäisivät ainoastaan laskentatehoa. Näitä kysymyksiä käsitellään mm. Martin Davisin artikkelissa *The Myth of Hypercomputation* [7, s.195-210]. Katso myös [7, s.213-238].

Toisaalta, kuten sanottu, useat argumentit tukevat Church-Turing teesiä [13, s.318-323]. Sitä tukee myös se seikka, että kaikki nykyiset tietokoneet ovat Turingin koneita.

8 Pohdintaa

Churchin lauseen merkitystä avataan joskus sanomalla, että sen mukaan ei ole olemassa *laskennallista menetelmää*, jonka avulla voidaan selvittää mielivaltaisesta RA-kaavasta[13], onko se teoreema vai ei [19, s.127]. Tällöin kuitenkin täytyy olettaa, että Church-Turing teesi pätee, vaikka sitä ei suoraan sanottaisikaan.

Jos halutaan olla tarkkoja, niin olisi edellisen luvun nojalla parempi viitata Churhin lauseen yhteydessä esimerkiksi Turing-laskettavuuteen. Voidaan nimittäin sanoa varmuudella, että ei ole olemassa Turingin konetta, jonka avulla voidaan selvittää mielivaltaisesta RA-kaavasta, onko se teoreema vai ei. Tämä viittaa suoraan siihen laskettavuuteen, joka ihmisillä on käytettävissään esimerkiksi allekkainlaskun tai tietokoneiden muodossa, eikä tarvitse olettaa jossain määrin epävarmaa Church-Turing teesiä todeksi.

Miten Churchin lause ja Gödelin todistama lukuteorian formalismin epätäydellisyys sitten liittyvät toisiinsa? Turing huomautti, että jos olisikin todistettu päinvastaista kuin Gödelin epätäydellisyyslause eli täydellisyys, niin ratkaisuongelmaan löytyisi helposti ratkaisu [21, Luku 11].

Oletetaan lukuteorian formalismin olevan täydellinen, eli että jokainen validi kaava on teoreema. Jokaiselle suljetulle kaavalle $\mathbf{A}$ pätee määritelmän 4.21 nojalla joko $\mathbb{N} \models \mathbf{A}$ tai $\mathbb{N} \models \neg\mathbf{A}$. Täydellisyydestä seuraa, että vastaavasti joko $\mathbf{A}$ tai $\neg\mathbf{A}$ on teoreema. On mahdollista tehdä algoritmi R, joka muodostaa järjestyksessä kaikki lukuteorian formalismin teoreemat[14]. Jos nyt kysytään onko joku mielivaltainen suljettu kaava $\mathbf{A}$ teoreema, voitaisiin muodostaa teoreemoja algoritmilla R niin pitkään kunnes vastaan tulisi joko $\mathbf{A}$ tai $\neg\mathbf{A}$. Jos algoritmi muodostaisi kaavan $\mathbf{A}$, niin tietäisimme sen olevan teoreema. Jos se muodostaisi kaavan $\neg\mathbf{A}$, niin eheyslauseen (lause 5.1) nojalla $\neg\mathbf{A}$ on validi. Nyt määritelmän 4.21 perusteella $\mathbf{A}$ ei ole validi, mistä taas seuraa eheyslauseen nojalla, että $\mathbf{A}$ ei ole teoreema. Näin voitaisiin siis selvittää mielivaltaisesta suljetusta kaavasta, onko se teoreema vai ei. Tähän voisi mennä aikaa, mutta teoriassa olisi varmaa, että vastaus tulisi löytymään.

[13]. . . tai muunkaan lukuteorian formalismin kaavasta. . .

[14]Karkean esimerkin tällaisesta algoritmista saa soveltamalla lauseen 6.2 funktiota f.

Mutta kuten tiedetään lukuteorian formalismi ei ole täydellinen. On siis olemassa suljettu kaava U, joka on validi, mutta ei teoreema [16, s. 143]. Koska U on validi, niin $\neg U$ ei ole validi, mistä seuraa eheyslauseen nojalla, että $\neg U$ ei myöskään ole teoreema. Vaikka algoritmilla R muodostettaisiin teoreemoja mielivaltaisen pitkän ajan, ei silti voitaisi varmistua onko kaava U, tai sen johdannainen, teoreema vai ei, sillä sille tai sen negaatiolle ei ole olemassa päättelyjonoa. Algoritmi ei kuitenkaan kerro tätä ja näin ollen jouduttaisiin ikuisesti odottelemaan, josko päättelyjono kuitenkin löytyisi.

Tämä odottelu liittyy myös siihen, että pelkästä teoreemasta ilman päättelyjonoa ei voi suoraan nähdä kuinka monta päättelyaskelta sen saavuttamiseksi korkeintaan tarvittaisiin, sillä päättelysääntö *modus ponens* lyhentää kaavoja [16, s.139]. Mutta tietysti, jos kaavasta näkisi kuinka pitkän päättelyjonon se korkeintaan tarvitsee, niin olisi jo kyse primitiivirekursiivisesta ongelmasta. Churchin lauseen nojalla kyseessä ei ole edes rekursiivinen prosessi.

Palautetaan mieleen sukupuuesimerkki. Mielivaltaisen kaavan teoreemuuden selvittäminen vastaa seuraavaa tilannetta. Halutaan selvittää, onko henkilö X Joosepin jälkeläinen. Sukupolvia on ääretön määrä, eikä meillä ole mitään tietoa kuinka monen sukupolven päässä Jooseppi korkeintaan on. Lisäksi puuttuisi tieto siitä, onko X Joosepin jälkeläinen vai ei. Nyt ikuisen sukututkimuksen vaara on ilmeinen.

On syytä panna merkille, että Gödelin epätäydellisyyslauseesta ei kuitenkaan suoraan seuraa kielteistä vastausta Hilbertin ratkaisuongelmaan. Voisi nimittäin periaatteessa olla olemassa algoritmi, joka kykenee tunnistamaan myös U:n kaltaiset riippumattomat kaavat. Jos siis sille annettaisiin syötteeksi riippumaton suljettu kaava, niin se voisi yksiselitteisesti ilmoittaa, ettei se ole teoreema. Muut tapaukset algoritmi voisi selvittää samaan tapaan kuin täydellisyyden tapauksessa. Churchin lause kuitenkin osoittaa, että tällaista algoritmia ei ole olemassa.

Onko Churchin lauseella sitten käytännössä mitään merkitystä? Jos tietokone laitettaisiin laskemaan lukuteorian teoreemoja algoritmilla R, niin ehkä laskettavien teoreemojen ulkopuolelle jäisi vain outoja U:n tapaisia kaavoja? Eikö voisi olla niin, että kaikki mielenkiintoiset ja mahdollisesti hyödylliset teoreemat olisivat laskettavissa?

Käytännön esteet tulevat teoreemojen laskemisessa kuitenkin nopeasti vastaan. Oletetaan, että joku merkittävä lukuteorian lause, kuten Goldbachin hypoteesi olisi teoreema, mutta emme tietäisi sitä. Laitetaan nyt tietokoneet laskemaan kaikkia RA-kielen teoreemoja lauseen 6.2 tapaisella karkealla algoritmilla. Erilaisia vaihtoehtoja teoreemojen muodostamiseen on niin valtavasti, että maailmankaikkeuden arvioitu elinikä on pian käytetty loppuun, ennen kuin päästään monimutkaisempiin teoreemoihin asti, eikä Goldbachin hypoteesin mahdollinen päättelyjono ole ilmeisesti kovin lyhyt, onhan sen ratkaisuyrityksiin käytetty huomattavan paljon aikaa.

Michael J. Beeson kirjoittaa Churchin lauseen käytännön merkityksestä artikkelissaan *The Mechanization of Mathematics* [7, s.91-92]. Hän listaa muutamia Churchin lauseen ”porsaanreikiä”, joiden valossa matematiikan teoreemoja ratkovaa algoritmia kannattaa vielä etsiä.

- Onko olemassa mielenkiintoisia aksioomajärjestelmiä X siten, että X:lle on muotoiltavissa algoritmi, joka pystyy aina ratkaisemaan, seuraako kaava $\mathbf{A}$ X:stä vai ei?

- Onko olemassa mielenkiintoisia algoritmeja f aksioomajärjestelmälle X, jotka joissakin tapauksissa ratkaisevat seuraako kaava $\mathbf{A}$ X:stä. Jos tällainen algoritmi olisi, sen avulla voitaisiin selvittää joitakin ongelmia, joihin ei tiedetty vastausta aiemmin.

- Vaikka algoritmi f toimisikin ainoastaan jollekin tarkasteltavalle X:lle, se voisi silti tuottaa vastauksia joihinkin matemaattisiin ongelmiin, joihin ei aiemmin ole vastattu.

Beeson käsittelee näitä eri tapauksia tarkemminkin, mutta jätän tarkastelun suosiolla tämän tutkimuksen ulkopuolelle.

Viitteet

[1] Alonzo Church. Correction to a note on the entscheidungsproblem. *The Journal of Symbolic Logic*, 1(3), September 1936.

[2] Alonzo Church. A note on the entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1), March 1936.

[3] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2), April 1936.

[4] Alonzo Church. *Introduction to Mathematical Logic*. Annals of Mathematics Studies. Princeton University Press, London, 1944.

[5] Martin Davis. *Tietokoneen esihistoria Leibnizista Turingiin*. Art House Oy, Helsinki, 2003.

[6] Rene Descartes. *Metodin esitys*. Teoksia ja kirjeitä. WSOY, Juva, 1994. Käännös: Hollo, J.A.

[7] Christof Teuscher (ed.). *Alan Turing : life and legacy of a great thinker*. Springer, Berlin, 2004.

[8] Ivor Grattan-Guinness (ed.). *Companion Encyclopedia of the History and Philosophy of the Mathematical Sciences*. Routledge, Cornwall, 1994.

[9] Andrew Hodges. *Turing*. Otava, Keuruu, 1997.

[10] John E. Hopcroft and Jeffrey D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley, 1969.

[11] J. P. Jones and Y. V. Matijasevic. Proof of recursive unsolvability of hilbert's tenth problem. *The American Mathematical Monthly*, 98(8), October 1991.

[12] Stephen Cole Kleene. Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1), January 1943.

[13] Stephen Cole Kleene. *Introduction to Metamathematics*. Bibliotheca Mathematica. North-Holland Publishing Co., Amsterdam, fifth edition, 1967.

[14] Dexter C. Kozen. *Automata and Computability.* Undergraduate Texts in Computer Science. Springer, 1997.

[15] Lassi Kurittu. Johdatus logiikkaan. Jyväskylän yliopisto.

[16] Lassi Kurittu. Matemaattinen logiikka. Jyväskylän yliopisto.

[17] Piergiorgio Odifreddi. *Classical Recursion Theory.* Studies In Logic and The Foundations of Mathematics. North-Holland, Amsterdam, paperback, 2nd impr. edition, 1999.

[18] Piergiorgio Odifreddi. *Classical Recursion Theory Volume II.* Studies In Logic and The Foundations of Mathematics. North-Holland, Amsterdam, first edition, 1999.

[19] Joel W. Robbin. *Mathematical Logic: A First Course.* W. A. Benjamin, INC., 1969.

[20] Barkley Rosser. Extensions of some theorems of godel and church. *Journal of Symbolic Logic*, 1(3), September 1936.

[21] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, (42), 1936. s. 230-265.

[22] Jouko Väänänen. *Matemaattinen logiikka.* Gaudeamus, Helsinki, 1987.

[23] Alfred Norton Whitehead and Bertrand Russell. *Principia Mathematica.* University Press, Cambridge, 1910.

9 Liitteet

9.1 Teoreeman 4.16 todistus

Tulee siis osoittaa: $\vdash S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A}) \rightarrow \exists \mathbf{x_n} \mathbf{A}$, jos sijoitus ei sido muuttujaa $\mathbf{x_n}$. Päättelylauseen [15, s.14,46] nojalla riittää osoittaa, että $S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A}) \vdash \exists \mathbf{x_n} \mathbf{A}$.

Todistus:

$\mathbf{A}_1 = S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A})$	Oletus
$\mathbf{A}_2 = \forall \mathbf{x_n}(\neg\mathbf{A}) \rightarrow S^{\mathbf{x_n}}_{\mathbf{t}}(\neg\mathbf{A})$	Aksiooma 4 ja oletus
$\mathbf{A}_3 = \forall \mathbf{x_n}(\neg\mathbf{A}) \rightarrow \neg S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A})$	$\mathbf{A}_2$ ja perustelu 1.
$\mathbf{A}_4 = \neg\neg S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A}) \rightarrow \neg\forall \mathbf{x_n}\neg\mathbf{A}$	$\mathbf{A}_3$ ja perustelu 2.
$\mathbf{A}_5 = S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A}) \rightarrow \neg\forall \mathbf{x_n}\neg\mathbf{A}$	$\mathbf{A}_4$ ja aksiooma 3
$\mathbf{A}_6 = S^{\mathbf{x_n}}_{\mathbf{t}}(\mathbf{A}) \rightarrow \exists \mathbf{x_n}\mathbf{A}$	

Perustelut:

1. Kaava $\neg\mathbf{A}$ on lyhenne kaavasta $\mathbf{A} \rightarrow f$. Jos sijoitamme tähän kaavaan termin $\mathbf{t}$ jonkin muuttujan paikalle, niin muutos voi tapahtua ainoastaan kaavan osassa $\mathbf{A}$, sillä osassa $\rightarrow f$ ei ole muuttujia.

2. Tässä käytetään propositiologiikan teoreemaa

$$\vdash [p \rightarrow g] \rightarrow [\neg q \rightarrow \neg p].$$

Sen, että kyseinen kaava on teoreema, voi tarkistaa vaikkapa totuustaululla.

$\square$

9.2 Rekursiiviset funktiot

Rekursiivisten funktioiden määrittelyssä käytetään samoja merkintöjä kuin RA-kielessä. On huomattava, että termin käsite eroaa RA-kielen vastaavasta ja ettei funktiokirjain ei ole sama asia kuin funktiosymboli.

Näiden formaalien merkintöjen semantiikka määritellään soveltuvin osin samoin kuin RA-kielen semantiikka.

Määritelmä 9.1 (Funktiokirjaimet)
Vakiot $\mathbf{f}^n_i$ ovat n-paikkaisia funktiokirjaimia kaikilla $i \in \mathbb{N}$. Jos kirjoitetaan $\mathbf{f}$ ilman indeksejä, niin tarkoitetaan mielivaltaista funktiokirjainta, jonka paikkaluku nähdään tilanteen mukaan.

Määritelmä 9.2 (Termit)

- $\mathbf{0}$ *on termi.*

- $\mathbf{x_n}$ *on termi kaikilla* $n \in \mathbb{N}$.

- *Jos* $\mathbf{t}$ *on termi, niin* $S(\mathbf{t})$ *on termi. (Huom.* $\mathbf{k}_n$ *on termi kaikilla* $n \in \mathbb{N}$.*)*

- *Jos* $\mathbf{t}_1, \ldots, \mathbf{t}_n$ *ovat termejä, niin* $\mathbf{f}_i^n(\mathbf{t}_1, \ldots, \mathbf{t}_n)$ *on termi kaikilla* $i \in \mathbb{N}$.

- *Mikään muu ei ole termi.*

Määritelmä 9.3 (Yhtälöt)
Olkoon $\mathbf{t}$ *ja* $\mathbf{s}$ *termejä ja* $\mathbf{t}$ *muotoa* $\mathbf{f}(\mathbf{t}_1, \ldots, \mathbf{t}_n)$*, missä* $\mathbf{f}$ *on* n*-paikkainen funktiokirjain ja* $\mathbf{t}_i$ *termi, joka ei sisällä muita funktiokirjaimia* $i = 1, \ldots, n$. *Sanotaan, että* $\mathbf{t} \cong \mathbf{s}$ *on yhtälö.*

Määritelmä 9.4 (Sijoitussääntö)
Olkoon $\mathbf{e_1}$ *yhtälö, joka sisältää muuttujasymbolin vähintään yhden* $\mathbf{x_n}$*, missä* $n \in \mathbb{N}$. *Olkoon* $\mathbf{e_2}$ *yhtälö, joka sadaan, kun jokaisen* $\mathbf{x_n}$*:n paikalle sijoitetaan numeraali* $\mathbf{k}_m$*, missä* $m \in \mathbb{N}$. *Sanotaan, että yhtälö* $\mathbf{e_2}$ *on johdettu yhtälöstä* $\mathbf{e_1}$ *sijoitussääntöä käyttämällä.*

Määritelmä 9.5 (Korvaussääntö)
Olkoon $\mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i}) \cong \mathbf{k}_m$ *yhtälö, missä* $m, n_j \in \mathbb{N}$ *kaikilla* $j = 1, \ldots, i$. *Olkoon* $\mathbf{t} \cong \mathbf{s}_1$ *yhtälö, joka ei sisällä muuttujasymboleita ja missä* $\mathbf{s}_1$ *sisältää vähintään yhden kerran termin* $\mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i})$. *Olkoon* $\mathbf{t} \cong \mathbf{s}_2$ *yhtälö, joka saadaan yhtälöstä* $\mathbf{t} \cong \mathbf{s}_1$*, kun korvataan yksi tai useampi termin* $\mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i})$ *esiintymä* $\mathbf{s}_1$*:ssä. Sanotaan, että yhtälö* $\mathbf{t} \cong \mathbf{s}_2$ *on johdettu yhtälöstä* $\mathbf{t} \cong \mathbf{s}_1$ *ja* $\mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i}) \cong \mathbf{k}_m$ *korvaussääntöä käyttämällä.*

Määritelmä 9.6
Olkoon E *joukko yhtälöitä. Sanotaan, että yhtälö* $\mathbf{e}$ *on johdettu* E*:stä (merkitään* $E \Rrightarrow \mathbf{e}$*), jos jokin seuraavista ehdoista pätee.*

1. $\mathbf{e} \in E$.

2. $E \Rrightarrow \mathbf{a}$ *ja* $\mathbf{e}$ *on saatu yhtälöstä* $\mathbf{a}$ *sijoitussääntöä käyttämällä.*

3. $E \Rrightarrow \mathbf{a}$, $E \Rrightarrow \mathbf{b}$ *ja* $\mathbf{e}$ *on saatu yhtälöistä* $\mathbf{a}$ *ja* $\mathbf{b}$ *korvaussääntöä käyttämällä.*

Määritelmä 9.7 (Rekursiiviset funktiot)
Olkoon E joukko yhtälöitä ja $\mathbf{f}$ i-paikkainen funktiosymboli siten, että $E \Rrightarrow \mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i}) \cong \mathbf{k}_m$, missä $m, n_j \in \mathbb{N}$ kaikilla $j = 1, \ldots, i$. Jos ei ole olemassa $t \in \mathbb{N}$ siten, että $t \neq m$ ja $E \Rrightarrow \mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i}) \cong \mathbf{k}_t$, niin sanomme, että funktio $f(x_1, \ldots, x_i) = v(\mathbf{f}(\mathbf{k}_{n_1}, \ldots, \mathbf{k}_{n_i})) = v(\mathbf{k}_m) = m$ on rekursiivinen.