Rekursiiviset funktiot

Laskennan teorian opintopiiri
Anna Haataja

Rekursiiviset funktiot ovat yksi matemaattisen logiikan ja tietojenkäsittelytieteen perustyökaluista. Ne perustuvat luonnollisestikin rekursion käsitteeseen ja ovat siis läheistä sukua ajatuksen tasolla induktiivisille päättelyille. Rekursiivisen funktion matemaattinen määritelmä nojaa ajatukseen alkutilanteesta ja sen pohjalta tehtävän rakennelman säännön tietämisestä. Eksaktisti tämä voidaan esittää seuraavaan tapaan:

Oletetaan, että $f(\mathbf{x})$ ja $g(\mathbf{x},y,z)$ ovat funktioita. Määritellään rekursiivinen funktio $h(\mathbf{x},y)$ siten, että

(i) $h(\mathbf{x},0)=f(\mathbf{x})$
(ii) $h(\mathbf{x},y+1)=g(\mathbf{x},y,h(\mathbf{x},y))$.

Määritelmän funktio f vastaa siis rekursion alkuarvoa ja g rekursiosääntöä. Funktion määritelmä matematiikassa vaatii yksikäsitteisyyttä kaikille funktion arvoille, joten nämä molemmat apufunktiot tulee määritellä ristiriidattomasti. Huomattavaa on myös, että määritelmässä muuttuja $\boldsymbol{x}$ on n-dimensioinen, eikä kerroinkuntaa ole määrätty reaalilukujen kunnaksi.

Määritelmän selventämiseksi käydään läpi pari helppoa esimerkkiä: Vakiofunktion $h(x,y)=1$ tapauksessa alkuarvo $f(x)=1$ ja rekursiosääntö $g(x,y,z)=z$. Näin vakiofunktio voidaan esittää määritelmän mukaisessa muodossa

(i) $h(x,0)=f(x)=1$
(ii) $h(x,y+1)=g(x,y,h(x,y))=h(x,y)$.

Lasketaan muutama arvo rekursion toimivuuden testaamiseksi: Selvästi $h(x,0)=1$, joten $h(x,1)=h(x,0+1)=h(x,0)=1$ ja $h(x,2)=h(x,1+1)=h(x,1)=h(x,0)=1$. Siis rekursiosäännön avulla kaikki funktion arvot palautuvat alkuarvoon, eli funktio on vakioarvoinen.

Toinen esimerkki on ranskalaisen matemaatikon Édouard Lucasin vuonna 1883 keksimä Hanoin torni -peli. Pelin taustana on legenda, jonka mukaan erään hindutemppelin munkeilla on ratkaistavanaan 64-osainen kultainen Hanoin torni. Pelin ratkaisusta seuraisi maailmanloppu. Onko tämä legenda totta vai ei, ei onneksemme ole suurikaan murhe, sillä jos munkit tekisivät yhden siirron sekunnissa, täydellisellä suorituksellakin heillä kuluisi siihen sangen kauan, peräti 2^{64}-1 sekuntia eli vajaat 600 miljardia vuotta. Kuinka tämä pystytään laskemaan? Voimme huomata, että Hanoin torni noudattaa rekursiota, mitä siirtojen määrään tulee. Klassisessa tapauksessa meillä on kolme perustaa, joille osia voidaan pinota. Tarkoituksena on siirtää osat ensimmäiseltä perustalta viimeiselle perustalle ilman, että yksikään osa jää isomman osan alle. Jos meillä on vain yksi torninkappale, tarvitsemme tietenkin vain yhden siirron pelin lopettamiseen. Siis siirtojen lukumäärää kuvaavan funktion $h=h(x,y)$ alkuarvo $f(x)=1$. Tapauksessa $y=2$ tarvitsemme kolme siirtoa: ensimmäiseksi pienempi osa keskelle, sitten isompi sen viereen ja viimeisenä pienempi osa isomman päälle. Muutaman kokeilun jälkeen huomaamme, että siirtojen määrä voidaan päätellä yhtä kerrosta pienemmän tornin siirtojen lukumäärästä kertomalla sen kahdella ja lisäämällä yhden: funktio $h(x,y+1)=2h(x,y)+1$, kun $y>1$. Rekursiosääntö $g(x,y,h(x,y))=2h(x,y)+1$, joten olemme saaneet määritelmän mukaisen rekursiivisen funktion kuvaamaan siirtojen määrää. Laskemalla siis funktion h muuttujan arvolla $y=64$ saamme ennustettua maailmanloppuun kuluvan ajan.

Muita tärkeitä esimerkkejä yksinkertaisista rekursiivisista funktioista ovat yhteenlasku, rajoitettu vähennyslasku ja eksponenttifunktio. Matematiikoiden suosikkiesimerkki on kuitenkin

Ackermannin funktio $A=A(x,y)$, jossa rekursiota esiintyy peräti kahdessa muuttujassa:

$A_0(y)=y+1$

$A_{x+1}(0)=A_x(1)$

$A_{x+1}(y+1)=A_x(A_{x+1}(y))$.

Ackermannin funktion kasvuvauhti ylittää eksponentiaalisen kasvunkin, joten sen arvoja on hyvin hankala käsitellä manuaalisesti. Arvoja voi löytää laskettuna esimerkiksi *Wolram Alphasta* tai taulukkokirjoista. Funktiota saatetaan käyttää tietokoneiden rekursion optimoinnin testaamiseen, mutta alunperin se syntyi esimerkiksi funktiosta, joka on rekursiivinen, muttei primitiivirekursiivinen.

Rekursiiviset funktiot voi myös määritellä Turingin koneen avulla, jolloin usein puhutaankin Turing-laskettavasta funktiosta. Määritelmä on muotoiltu seuraavasti:

Olkoon f n-paikkainen funktio. Funktio f on rekursiivinen, jos on olemassa Turingin kone A, jolle kaikilla $x_1, \dots, x_n, y$ pätee:

(i) Jos $f(x_1, \dots, x_n)=y$ ja A on käynnistetty annetulla alkuarvolla, niin A noudattaa funktion sääntöä ja pysähtyy vain ja ainoastaan arvolla y.

ja

(ii) Jos $f(x_1, \dots, x_n)$ on määrittelemätön, niin A ei voi koskaan pysähtyä, jos se on käynnistetty annetulla alkuarvolla.

Tässä määritelmässä on oleellista, että funktion rekursiivisuus on sidottu vain koneen toimintaan, eikä varsinaisesta rekursiosäännöstä sanota mitään. Alkuarvo on tärkeä tässäkin tapauksessa; se kertoo, minkä kiintopisteen mukaan kone rupeaa laskemaan muita arvoja, sillä toisin kuin ihminen, Turingin kone palauttaa rekursion aina alkuarvoon asti, eikä ensimmäiseen laskettuun arvoon.

Rekursiivisten funktioiden teorian tutkimus on oikeastaan alkanut vasta 1900-luvulla, mutta siitä huolimatta erityisesti matemaatikot ovat kehittäneet rekursiivisen funktion käsitteestä lukuisia johdannaisia. Tärkeimpiä näistä ovat primitiivirekursiiviset funktiot, jotka ovat rekursiivisten funktioiden perheen osajoukko, ja μ-rekursiiviset funktiot. Primitiivirekursiiviset funktiot on määritelty perheenä, johon kuuluvat vakiofunktiot, seuraaja- ja projektiofunktiot, ja joka on suljettu yhdistämisen ja rekursion suhteen. μ-rekursiiviset funktiot on on määritelty primitiivirekursiivisten funktioiden pohjalta lisämääreenä vain, että perhe on suljettu myös minimisaation suhteen. Tässä minimisaatio-operaattori μ etsii funktion pienintä nollakohtaa aloittaen haun nollasta. Määritelmistä voidaan huomata, että primitiivirekursiiviset funktiot ovat μ-rekursiivisten funktioiden ja μ-rekursiiviset funktiot rekursiivisten funktioiden osajoukko. Aikaisempi esimerkkimme Ackermannin funktiosta sijoittuu tässä hierarkiassa μ-rekursiivisten funktioiden joukkoon.

Rekursiivisista funktioista on paljon iloa matemaattisten ongelmien ratkaisemisessa. Niiden avulla ongelma voidaan palauttaa takaisin tunnettuun muotoon tai arvoon. Rekursiivisten funktioiden arvojen laskeminen on mekaanista, joten alkuarvon ja rekursiosäännön keksimisen jälkeen loput todistuksesta voidaan jättää tietokoneen laskettavaksi. Iso ongelma tämän menetelmän kanssa on kuitenkin laskentaan kuluvan ajan arviointi. Kuten Turingin kone -määritelmästä voidaan huomata, on mahdollista, että kone törmää sellaiseen muuttujan arvoon, joka ei ole määritelty, tai haettu arvo on niin pitkällä rekursioketjussa, että aikavaativuustaso käy liian suureksi. Tietojenkäsittelytieteen puolella rekursio on kuitenkin tärkeä ohjelmointikeino, jonka avulla voidaan esimerkiksi kiertää luuppirakenteen puuttuminen Turing-täydellisissä ohjelmointikielissä.

Lähteet:

Cutland, Nigel: Computability An introoduction to recursive function theory. Cambridge University Press, Iso-Britannia, 2000.

Jones, Neil: Computability Theory An Introduction. Academic Press, USA, 1973.

Väänänen, Jouko: Matemaattinen logiikka. Gaudeamus, Helsinki, 1988.

http://en.wikipedia.org/wiki/Tower_of_Hanoi
http://en.wikipedia.org/wiki/M-recursive_function
http://en.wikipedia.org/wiki/Recursion_(computer_science)
http://en.wikipedia.org/wiki/Recursive_function_theory
http://en.wikipedia.org/wiki/Primitive_recursive_function
http://www.tutorialspoint.com/cprogramming/c_recursion.htm