

# 58110-3: Tieteellisen kirjoittamisen kurssi

## Algoritmiaiheita

28. elokuuta 2006

”A scholar is just a library’s way of making another library.” *Daniel C. Dennett*

## 1 Ohjeita

Tässä listassa on ehdotuksia tieteellisen kirjoittamisen kurssin algoritmiryhmän aiheiksi. Valitse listalta joitain kandidaattiaiheita. Listan aiheet ovat suuntaa-antavia; lopullinen rajaus ja otsikko sovitaan kurssin kuluessa. Voit myös ehdottaa omaa aiheitasi. Oma aihe on kaikkein paras vaihtoehto, kunhan se sopii algoritmiryhmän aihepiiriin, ja siitä löytyy riittävästi sopivaa materiaalia.

Kukin aihe tai aihepiiri on varustettu listalla avainsanoja, jotka kuvaavat aiheeseen liittyviä matematiikan ja tietojenkäsittelyn osa-aloja tai hyödyllisiä kursseja. Nämä eivät ole esitietovaatimuksia, mutta on selvää että taustatiedot ja menetelmiin liittyvä kiinnostus saattavat nopeuttaa artikkelien ymmärtämistä ja kirjoitustyötä. Peruslähtökohta on kuitenkin se, että opinto-oppaassa mainitut esitietovaatimukset antavat riittävät valmiudet kaikkiin aiheisiin.

Listalla on kustakin aiheesta mainittu muutama lähdeartikkeli tai kirja, joiden kautta voi päästä helpommin aihepiiriin kiinni. Valintariskin minimoimiseksi kannattaa silmäillä läpi esimerkiksi yksi lähdeartikkeli ennenkuin kiinnität valintasi.

Aiheisiin tai aihepiireihin on merkitty myös ohjaajan arvio kirjaimilla A–C. Kirjain A tarkoittaa aihepiiriä, joka on lähempänä ohjaajan omaa tutkimusta, ja kirjain C taas aihepiiriä, jossa ohjaajakin varmasti oppii uutta. Kaikista aiheista saa toki ohjausta, myös omasta aiheestasi.

## 2 Aiheita

### 2.1 Laskennan vaativuusteoriaa

*Ohjaajan arvio:* B. *Kursseja:* Laskennan teoria, logiikka 1, matemaattinen logiikka.

**Kvantifioidut Boolean kaavat** [CGS98, GW99, Rin99]

Kvantifioidut Boolean kaavat (*quantified Boolean formulae, QBF*) laajentavat propositiologiikkaa sallimalla eksistentiaali- ja universaalikvanttorit. Lähdeartikkeleissa tarkastellaan esimerkiksi QBF-kaavojen SAT-ongelmaa eli kaavojen toteutuvuutta, joka on NP-täydellisyyden perustapaus.

**Formaalikielten samuusongelmat** [Sén00]

Käyttäytyvätkö annetut kaksi automaattia samalla tavalla, eli hyväksyvätkö ne saman kielen? Vastikään on saatu uusia tuloksia pinoautomaateista (*push-down automata*) käyttäen loogisia menetelmiä.

*Liittyy:* Matemaattinen logiikka, Gödel-numerointi.

## Resoluutiodistusten kompleksisuus [ABM01, PR01]

Toteutuvuusongelma on tutkituimpia NP-täydellisiä ongelmia. Viime aikoina on saatu uusia tuloksia toteutuvuustodistusten kompleksisuudesta.

## Kiintoparametrivaativuus [CKJ01]

Kiintoparametrivaativuus (*fixed-parameter complexity*). Tietyille NP-täydellisille ongelmille on olemassa algoritmeja joiden aikavaativuus on luokkaa  $O(f(k) * poly(n))$ , missä  $n$  on syöteen koko,  $k$  ongelman parametri (esimerkiksi solmupeitteen koko),  $f(k)$  on funktio, joka ei riipu  $n$ :stä ja  $poly(n)$  on  $n$ :n polynomi. Tällaiset algoritmit pystyvät käsittelemään suuriakin syötteitä, jos parametri  $k$  on kiinnitetty riittävän pieneksi.

## 2.2 Tietorakenteita

*Ohjaajan arvio:* B. *Kursseja:* Tietorakenteet, algoritmien suunnittelu ja analyysi.

### Keko [Cha99b]

Keko (*heap*) on klassinen tietorakenne [Tar83, luku 3]. Verkon virittävän puun laskentaa varten on kehitetty uusi ”pehmeä” kekorakenne.

*Liittyy:* Mahdollisesti informaatioteoriaa.

### Tietorakenteiden vyörytys [Oka98]

(*Data-structural bootstrapping.*)

### Joukkojen erilliset yhdisteet [TvL, TW89]

Joukkojen yhdisteiden säilyttäminen ja purkaminen.

### Binääriset etsintäpuut [ST85]

Splay-puu on itsesäätyvä versio perinteisestä binäärisestä hakupuusta.

*Liittyy:* Tasoitettua analyysiä.

### Tietorakenteiden tiivistys [Jac89, Cla96]

Tiedon tiivistyksessä on tavoitteena esittää tieto (esim. tekstidokumentti) mahdollisimman pienessä tilassa siten, että alkuperäinen tieto voidaan palauttaa virheettää. Tietorakennetta tiivistettäessä lisätavoitteena on säilyttää tietorakenteen toimintakyky. Esimerkki: Binääripuu halutaan esittää muodossa, jossa kullekin solmulle löydetään vakioajassa sen vasen ja oikea lapsi. Perinteinen linkein esitetty puu vie  $O(n \log n)$  bittiä tilaa, missä  $n$  on puun solmujen määrä (jokaiseen solmuun on talletettu 2 linkkikenttää, kukin luokkaa  $\log n$  bittiä). On kuitenkin mahdollista esittää puu  $O(n)$  bitillä siten, että vasen ja oikea lapsi löytyvät vakioajassa.

*Liittyy:* Informaatioteoriaa.

## Staatin tekstin indeksointi loppuosatietorakenteilla [MM93]

Suuri osa internetin sisällöstä on tekstiä, joka muuttuu suhteellisen hitaasti. Tällaiselle staatilliselle tekstile on kehitetty indeksirakenteita, jotka mahdollistavat erittäin nopeat tekstihaut. Loppuosatietorakenteet kuten suffiksipuut ja -taulukot ovat perusesimerkkejä tällaisista rakenteista.

*Kursseja:* Merkkijonomenetelmät.

## 2.3 Kombinatorista optimointia

*Ohjaajan arvio:* A. *Liittyy:* Kombinatoriikka, verkot, puut. *Kursseja:* Diskreetti matematiikka 2, kombinatorinen optimointi, laskennan teoria.

### Kauppamatkustajan ongelma [BDvD<sup>+</sup>98]

Kauppamatkustajan ongelma (*travelling salesman problem, TSP*) on yksi yleisimmistä NP-täydellisistä ongelmista, ja niinpä sen monille erikoistapauksille on kehitetty ratkaisualgoritmeja, esimerkiksi TSP euklidisessa tasossa.

### Verkon virittävän puun laskenta [Tar83, luku 8], [Cha99a]

Virittävän puun (*spanning tree*) laskenta on klassinen algoritmiongelma. Nyt siihen on löytynyt uusi algoritmien lähestymistapa.

### Avioliitto-ongelma [GS62, GS85, BR97, IMMM99]

Avioliitto-ongelmassa (*stable marriage problem*) pitää parittaa annetut naiset ja miehet toisilleen siten, etteivät he tee syrjähyppyjä, kun kaikkien mieltymykset tunnetaan. Tämän jo klassisen algoritmisen ongelman perustapaus ratkeaa polynomisessa ajassa, mutta laajennuksissa on törmättykin vaikeuksiin.

### Verkon murtolukuväritys [HS88, LY94, Jan03]

Murtolukuväritysongelmat (*fractional graph coloring*) ovat muunnoksia tunnetuista verkko-väritysongelmista. Murtolukuväritystä voi soveltaa esimerkiksi tiedonsiirron ajoittamiseen langattomassa verkossa. Kaarivärityksen murtolukuversio ratkeaa polynomisessa ajassa, mutta solmuvärityksen approksimointikin on vaikeaa millä tahansa vakiosuhteella. Monille verkkojen luokille voidaan kuitenkin löytää approksimaatioalgoritmeja.

*Liittyy:* Approksimaatioalgoritmit.

### Steiner-puut [Vaz03]

Mitä nykyään tiedetään Steiner-puihin (*Steiner tree, directed Steiner tree*) liittyvien optimointiongelmien approksimoituvuudesta ja ei-approksimoituvuudesta?

*Liittyy:* Approksimaatioalgoritmit.

## 2.4 Laskennallista geometriaa

*Ohjaajan arvio:* C. *Kursseja:* Geometriset menetelmät.

### Leikkaavien janojen ongelma [Bal95]

### Monikulmioiden kolmiointialgoritmit [AGR00]

## Kineettiset tietorakenteet [BGH99, KSS00]

Tehokkaita tietorakenteita liikkuvien kohteiden ominaisuuksien käsittelyyn. Esimerkki: liikkuvan pistejoukon konveksin peitteen ylläpito. Kinetisointi: muutetaan algoritmi liikkuvalle datalle sopivaksi.

## 2.5 Logiikkaa sekä ohjelmointikielten syntaksia ja semantiikkaa

*Ohjaajan arvio:* C. *Kursseja:* Johdatus funktionaaliseen ohjelmointiin, logiikka 1, spesifioinnin ja verifoinnin perusteet.

### Rajoitetietokantojen kyselykielet [AHV95, sivut 94–98], [BL97, BL99, Lib99]

Rajoitetietokantojen kyselykielet (*constraint query languages*) käsittelevät sellaisia tietokantoja, joihin on talletettu informaatiota jostakin äärettömästä rakenteesta äärellisenä kokoelmana sitä rajoittavia lauseita. Esimerkiksi kolmiulotteinen avaruus on periaatteessa ääretön, mutta sen osia voidaan kuvailla sellaisilla lauseilla kuin ”kaikki ne pisteet joiden  $x$ -koordinaatti on positiivinen”. Mutta miten näin esitetystä informaatiosta vastataan käyttäjän kysymyksiin?

*Liittyy:* Tietokannat, logiikka.

### Rajoitelogiikkaohjelmointi [Col90]

Rajoitelogiikkakielet.

*Liittyy:* Prolog-ohjelmointi.

### Funktionaalisen ohjelmointikielen suoritusmalli [Pip97, BJd97, ACM00, BJ00, Lei99]

Funktionaalisen ohjelmointikielen suoritusmalli on  $\lambda$ -lausekkeiden  $\beta$ -sieventäminen. Mitä se maksaa algoritmisena ilmaisuvoimana? Entä laskentaresursseina?

*Liittyy:* Jonkin funktionaalisen kielen osaaminen on varmasti eduksi. Esimerkkejä funktionaalisisista kielistä: Haskell (puhdas), Lisp.

### Spesifikaatiot ovat väitteitä, ohjelmat niiden todistuksia [Bac90, CNSv94, Wad00]

Formaali looginen ohjelmankehitys.

*Liittyy:* Logiikkaa, spesifointia ja verifointia.

### Ohjelmien aksiomaattinen semantiikka [Hoa]

Heikoimman ennakkoehdon semantiikka, todistetusti oikeellisten ohjelmien kehittäminen.

*Liittyy:* Logiikkaa, spesifointia ja verifointia.

### (LA)LR-jäsennys [ASU86, SSS88, SSS90]

Kielen jäsenys esimerkiksi kääntäjien toteutuksessa.

## 2.6 Koneoppimista, tekoälyä ja data-analyysia

*Ohjaajan arvio:* B. *Liittyy:* Todennäköisyyslaskentaa. *Kursseja:* Koneoppiminen, todennäköisyyslaskennan ja tilastotieteen kurssit, tekoäly, kolme käsitettä -kurssit.

### Todennäköisesti oikea käsitteenoppiminen [KLV94]

Käsitteenoppimisen teoriaa. Perustapauksessa opitaan Boolean kaavoja sillä oletuksella, että opittava asia on täysin ilmaistavissa Boolean kaavalla. Perus-PAC-oppimisella ei ole välttämättä kovin paljoa tekemistä reaali maailmassa toimivien koneoppimismenetelmien kanssa, joten aihetta tulisi pitää itsessään kiinnostavana tai yleissivistävänä.

## Päätöspuiden oppiminen [Qui86]

Heuristiikkoja päätöspuiden muodostamiseen, päätöspuiden oppiminen.

*Liittyy:* Informaatioteoriaa.

## Geneettiset algoritmit koneoppimisessa [GRS90]

Geneettiset algoritmit päätössääntöjen oppimisessa.

Huom: Geneettiset algoritmit ja geneettinen ohjelmointi sopivat usein paremmin muuhun kombinatoriseen optimointiin kuin induktiiviseen käsitteenoppimiseen (ylisovitusvaara). Aihetta voisi siis muokata enemmän optimoinnin suuntaan.

## Robottien navigointi [Thr98]

Oppiminen robottien navigoinnissa, robottien navigointialgoritmit.

*Liittyy:* Matriisilaskentaa, Markovin prosesseja, Kalman-suodattimia, *particle filtering*-tyyppiset tilasiirtymäjärjestelmiin liittyvät tilastolliset estimointimenetelmät.

## Itseorganisoivat kartat ja niiden sovellukset [Koh96]

Itseorganisoivan kartan tarkoituksena on projisoida suuriulotteisessa vektoriavaruudessa sijaitseva data pieniulotteiseen vektoriavaruuteen (esim. 2-d) esimerkiksi datan visualisointia varten tai siksi että tyypilliset etäisyysmitat (euklidinen) käyttäytyvät huonosti suuriulotteisissa avaruuksissa.

Itseorganisoituvat kartat voidaan nähdä myös pääkomponenttianalyysin epälineaarisenä vastineena (ei suoria ominaisvektoreita, eikä ortogonaalista kantaa).

*Liittyy:* Matriisilaskentaa.

## Riippumattomien komponenttien analyysi [Kar97]

Sokea lähteiden erottelu. Havaittu signaalivektori  $x$  oletetaan  $k$ :n tilastollisesti riippumattoman epägaussisen signaalin sekoitteenä ( $x = As$ ). Tehtävänä on selvittää alkuperäinen signaalivektori oletusten ja havaitun signaalivektorin perusteella niin pitkälle kuin mahdollista ( $s = Wx$ ).

Algoritmit käyttävät pääkomponenttianalyysiä esikäsittelyyn (valkaisu, dimension pudotus), joten pääkomponenttianalyysin ymmärtäminen lienee myös tarpeen.

Sovelluksia mm. laskennallisessa neurotieteessä.

*Liittyy:* Matriisilaskentaa, informaatioteoriaa.

## Oppivat peliohjelmat [Tes92]

TD-oppimisalgoritmi.

*Liittyy:* Markovin prosesseja.

## Tukivektorikoneet [OFG97]

Tukivektorikoneet (*support vector machines, SVM*) ovat 1990-luvun loppupuolen kuumin koneoppimisparadigma. SVM:iä ja muita ydinfunktioihin (*kernel function*) perustuvia data-analyysimenetelmiä on sovellettu menestyksellisesti moniin tehtäviin, joihin aiemmin käytettiin etupäässä neuraaliverkkoja.

*Liittyy:* Optimointia.

## Oppimisen tehostaminen [Sch99]

Oppimisen tehostaminen (*boosting*) on koneoppimisparadigma, jossa heikosta oppimisalgoritmista – sellaisesta, joka toimii vain hieman paremmin kuin satunnainen arvaus – voidaan tehdä vahva oppija yhdistelemällä sopivasti heikon oppijan tuottamia malleja.

## Mekanismisuunnittelu [BR97, luku 7], [MT99, Nis99, Pap96]

Mekanismisuunnittelu (*mechanism design*) on “pelien suunnittelua” peliteoreettisessa mielessä. Toisin kuin tyypillisessä peliteorian ongelmassa, tässä ei aseteta itsekään “pelaajan” asemaan ratkaisemaan optimaalista tapaa pelata peliä, vaan asetutaan “keskusjohdon” asemaan suunnittelemaan pelin säännöt siten että itsekkäiden pelaajien omaa hyötyä maksimoivasta käytöksestä seuraa keskusjohdon haluama lopputulos. Esim. hyvin suunnitelluista pelisäännöistä voi seurata ettei itsekään pelaajan kannata petkuttaa tai vahingoittaa muita pelaajia.

Tunnettuna esimerkkinä mekanismisuunnittelusta voidaan mainita *Vickreyn huutokauppasääntö*, jota käytettiin esim. Googlen listausannissa.

Tietojenkäsittelytieteellä on liittymäkohtansa taloustieteisiin ja peliteoriaan. Mekanismisuunnittelua voi soveltaa myös Internetiin [Pap01]. Toistuvien pelien tapauksessa peliteoreettisilla aiheilla on liittymäkohtia koneoppimiseen: TD-oppiminen, palauteoppiminen, ja Markovin prosessit.

## Piilo-Markov-mallit [Ben99]

Piilo-Markov-malleja (*hidden Markov models, HMM*) käytetään moniin tietojenkäsittelyongelmiin puheentunnistuksesta geenisekvenssien analysointiin.

*Liittyy:* Markovin prosesseja. *Kursseja:* Graphical models.

## Nopeasti sekoittuvat Markovin ketjut [MR95, luvut 6.7 ja 11], [BDGJ99]

Nopeasti sekoittuvat Markovin ketjut (*rapidly mixing Markov chains*) tarjoavat keinon vaikeiden laskentaongelmien likimääräiseen ratkaisemiseen suurella todennäköisyydellä. Tyypillisesti algoritmit ovat yksinkertaisia, mutta niiden tehokkuuden todistaminen on mutkikasta.

*Liittyy:* Markovin prosesseja. *Kursseja:* Satunnaisalgoritmit.

## Robusti logiikka [Val00]

Logiikka on menettänyt suosiotaan tekoälyn piirissä, koska logiikkapohjaisista järjestelmistä tulee usein ”helposti särkyviä”, jos niiden on käsiteltävä reaali maailmasta saatavaa epävarmaa tietoa. Voidaanko asian tila korjata ?

*Liittyy:* Logiikkaa.

## 2.7 Muita aiheita

### Hahmonsovitus merkkijonoissa [KMP77, CR02]

Merkkijonomenetelmien avulla pyritään löytämään tekstistä hahmon osumia eli paikkoja, joissa hahmo esiintyy osin tai kokonaan.

*Ohjiaan arvio:* C. *Kursseja:* Merkkijonomenetelmät.

### Satunnaisalgoritmit [MR95]

Useisiin NP-koviin optimointiongelmiin on kehitetty satunnaisalgoritmeja, jotka suurella todennäköisyydellä tuottavat likimääräisesti oikean vastauksen. Monet satunnaisalgoritmit ovat hyvin yksinkertaisia, mutta niiden analyysi on epätriviaalia. Eri ongelmien analyysissä käytetään kuitenkin muutamia perustekniikoita. Millä eri tavoilla satunnaisalgoritmeja voidaan analysoida?

*Ohjaajan arvio:* B. *Liittyy:* Todennäköisyyslaskentaa. *Kursseja:* Satunnaisalgoritmit, algoritmien suunnittelu ja analyysi.

### **Peitto- ja pakkaus-LP** [GK98]

Peitto- ja pakkaustyyppisten LP-ongelmien sovelluksia ja approksimatiivista ratkaisemisista.

*Ohjaajan arvio:* B. *Liittyy:* Approksimaatioalgoritmit, lineaarinen ohjelmointi.

### **Piin laskukaava** [BBP97]

Helsingin Sanomien tiedesivuilla esiteltiin lauantaina 5. elokuuta 2000 Baileyn, Borweinin ja Plouffen ”ihmekaavaa” vakiolle  $\pi$ . Kaava antaa tehokkaan algoritmin vakion heksadesimaaliesitykselle – vaan miten ja millaisen?

*Ohjaajan arvio:* C. *Liittyy:* Matemaattinen yleissivistys eduksi.

### **Kvanttilaskenta** [Gru99, Sho99]

Kvanttilaskennassa (*quantum computing*) kehitetään algoritmeja sellaisille laskulaitteille, joissa informaation kulkua määräävätkin kvanttimekaniikan lait vanhan tutun binäärielektronikan sijaan. Vaan millaista on suunnitella niille algoritmejä?

*Ohjaajan arvio:* C. *Liittyy:* Todennäköisyyslaskentaa, matriisilaskuja.

### **Julkisen avaimen salakirjoitus** [RSA78]

RSA-algoritmi yms.

*Ohjaajan arvio:* C. *Liittyy:* Kryptografiaa, algebraa, lukuteoriaa.

### **Onpa maailma pieni -ilmiö** [Kle00]

Analysoitaessa erilaisia verkostoja on havaittu, että useimmiten mitä tahansa kahta verkon solmua yhdistää lyhyt polku (*small world phenomenon*). Esimerkiksi lentoreiteissä, internetissä, geenisäätelystä ja ihmisten tuttavuuspiireissä tämä ilmiö on havaittavissa. Mutta voidaanko tällaisten verkostojen syntyä selittää algoritmisesti?

*Ohjaajan arvio:* C.

### **Tulon summaus -ongelma** [SH96, AM00, Ste03]

Tulon summaus -ongelmassa (*sum-product problem*) tehtävänä on laskea moniulotteinen summa funktiosta, joka voidaan esittää alempiulotteisten funktioiden tulona.

*Ohjaajan arvio:* A. *Liittyy:* Algebra. *Kursseja:* Graphical models, Research seminar on algorithms: sums of products.

### **Nopea Fourier- ja Möbius-muunnos.** [Ken91], [CLRS01, luku 30]

*Ohjaajan arvio:* A. *Liittyy:* Fourier-muunnosta käsitellään signaalinkäsittelyn kursseilla.

Kolmogorov-kompleksisuus on eräänlainen satunnaisuuden mittari. Bittijonon monimutkaisuus on yhtä kuin lyhimmän sellaisen tietokoneohjelman pituus, jonka tulosteena on kyseinen bittijono. Toisin sanoen, jos ohjelma on olennaisesti ”yhtä pitkä” kuin bittijono itse, bittijonoa voidaan pitää ”satunnaisena” siinä mielessä ettei sen rakenteessa ollut mitään säännönmukaisuutta, jonka perusteella ohjelma voisi tehdä muuta kuin muistaa bittijonon suoraan sellaisena kuin se tulostetaan.

*Ohjaajan arvio:* A. Liittyi: Todennäköisyyslaskentaa, informaatioteoriaa, Turingin koneita.

## Viitteet

- [ABM01] D. Achlioptas, P. Beame, M. Molloy. A sharp threshold in proof complexity. *ACM Symposium on Theory of Computing*, 337–346, 2001.
- [ACM00] A. Asperti, P. Coppola ja S. Martini. (Optimal) duplication is not elementary recursive. *Annual ACM Symposium on the Principles of Programming Languages*, sivut 96–107, 2000.
- [AGR00] N.M. Amato, M.T. Goodrich ja E.A. Ramos. Linear-time triangulation of a simple polygon made easier via randomization. *16th ACM Symposium on Computational Geometry*, sivut 201–212, 2000.
- [AHV95] S. Abiteboul, R. Hull ja V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [AM00] S.M. Aji ja R.J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.
- [ASU86] A.V. Aho, R. Sethi ja J.D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986.
- [Bac90] R.C. Backhouse. Constructive type theory: A perspective from computing science. E.W. Dijkstra, toim., *Formal Development of Programs and Proofs*, chapter 1, sivut 1–32. Addison-Wesley, 1990.
- [Bal95] I.J. Balaban. An optimal algorithm for finding segment intersections. *11th ACM Symposium on Computational Geometry*, sivut 211–219, 1995.
- [BBP97] D. Bailey, P. Borwein ja S. Plouffe. On the rapid computation of various polylogarithmic constants. *Mathematics of Computation*, 66(218):903–913, 1997.
- [Ben99] Yoshua Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999.
- [BDGJ99] R. Bublely, M.E. Dyer, C.S. Greenhill, M. Jerrum. On Approximately Counting Colorings of Small Degree Graphs. *SIAM Journal of Computing*, 29(2):387–400, 1999.
- [BDvD<sup>+</sup>98] R.E. Burkard, V.G. Deineko, R. van Dal, J.A.A. van der Veen ja G.J. Woeginger. Well-solvable special cases of the traveling salesman problem: a survey. *SIAM Review*, 40(3):496–546, 1998.



- [BGH99] J. Basch, L.J. Guibas ja J. Hershberger. Data structures for mobile data. *Journal of Algorithms*, 31:1–28, 1999.
- [BJ00] A.M. Ben-Amram ja N.D. Jones. Computational complexity via programming languages: Constant factors do matter. *Acta Informatica*, 37:83–120, 2000.
- [BJd97] R. Bird, G. Jones ja O. de Moor. More haste, less speed: Lazy versus eager evaluation. *Journal of Functional Programming*, 7(5):541–547, 1997.
- [BL97] M. Benedikt ja L. Libkin. Languages for relational databases over interpreted structures. *ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems*, sivut 87–96, 1997.
- [BL99] M. Benedikt ja L. Libkin. Exact and approximate aggregation in constraint query languages. *ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems*, sivut 102–113, 1999.
- [BR97] M. Balinski ja G. Ratier. On stable marriages and graphs, and strategy and polytopes. *SIAM Review*, 39(4):575–604, 1997.
- [CGS98] M. Cadoli, A. Giovanardi, ja M. Schaerf. An algorithm to evaluate quantified boolean formulae. *National Conference of the AAAI*, sivut 262–267, 1998.
- [Cha99a] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. Tech Report 99-099, NECI Research, 1999. To appear in the *Journal of the ACM*.
- [Cha99b] B. Chazelle. The soft heap: An approximate priority queue with optimal error rate. Tech Report 99-097, NECI Research, 1999. To appear in the *Journal of the ACM*.
- [CKJ01] J. Chen, I. A. Kanj ja W. Jia. Vertex cover: further observations and further improvements. *Journal of Algorithms*, 41 (2001), sivut 280–301.
- [Cla96] D. Clark. *Compact Pat Trees*. Väitöskirja, University of Waterloo, Canada, 1996.
- [Col90] Colmerauer. An introduction to Prolog III. *Communications of the ACM*, 33(7):69–90, 1990.
- [CNSv94] T. Coquand, B. Nordström, J.M. Smith ja B. von Sydow. Type theory and programming. *Bulletin of the EATCS*, 52:195–228, 1994.
- [CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest ja C. Stein. *Introduction to Algorithms*. Second Edition. MIT Press and McGraw-Hill, 2001.
- [GK98] Naveen Garg ja Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *Proc. 39th Ann. Symp. on Foundations of Computer Science*, sivut 300–309, 1998.
- [GRS90] Grefenstette, Ramsey ja Schultz. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4):355–382, 1990.
- [CR02] M. Crochemore ja W. Rytter. *Jewels of Stringology*. World Scientific Publishing, 2002.
- [Gru99] J. Gruska. *Quantum Computing*. McGraw-Hill, 1999.

- [GS62] D. Gale ja L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.
- [GS85] D. Gale ja M. Sotomayor. Some remarks on the stable marriage problem. *Discrete Applied Mathematics*, 11:223–232, 1985.
- [GV99] A. Gammerman, V. Vovk. Kolmogorov Complexity: Sources, Theory and Applications. *The Computer Journal*, 42(4):252–255, 1999.
- [GW99] I.P. Gent ja T. Walsh. Beyond NP: the QSAT phase transition. *National Conference of the AAAI*, sivut 648–653, 1999.
- [Hoa] C. A. R. Hoare. An axiomatic basis for computer programming *Communications of the ACM*.
- [HS88] Bruce Hajek ja Galen Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, 1988.
- [IMMM99] K. Iwama, D. Manlove, S. Miyazaki ja Y. Morita. Stable marriage with incomplete lists and tries. *International Colloquium on Automata, Languages and Programming*, sivut 443–452, 1999.
- [Jac89] G. Jacobson. *Succinct Static Data Structures*. Väitöskirja, CMU-CS-89-112, Carnegie Mellon University, 1989.
- [Jan03] Klaus Jansen. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. *Theoretical Computer Science*, 302(1–3):239–256, 2003.
- [JLV99] T. Jiang, M. Li, P. Vitanyi. New applications of the incompressibility method: Part I. *The Computer Journal*, 42(4):287–293, 1999.
- [JLV00] T. Jiang, M. Li, P. Vitanyi. Average-case analysis of algorithms using Kolmogorov complexity. *Journal of Computer Science and Technology*, 15(5):402–408, 2000.
- [Kar97] Karhunen, Oja, Wang, Vigarío, Joutsensalo. A class of neural networks for independent component analysis. *IEEE Transactions on Neural Networks*, 8(3):486–504, 1997.
- [KLV94] Kearns, Li, Valiant. Learning boolean formulas. *Journal of the ACM*, 41(6):1298–1328, 1994.
- [Ken91] R. Kennes. Computational aspects of the moebius transform of a graph. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:201–223, 1991.
- [KSS00] D. Kirkpatrick, J. Snoeyink ja B. Speckmann. Kinetic collision detection for simple polygons. *16th ACM Symposium on Computational Geometry*, sivut 322–330, 2000.
- [Kle00] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. *Proc. 32nd ACM Symposium on Theory of Computing*, 2000, sivut 163–170.
- [KMP77] Knuth, Morris ja Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6(2):323–350, 1977.
- [Koh96] Kohonen, Oja, Simula, Visa ja Kangas. Engineering applications of the self-organizing map. *Proc. IEEE*, 84(10):1358–1384, 1996.

- [Lei99] D. Leivant. Applicative control and computational complexity. *Computer Science Logic*, sivut 82–95, 1999.
- [Lib99] L. Libkin. Query languages with arithmetic and constraint databases. *ACM SIGACT News*, 30(4):41–50, 1999.
- [LY94] Carsten Lund ja Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
- [MM93] U. Manber ja G. Myers. Suffix arrays: A new method for on-line string searches. *SIAM J. Comput.*, 22(5):935–948, Oct. 1993.
- [MR95] R. Motwani ja P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [MT99] D. Monderer ja M. Tennenholtz. Distributed games: From mechanisms to protocols. *National Conference of the AAAI*, sivut 32–37, 1999.
- [Nis99] N. Nisan. Algorithms for selfish agents. *Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, sivut 1–15, 1999.
- [Oka98] C. Okasaki. *Purely Functional Data Structures*. Cambridge University Press, 1998.
- [OFG97] E. Osuna, R. Freund ja F. Girosi. Training support vector machines: an application to face detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, sivut 130–136.
- [Pap96] C.H. Papadimitriou. Computational aspects of organization theory (extended abstract). *European Symposium on Algorithms*, sivut 559–565, 1996.
- [Pap01] C.H. Papadimitriou. Algorithms, Games, and the Internet. *ICALP*, 2001.
- [Pip97] N. Pippenger. Pure versus impure lisp. *ACM Transactions on Programming Languages and Systems*, 19(2):223–238, 1997.
- [PR01] T. Pitassi ja R. Raz. Regular resolution lower bounds for the weak pigeonhole principle. *ACM Symposium on Theory of Computing*, 347–355, 2001.
- [Qui86] Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [Rin99] J. Rintanen. Improvements to the evaluation of quantified boolean formulae. *International Joint Conference on Artificial Intelligence*, sivut 1192–1197, 1999.
- [RSA78] Rivest, Shamir ja Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of ACM*, 21(2):120–126, 1978.
- [Sch99] R. Schapire. A brief introduction to boosting. *Proceedings of International Joint Conference on Artificial Intelligence*, sivut 1401–1405, 1999.
- [Sén00] G. Sénizergues. Complete formal systems for equivalence problems. *Theoretical Computer Science*, 231:309–334, 2000.
- [SH96] R.E. Stearns ja H.B. Hunt III. An algebraic model for combinatorial problems. *SIAM Journal on Computing*, 25(2):448–476, 1996.

- [Sho99] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.
- [SSS88] S. Sippu ja E. Soisalon-Soininen. *Parsing Theory*, volume I: Languages and Parsing. Springer-Verlag, 1988.
- [SSS90] S. Sippu ja E. Soisalon-Soininen. *Parsing Theory*, volume II: LR(k) and LL(k) Parsing. Springer-Verlag, 1990.
- [ST85] Sleator ja Tarjan. Self-adjusting binary search trees. *Journal of ACM*, 32(3):652–686, 1985.
- [Ste03] R.E. Stearns. Deterministic versus nondeterministic time and lower bound problems. *Journal of the ACM*, 50(1):91–95, 2003.
- [Tar96] J. Tarhio. A sublinear algorithm for two-dimensional string matching. *Pattern Recognition Letters*, 17 (1996), 833–838.
- [Tar83] R.E. Tarjan. *Data Structures and Network Algorithms*. SIAM, 1983.
- [TvL] Tarjan ja van Leeuwen. Worst-case analysis of set union algorithms. *Journal of ACM*, 31(2):245–281.
- [TW89] Tarjan ja Westbrook. *SIAM Journal of Computing*, 18(1):1–11, 1989.
- [Tes92] G. Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8(3/4):257–278, 1992.
- [Thr98] Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [Val00] L.G. Valiant. Robust Logics. *Artificial Intelligence*, 117(2):231–253, 2000.
- [Vaz03] V.V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2003.
- [Wad00] P. Wadler. 19th century logic and 21st century computing. *Dr. Dobb's Journal*, 2000. <<http://www.ddj.com/articles/2000/0013/0013g/wadler.pdf>>.