Sertifitseeritud tarkvarast sertifitseeritud teaduseni

Tarmo Uustalu
Küberneetika Instituut

Avalik loeng Eesti Teaduste Akadeemias, 6.4.2011

Kui ei usalda, kontrolli!

- Kui keegi midagi väidab (nt “Kaup on hea”), kuidas selles veenduda?
- Usaldada?
- Aga kui saab, siis ehk kontrollida?
- Usaldus põhineb skeptiku positiivsel hoiakul väitja suhtes või skeptiku jaoks autoriteetse kolmanda poole soovitusel väitja kohta
- Kontroll lubab skeptikul jääda sõltumatuks!
- Seega kindlam
- Aga kontrollimine on praktiline ainult siis, kui pole liiga raske

Sertifitseerimine

- Sertifitseerima — *certus* + *ficare* — täht-tähelt: kindlust looma
- Paljude jaoks tähendab sertifikaat soovitust, st kindlus taandub usaldusele
- Sertifikaat selle loengu mõistes on palju tugevam garantii, põhineb praktilisel kontrollitavusel, ei nõua usaldust väitja suhtes:

sertifikaat

=

vihjed, mille abil skeptik saab mingit liiki väite tõesust iseseisvalt efektiivselt kontrollida

Sertifitseerimine

- Sisuliselt on sertifikaat tõestus, mida saab mehaaniliselt kontrollida, või kogu oluline info tõestuse mehaaniliseks rekonstrueerimiseks ja selle kontrolliks.
- Tüüpstsenaariumis väitja (veenmisest huvitatud pool) tõestab väite ning pakib selle sertifikaadiks.
- Tõestuse koostamine on üldiselt loomingulisust nõudev intellektuaalne töö
- Skeptik (tõest huvitatud pool) rekonstrueerib kandidaattõestuse ja kontrollib, et see tõepoolest on tõestus

Tänane loeng

- Põgus ekskurss sertifitseerimise ajalukku ning tulevikuväljavaadetesse
- Tarkvara (laiemalt arvutisüsteemid):

  kindluse loomine tarkvara (arvutisüsteemide) korrektsuse ja ohutuse kohta,

  Tarkvara pole mitte ainult keeruline, vaid mahukas! Sertifikaate peab saama arvutiprogrammi toel koostada, arvutiprogrammiga automaatselt kontrollida
- (Reaal)teadus, nt matemaatika:

  kindluse loomine teadustulemuste kohta,

  Uus: Ka matemaatika võib olla mitte ainult keeruline, vaid ka mahukas, ka siin on vajalik arvuti abi!

Sertifitseeritud tarkvara

- Programmeerimine, teadaolevalt, on vigadealdis töö, palju kasutusel olevat tarkvara on vigane
- Tarkvara peaks olema
  - korrektne, st peab tegema seda, milleks ette nähtud
  - ohutu, st ei tohi teha halba
- Vigade rohkus tuleb programmide loogilisest keerukusest ning suurest mahust (palju juhte)
- Tarkvara vigadel on kõrge hind
- Testimisest ei piisa, kuna vaatab lõpmatust hulgast käivitustest läbi vaid väikese lõpliku alamhulga

Tarkvara verifitseerimine

- Tarkvara korrektsuse ja ohutuse tõestamine (verifitseerimine) on olnud arvutiteaduse suureks unistuseks 1960ndatest
- Enam kui kriitiline täna (Sir Tony Hoare'i verifitseeritud tarkvara väljakutse)
- Kuid täna on unistus muutunud ka reaalsuseks, sest on arendatud sobivad tehnoloogiad

Tarkvara verifitseerimine

- Moodsaid ideid:
- Tõestada mitte täiesti üldkujulisi programmide omadusi, vaid lihtsamaid (nt “poliitikate” järgimist)
- Kui tõestada üldisi omadusi, siis teha seda võimalikult sobivates loogikaformalismides, mis soosivad ülevaatlikkust, modulaarsust
- Tõestuse koostamist ei saa nõuda igaühelt; ideaalselt peaks seda tegema programmeerija, sest tema teab, kuidas ta mõtles
- Tõestused talletada sertifikaatides, mida programmi iga tarbija saab oma kontrollijaga kontrollida (nt annotatsioonid programmis tüübikontrollijale vm)

Arvutitoe, automatiseerituse vajadus

- Programmid ei ole mitte ainult keerulised
- Nad on ka suured ja nende abstraktsed olekuruumid hoopis suured (“olekuruumi plahvatus”)
- Käsitsi tõestamine või isegi käsitsi tõestuste kontroll kõige pisemategi huvitavate programmide juures ei ole seetõttu realistlikud
- Sertifikaatide koostamine peab olema programmiga automatiseeritud (automaatne teoreemitõestamine vms) või toetatav (interaktiivne teoreemitõestamine)
- Sertifikaatide kontroll peab olema automaatne

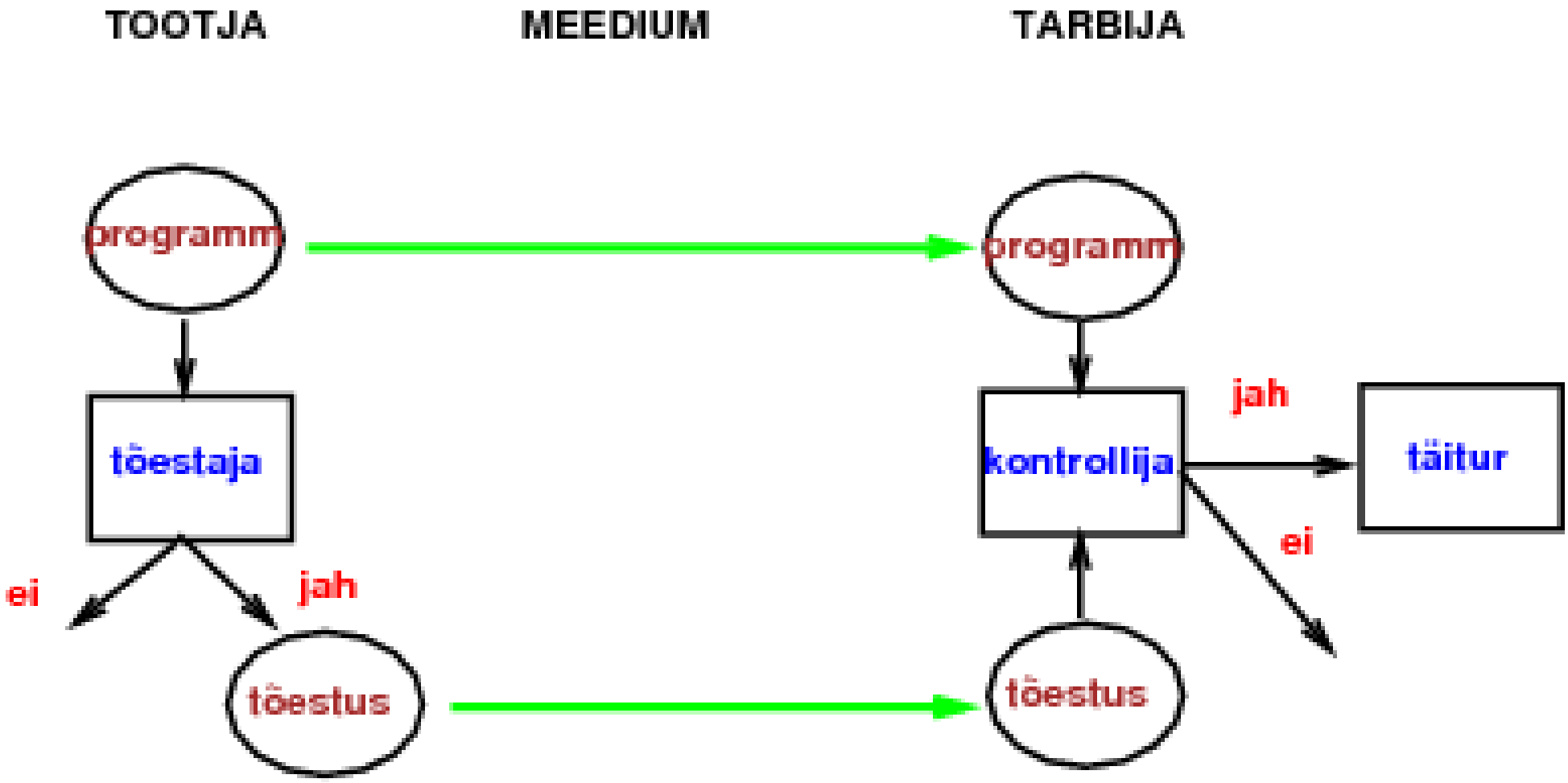
Arvutitoe, automatiseerituse vajadus

- Verifitseerimise tugitehnoloogia on võimsalt arenenud 1990ndatel, aga eriti 2000ndatel
- Tüübisüsteemid (sh sõltuvate tüüpidega programmeerimine), sertifikaatide koostamine, kontroll automaatsed
- Staatiline analüüs, mudelikontroll
- Automaatsed teoreemitõestajad (piiratud, ei leia alati tõestust)
- Interaktiivsed tõestusassistendid programmiloogikatele, universaalsed interaktiivsed tõestusassistendid

Sertifitseeritud tarkvara kasutamine

- Enne tarkvara kasutuselevõttu, peaks tarkvara tarbija (skeptik) tarkvara tootja (väitja) esitatud sertifikaadi kontrollima
- Kui tulemus on positiivne, võib tarkvara käivitada
- Kui tulemus on negatiivne, on kindlam loobuda
- Stsenaarium ei allu manipuleerimisele: kui tarkvara või sertifikaat on rikutud, siis kontroll ei õnnestu
- Tõestaja tootja poolel ei pea olema usaldusväärne. Kui tootja väljastab katkisi sertifikaate, siis tarbija avastab selle.
- Oluline on, et tõestuste kontrollija tarbija poolel oleks usaldusväärne.
- Tarbija peab usaldama ka tõestustes kasutatud programmikeele semantikat, et tegelik täiturmehhanism respekteerib seda

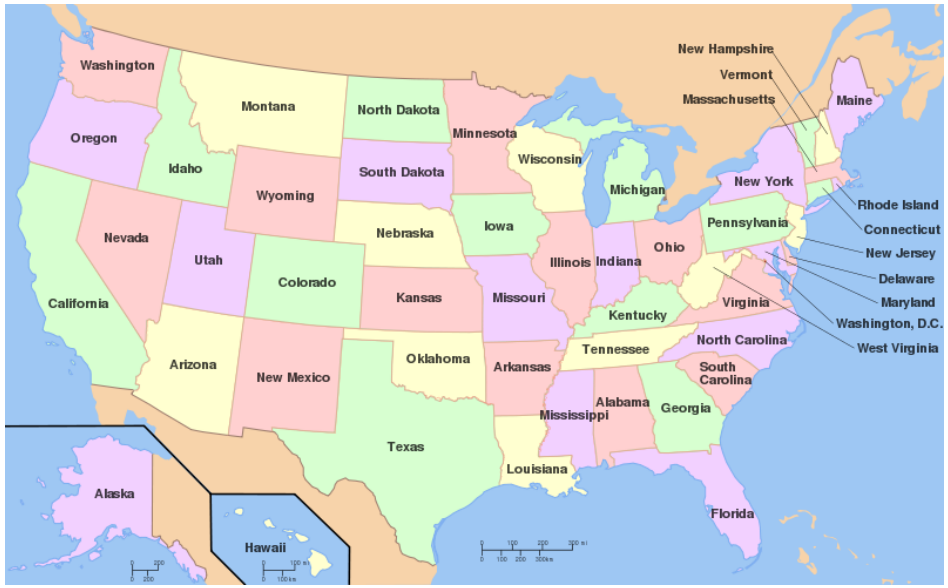
Seritifitseeritud koodi kasutamine



Sertifitseeritud teadus?

- Tarkvara on keeruline ja mahukas, sertifitseerimise ja arvutitoe vajadus arusaadav
- Aga teadus?
- Uus: sama kehtib tänapäeval siin
- Nt matemaatikas tulemused, mille tõestused on keerulise loogilise struktuuriga ja mahuka kombinatoorikaga
- Sama on probleemiks arvutiteaduses
- Vajadus on üles kerkimas teoreetilises füüsikas ja mujal

Nelja värvi teoreem



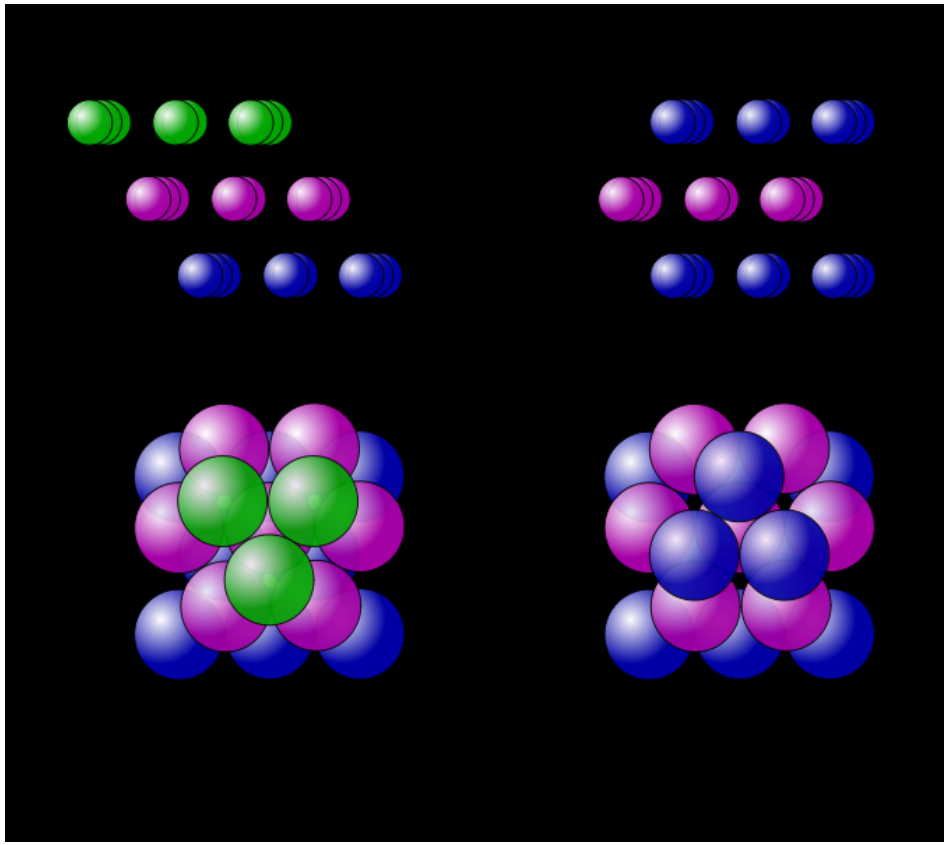
- Väide: Iga maakaart on värvitav nelja värviga nii, et naabermaad on eri värvi

- Hüpotees: Francis Guthrie 1852 (üritas Inglismaad värvida)
- Esimene vigane tõestus: Alfred Kempe 1879, viga leitud: Percy Heawood 1890
- Järgmine vigane tõestus: Peter Guthrie Tait 1880, viga leitud: Julius Petersen 1891

Nelja värvi teoreem

- Esimene ümberlükkamata tõestus: Kenneth Appel, Wolfgang Haken 1976. Tõestus kasutas arvutiprogrammi abi 1936 juhu läbivaatamiseks. Juhud olid leitud käsitsi.
- Pretsedent matemaatikas. Arvuti abiga koostatud tõestus liiga mahukas käsitsi kontrolliks. Programmi enda korrektsus, samuti juhtude sisenditeks kodeerimise korrektsus ebaselge
- Lihtsam tõestus, ent siiski juhtude läbivaatusega arvuti abil: Robertson, Sanders, Seymour, Thomas 1997.
- Arvutiprogrammi abiga koostatud ning automaatselt kontrollitav tõestus: Georges Gonthier 2005.

Kepleri hüpotees



- Väide: Sama raadiusega keradega (nt apelsinide) pakkimise optimaalne tihedus on $\pi/3\sqrt{2} \approx 74\%$, saavutav kuup-kompaktse ja kuusnurk-kompaktse paigutusega
- Hüpotees: Johannes Kepler 1611

- Esimene samm tõestuse suunas: Carl Friedrich Gauss 1831, korrapäraste paigutuste jaoks väide kehtib, aga võibolla korrapäratutega saab tihedamalt pakkida
- David Hilbert 1900 kaasas probleemi matemaatika 23 lahendamata probleemi nimekirja (osa 18. probleemist)

Kepleri hüpotees

- Tõestuse skeem: László Fejes Tóth 1953, probleem on taandatav hiigelhulga lõplike juhtude läbivaatusele
- Tõestus: Thomas Hales 1996, 250 lk märkmeid ning 3 GB arvutiprogramme, andmeid ja tulemusi
  Annals of Math retsenseeris artiklit 4 aastat; László poeg Gábori juhitud 12 eksperdist koosnenud paneel jõudis järeldusele, et nad on "99% veendunud"
- Automaatselt kontrollitava tõestuse koostamine: Thomas Hales, al. 2003, töös, projekt Formal Proof of Kepler (Flyspeck), hinnaguline maht 20 aastat

Tarkvaraproduktid vs matemaatikatulemused

- Matemaatika ei erine liiga palju tarkvarast!
- Nii tarkvara kui ka matemaatilisi definitsioone saab kirjutada ja väiteid nende kohta püstitada ilma tõestuseta (ning nii tarkvara ja matemaatikat tehakse!)
- Kui tõestus on keeruline, on selle leidmine raske intellektuaalse leidlikkuse poolest, mida see vajab
- Kui tõestus on mahukas (suure arvu juhtude läbivaatus), tuleb rasket tööd teha palju
- Kui tõestus on keeruline, on selle käsitsikontroll vigadealdis (tähelepanu hajub jm inimlikud vead)
- Kui tõestus on mahukas, on selle käsitsikontroll praktiliselt võimatu (pole ajaressurssi)
- Nii tarkvara kui ka matemaatika kvaliteet sõltuvad tõestuste üleskirjutuste rangusest, arvuti abist tõestuste koostamiseks ning automatiseeritud tõestuste kontrollist

Muud teadused?

- Sertifitseerimistehnoloogia on aktuaalne ka muudes teadustes
- Arvutiteadus: programmeerimisteooria (kompilaatorite korrektsus), infoturve (protokollide turvalisus)
  Vead on eriti kerged tulema tõenäosuslikke ja kvantarvutusi puudutavate arutelude juures
- Füüsika: teoreetiline füüsika (kvantfüüsika)

Loogilised vs empiirilised tõed

- Kui matemaatika on puhtloogiline, ei sõltu välismaailmast, siis füüsika ja muud loodusteadused on empiiriline
- Nt Newtoni mehaanika pole loogiliselt tuletatav, on vaid üks võimalikest mehaanikatest, mida katseandmed kinnitavad (teatud skaaladel)
- Kuidas saavutada kindlust empiirilistes tõdedes?
- Teatud baasseadusi tuleb usaldada (nt katseandmete põhjalt), ei saa tõestada. Kõik muud väited on tõestatavad eeldusel, et need baasseadused on tõesed
- Sama on tarkvaraga: Kui programmi kohta on tõestatud mingi korrektsusväide, siis see jääb põhinema usul, et programmide tegelik täiturmehhanism käitub kooskõlas tõestuses aluseks võetud programmikeele semantikakirjeldusega
- Programmikeele semantika on võrreldav füüsikaseadusega: mõlemad on ekstrapolatsioonid tegelikkuselt

Sertifitseerimise kultuur?

- Kui palju toredam oleks IT, kui võiksime kasutada sertifitseeritud tarkvara!
- Tehnoloogiad arendamisel: tõestust kandev kood, tõestusi säilitav/genereeriv kompileerimine
- Sertifitseeritud teadus?
- Kuidas oleks, kui teaduse hindamine ei toimuks kvantiteedi (publikatsioonide, tsiteeringute arvud), vaid kvaliteedi järgi? Loeksid ainult sertifitseeritud tulemused...
- Matemaatikas täna: antakse välja ajakirju *J of Formalized Mathematics*, *J of Formalized Reasoning*
- Arvutiteaduses: arvutiteaduse formaliseerimise "võistlused", nt programmeerimisteooria formaliseerimise kohta *POPLmark Challenge*, idee, et juhtivatele konverentsidele esitatud artiklid peaksid olema formaliseeritud, krüptoloogias "tõestatava turvalisuse" trend

Sertifitseerimise ökonoomika

- Sertifitseerimise kultuuris võib tekkida sertifitseerimise ökonoomika
- Sertifikaadid on intellektuaalne omand ja neil on väärtus
- Sertifitseerimise tööstus?
- Kas tasuta sertifikaatide ideoloogia või sertifikaatide äri koos sertifikaatide kaitsega?
- Kangusastmetega sertifikaadid: saad kindlust niipalju, kui soovid/jaksad maksta
  (tugevam väide või väiksemalt usaldatavalt baasilt ehitatud tõestus on kangemad)
- Jne

Sertifitseeritud tarkvara / arvutiteadus Eestis

- Kübl loogika ja semantika rühm
- europrojektid MOBIUS (tõestust kandev kood), FoVeOOS (objektorienteeritud tarkvara formaalne verifitseerimine), HATS (kõrgelt adaptiivne usaldusväärne tarkvara)
- tööd programmeerimisteooria formaliseerimisest: mittetermineeruvuse ja interaktsiooni semantilised struktuurid, vastavate keelte semantikakireldused, transaktsioonilise mälu algoritmid ja nende korrektsus

Kokkuvõtteks

- Inimese aruteluvõime ei ole põhimõtteliselt arvuti omast nõrgem, aga tema keerukuse ning mahu piirid on madalal
- Arvutiprogrammide abiga koostatud ja automaatselt kontrollitavad sertifikaadid võivad anda kindlust seal, kus tõestuste käsitsi kontrollimine, ammugi koostamine ei ole realistlikud
- Tarkvara korrektsuse ja ohutuse, kaasaegse matemaatika teatud osade juures on sellise lähenemise kasutuselevõtt möödapääsmatus
- Sertifitseerimise võimalused on avarad, parem tarkvara, parem matemaatika
- Ohuks on asja pöördumine äriks (nagu muu intellektuaalse vara puhul)