

**ARVUTID, KÜBERRUUM
JA TEHISMÕISTUS**

ACTA UNIVERSITATIS TALLINNENSIS

Realia et Naturalia

SARJA KOLLEGIUM

Mihkel Kangur (Tallinna Ülikool)

Mihkel Koel (Tallinna Tehnikaülikool)

Tiit Land (Tallinna Ülikool)

Romi Mankin (Tallinna Ülikool)

Hannes Palang (Tallinna Ülikool)

Margus Pensa (Tallinna Ülikool)

Ruth Shimmo (Tallinna Ülikool)

Jüri Siigur (Keemilise ja Bioloogilise Füüsika Instituut)

Raivo Stern (Keemilise ja Bioloogilise Füüsika Instituut)

Tallinna Ülikool

Enn Tõugu

ARVUTID, KÜBERRUUM JA TEHISMÕISTUS

**NOPPEID ARVUTITE
IMEPÄRASEST EDULOOST**

TLÜ Kirjastus
Tallinn 2018



ACTA Universitatis Tallinnensis

Acta Universitatis Tallinnensis. Realia et Naturalia

Enn Tõugu

Arvutid, küberruum ja tehismõistus

Noppeid arvutite imepärasest eduloost

Keeletoimetaja Siiri Soidro

Kujundanud ja küljendanud Sirje Ratso

Sarja makett: Rakett

Autoriõigus: Enn Tõugu, 2018

Autoriõigus: Tallinna Ülikooli Kirjastus, 2018

ISSN 2461-1875

ISBN 978-9985-58-847-5

TLÜ Kirjastus

Narva mnt 25

10120 Tallinn

www.tlupress.com

Trükk: Pakett

SISUKORD

Sissejuhatus	9
------------------------	---

I osa. Arvutid ja tarkvara 13

1. Eellugu, peamiselt matemaatikas	15
Algoritmid	15
Info	17
Andmed	19
Kahendarvud	22
Loogikast	25
2. Esimesed arvutid	30
Vana aja arvutid	30
Esimesed päris arvutid	33
3. Kuidas arvuti töötab	39
Arvutite arhitektuurist	40
Arvutusseade	41
Arvuti mälu	43
Arvuti abstraktne mudel ja von Neumanni arhitektuur	44
Kuidas arvuti mõistab programmi	46
Käsusüsteemid ja assembler	48
4. Arvutid jõuavad Eestisse	50
5. Arvutite põlvkonnad ja Moore'i seadus	55
Kolmanda põlvkonna arvutid	56
Moore'i seadus	58
Nõukogude arvutustehnika	59
Arvutid pärast kolmandat põlvkonda	60
6. Tarkvara vaevaline tulek	63
Andmed ja andmetüübid	64
Andmekirjelduskeeled	66
Programmeerimiskeeled ja translaatorid	69

Skriptikeeled	82
Esimesed operatsioonisüsteemid	83
Kuidas operatsioonisüsteem töötab	86
7. Miniarvutid ja andmebaasid	91
Arvuti PDP-11	91
Andmebaasid	93
8. Jaapanlased projekteerivad tarka arvutit	100
Prolog	100
Viienda põlvkonna arvutite projektid	103
LISP-arvutid	105
9. Ootamatult tulevad PC-d	107
Mikroprotsessor	107
Esimesed PC-d	109
IBM PC ja Microsoft	112
Apple'i firma	115
Sülearvutid ja tahvelarvutid	116
II osa. Internet ja küberruum	121
10. Andmeside põhimõisteid	123
11. Kuidas leiutati internet ja elektronpost	128
ARPANET	129
Internet	130
Elektronpost	132
12. Veeb ja küberruum	134
Veebi algus	135
Sajandivahetuse IT-mull	137
P2P-partnervõrgud	138
Küberruum	139
Küberturvalisus	143
13. Arvutid on järsku kõigjal	146
RFID-kiibid	147
Värkvõrk	148
Nutitelefonid	150

III osa. Tehismõistus	155
14. Tehismõistuse algus ja selle kriisid	157
Tehismõistuse kriisid	157
Ekspertsüsteemid	159
15. Tehismõistuse tänapäev	162
Närvivõrgud ja pertseptron	163
Kolm vaadet tehismõistusele	166
Arvutite üliintelligentsuse muut	170
Lisa 1. Tarkvaramudelid	175
Lisa 2. Mida pertseptron suudab ja kuidas ta õpib	185
Fotode allikad	189
Aineloend	191
Isikunimede loend	195

SISSEJUHATUS

Olen sellesse raamatusse kokku kogunud oma nägemuse arvutite eduloost, milles on oluline osa ka tarkvaral. See ei lange alati kokku üldlevinud ettekujutusega arvutite arengust, sest see pole olnud minu eesmärk. Olen arvutitega olnud õpingute ja töö kaudu seotud alates ajast, kui Nikita Hruštšov teatas 1958. aastal mõneti ootamatult nõukogude ajakirjanike ja laiema üldsuse jaoks, et arvutid ja küberneetika polegi saatanast. Minu kogemus räägib sellest, kuidas arvutid olid alguses paljas riistvara, kuidas matemaatikud alahindasid tarkvara ja selle keerukust, kuidas nooremad põlvkonnad ei imesta, et nutitelefone sisse mahub vabalt ära varem tervet arvutisaali enda alla võtnud arvuti Minsk 22 või isegi kolmanda põlvkonna arvuti IBM 360/20, seda muidugi funktsionaalselt, mitte materiaalselt.

Tegelikult oleme tänapäeval just praegu toimuva infotehnoloogilise revolutsiooni tunnistajad. Me ei tea, kuhu tehnikas ja ühiskonnas toimuvad muutused välja viivad. Mõned isegi arvavad, et arvutid võtavad peagi inimeste üle võimu. Igatahes pole infotehnoloogia põhjustatud suured muutused ühiskonnas veel lõppenud. Nendega arvestamiseks on igapäev kasulik püüda teatud määral mõista muutuste tehnilist tagapõhja kas või kõige üldisemal tasemel. Selles püüabki käesolev raamat lugejat aidata. Arvutite kõrval on raamatus eraldi osad ka küberruumi ja tehismõistuse kohta. Me võime kasutada analoogiat tehnilise revolutsiooniga, mis toimus kahe sajandi eest tänu aurumasina leiutamisele. See oli oluline osa esialgu raudteetranspordi ja hiljem autode arengus. Kuid see pole kogu tõde. Sellega kaasnesid ka uued ohud inimestele: aastas saab üksnes autoõnnetustes surma ligi 1,3 miljonit ja vigastada kuni 50 miljonit inimest. Aurumasina asemel on tänapäeval muutuste põhjustajaks arvuti. IT-revolutsiooni oluline osa on inimkonna sidumine küberruumis peaaegu silmapilk leviva info kaudu. Ka siin

on omad ohud paratamatud. Kuid millised need on? Kas tasub kõike arvutite intelligentsuse kaela ajada?

See raamat tutvustab arvutite ja nendel põhineva infotehnoloogia tormilist arengut sisuliselt, kuid ei eelda matemaatilisi või tehnilisi eelteadmisi. Tõsi küll, raamatu alguses näeme mõningaid valemeid ja skeeme, mis selgitavad seda, kuidas arvuti protsessor arvude ja teiste andmetega tehteid teeb. Samuti on võimalik küllalt lihtsalt selgitada andmetena esitatud programmi imepärast muutumist arvutile arusaadvaks ja tema poolt täidetavaks algoritmiks. Need on arvuti toimimise põhitõed, mille teadmine võiks olla igale haritud inimesele kasulik ja huvitav. Nendel teadmistel põhineb nii arusaamine sündmustest, mis toimuvad interneti põhjal tekkinud küberruumis, kui ka ennustamine, kas ja kuidas võivad arvutid saada inimesest targemaks. Kui lugejale tundub 3. peatükk siiski igav, võib piirduda selle peatüki esimeste lehekülgede uurimisega.

Raamatust näeme, et arvutite konstrueerimine sai alata siis, kui matemaatikute avastused algoritmide, info esituse ja loogika valdas olid selle ette valmistanud. Elektroonika ja pooljuhttehnoloogia kiire areng võimaldas möödunud sajandi keskpaigast alates arvutite töökiirust suurendada mõnekümnest tehtest sekundis kuni saja miljoni miljardi tehteni sekundis. Tarkvara andis arvutitele paindlikkuse, millele järgnes nende kasutusvaldkonna oluline laienemine nii, et tänapäeval on raske leida valdkonda, mida arvutid poleks mõjutanud.

Kronoloogiale on pööratud raamatus tähelepanu ainult sel määral, kui võrd see selgitab oluliste sündmuste järjestust ja omavaheolist sõltuvust. Kuid peatükid on siiski järjestatud arvutiasjanduse arengu kohaselt. Arvutite tööpõhimõtete, riistvara ja tarkvara tutvustus võtab enda alla esimesed kuus peatükki. 7., 8. ja 9. peatükk katavad möödunud sajandi seitsmekümneid ja kaheksakümneid aastaid, kui tekkisid miniarvutid, andmebaasid ja personaalarvutid. Sellesse ajavahemikku kuulub ka esimene tehisintellektibuum ja katsed konstrueerida tehisintellekti omav arvuti. Järgmised neli peatükki on pühendatud tänapäevale. Siin on juttu internetist,

veebist, küberruumist ja värkvõrgust. Viimased kaks peatükki on pühendatud tehisintellektile ja selle võimalikule arengule.

Raamatu kirjutamise käigus sai selgeks, et teoses ei ole võimalik täiesti vältida tehnilisi detaile. Tuli kirjutada protokollidest, mis määravad ära andmeside reeglistiku, samuti sellest, et andmetel on tüübid. Tuli rääkida ka programmide teatud omadustest, milleta ei saa seletada tarkvara arengut. Seetõttu võivad tehnikast ja tarkvarast kaugetele lugejatele tunduda osa peatükke kui mitte rasked, siis vähemalt igavad. Nendele inimestele mõeldes olen püüdnud iga peatüki (võimaluse korral ka peatüki alapunkti) alguses vältida tehnilisi detaile nii kaua kui võimalik. See peaks võimaldama lugeda iga peatüki algust sinnamaani, kus detailide olemasolu lugeja huvi hävitab. Ka sel viisil saab teatud ettekujutuse arvutite ja tarkvara arengust kuni tänapäevani.

Autori kogemus ütleb, et on lugejaid, kes isegi ilma vastava ala eelteadmisteta mõistavad vaevata arvutite ja algoritmide kohta kirjutatud. Kuid on ka neid, kellele on see peaaegu ületamatu raskus, nii nagu muusikaliselt andetule nootide lugemine või laulmine. Tuleb arvata, et on olemas **algoritmiline andekus**, nii nagu on olemas muusikaline andekus. Seetõttu saab autor ainult vabandada nende ees, kes tõepoolest sellest raamatust aru ei saa. Tean oma tähelepanekutest, et sel juhul ei aita isegi lugeja matemaatiline haridus. Samas võib see raamat paljudele humanitaarharidusega lugejatele osutada lihtsalt loetavaks ja kasulikuks. Sellele lugejate ringile võib raamatust rohkemgi tulu olla kui tehniliselt või matemaatiliselt haritutele, kes arvutitest niikuinii midagi ikka teavad. Head lugemist!

I OSA

ARVUTID JA TARKVARA

I. EELLUGU, PEAMISELT MATEMAATIKAS

Selleks et sisuliselt aru saada, mis on arvutid ja kuhu nende areng võib välja jõuda, läheb tarvis päris väheseid eelteadmisi. Need teadmised on inimestel tekkinud pikema aja vältel ja hoopiski mitte tehnika vallas. Seetõttu alustame juttu arvutitest hoopis mõne matemaatikas tuntud mõiste – **algoritmi**, **kahendarvude** ja **lausearvutuse** – tutvustamisega. Meil läheb vaja ka **info** ja **andmete** tähenduse seletust. Need kaks mõistet tekkisid juba tänapäeval, infotehnoloogia koidikul. Peab ütleva, et vaatamata nende erilisele tähtsusele pole info ja andmete mõisted inimestele kaugeltki nii arusaadavad, kui vaja oleks. See võibki olla üks väärarvamuste põhjus selle kohta, mida arvuti põhimõtteliselt suudab ja mida mitte.

Algoritmid

Arvutitel on pikk eellugu, mille alguseks võime lugeda antiikaja matemaatikas algoritmide avastamist. Võime näiteks tuua Eukleidese algoritmi, mis on kasutusel senini. Algoritmi nimetus pärineb küll alles 9. sajandist ja tuleneb araabia matemaatiku al Horezmi nimest, sest tema teadvustas endale algoritmi mõiste üldkujul.

Kõige üldisemas mõttes on algoritm tegevuse eeskiri. Me võime mõnel juhul piirata algoritmi mõistet nii, et vaatleme ainult info- töötluste algoritme. Siia alla kuuluvad arvutusalgoritmide, üldisemalt aga andmetöötluste algoritmide. Isegi sellise eeskirja esitamiseks on väga palju võimalusi. Kuid meil läheb tarvis ka siin toodud üldisemat algoritmi definitsiooni. Seda kas või robotite käitumise kirjeldamisel, kus sisuliselt pole tegemist arvutamisega. Mõnel juhul on algoritm hoopis oskuste esitus, nagu näeme hiljem tehisintellekti vaatlemisel.

Aastasade jooksul esitati algoritme vabas vormis inimkeeles. Näiteks positiivsete täisarvude liitmise algoritm „Tuleb alates arvu

sabast iga järguga teha numbrite liitmistehe, arvestades seejuures ka ülekannet”. Sellise algoritmi täitmist õpetati vanasti koolides eriti hoolikalt, sest polnud ei arvuteid ega isegi mitte kalkulaatoreid. See näide pole küll kuigi täpne algoritmi esitus, kuid seab antud algoritmi sisu siiski ära neile, kes mõistavad eesti keelt ja oskavad liita numbreid (ja neile veel ühte, kui on ka ülekanne). Arvutile nii algoritmi esitada ei saa. Nii jõudsimegi algoritmide täpse esituse vajaduse juurde. Sellest tuleb rohkem juttu hiljem mitmel korral.

Algoritmide täpse esitamisega hakkasid matemaatikud hoolega tegelema enne arvutite tulekut, eriti eelmise sajandi esimeses pooles. (Kas matemaatikud tõesti tundsid, et varsti tulevad arvutid?) Sellest ajast pärinevad niisugused mõisted nagu **Posti masin**, **Turingi masin**, **Markovi normaalalgoritm** ja **Churchi lambdaarvutus**. Igal juhul on nendes tegemist järgnevaga. Kõigepealt oletatakse, et on mõned tehted, mille tegemine on algoritmi täitjale (antud juhul matemaatikule) selge. Neid tehteid võib kindluse mõttes ka eraldi enne kirjeldada. Mõni tehe peaks olema ka selline, et selle täitmise vajadus sõltub varasemast arvutustulemusest. Edasi oletatakse, et algoritmi kirjeldamiseks on mingi keel, mille abil saab algoritmi täitja kirjeldusest üheselt aru. Arvutusliku algoritmi kirjeldus seisneb selles, et öeldakse, millist tehet ja milliste andmetega mingil töötamise hetkel tuleb täita. See ongi kõik. Täpselt esitatud algoritmi täitjat kutsutakse sageli masinaks, sest see toimib täpselt määratult **nagu masin**.

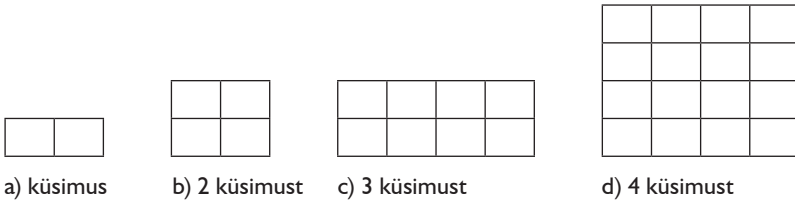
Saladus seisneb selles, kuidas öelda, millist tehet järgmiseks täita tuleb. Kui Turingi masina ja teiste teoreetiliste algoritmi-kirjelduste asemel vaadelda programmeerimiskeeli, mis on arvutite arusaadavad algoritmide eriti täpse kirjeldamise keeled, saame öelda, et täita tuleb enamikul juhtudel järgmisena kirjas olev tehe. See on üsna loogiline ja lihtsamad algoritmid ongi tehete (üldisemalt elementaarsete tegevuste) jadad. Sellised on ka tavalised toiduretseptid, mis on ju omamoodi, tõsi küll mittearvutuslikud, algoritmid.

Siiski peab olema algoritmis võimalik viidata mõnele kaugemale kohale ja öelda, et sealne tehe tuleb täita järgmisena. Kui see võimalus on tagatud ja väike lubatud (peamiselt aritmeetiliste) tehete komplekt targasti valitud, on tegemist mis tahes arvutusalgoritmi esitamist võimaldava keelega.

Alternatiivselt saame algoritme esitada nii, et algoritmi kirjeldatakse teatud koguse reeglite kaudu. Iga reegel sisaldab mingit tehet ja ütleb täpselt, millistel tingimustel tuleb selle reegli tehet täita. Selline on näiteks Eukleidese algoritm, mis alustab tööd mis tahes kahe positiivse täisarvuga ja kõlab nii: „Kui arvud pole võrdsed, siis lahutada suuremast arvust väiksem, asendada suurem arv saadud vahega ning arvutada edasi uue arvupaariga. Kui arvud on võrdsed, lõpetada töö ja anda vastuseks viimasena saadud arv.” See on kahest reeglist koosnev algoritm. Lugeja saab ise kontrollida, et selle algoritmi täitmine annab tulemuseks kahe arvu suurima ühisteguri, st suurima arvu, millega saab kumbagi etteantud arvu jäägita jagada. Proovige seda näiteks arvudega 15 ja 27. Ka sellisest reeglite keelest võib tänapäeva arvuti aru saada, kui reegleid kirjutatakse kindla eeskirja kohaselt. (Muidugi eeldame, et arvuti pole mitte paljas riistvara, vaid on varustatud programmiga, mis reegleid mõistab ja kasutada oskab.)

Info

Informatsiooni ehk info mõiste võeti laiemalt kasutusele möödunud sajandil sidetehnika arenedes. Infot on võimatu sisuliselt täpselt defineerida, sest see on fundamentaalne mõiste nagu näiteks **energia** füüsikas. Me saame siiski nimetada üht info põhiomadust: info vähendab määramatust. Eriti oluline on seejuures, et määramatuse vähenemist saame täpselt mõõta. Selleks sobib entroopia muutus mingis olukorras, kus saadakse infot. Entroopiat osatakse arvutada, kuid meie sellega siin rohkem ei tegele ning vaatleme lihtsamat, kuid olulist erijuhtu.



Joonis 1. Valik 2^n võimalusest nõuab n küsimust

Eelmise sajandi keskpaigaks oli sidetehnika juba üsna arenenud. Sideinsenerid hindasid sidekanalite läbilaskevõimet ja mõõtsid seda sümbolite arvuga, mis kanalist mingi ajaühiku jooksul läbi läheb. Kuigi kanali kaudu liiguvad sümbolid, saadakse iga sümboli saabumisel ka infot. Üheks eelmise sajandi suurimaks teadussaavutuseks sai info mõõtmise valemi

$$I = \log_2 n \quad (*)$$

sõnastamine Claude Shannoni poolt juhul, kus n on võrdtõenäosuslike võimaluste arv ja I info hulk bittides, mida vajatakse sellest võimaluste hulgast õige valiku tegemiseks. Olgu meil kaks võrdse tõenäosusega tekkivat võimalust. Ühe sellise võimaluse kindlaks-tegemine annab infot koguses $\log_2 2 = 1$ ehk ühe biti. Siit saame biti definitsiooni:

Infot, kumb kahest võrdvõimalikust juhust on õige, on üks bitt.

Valemit (*) sai kohe kasutada sidekanalit läbiva info mõõtmiseks. Kui võrdvõimalikke sümboleid on n tükki, siis iga sümbol toob $\log_2 n$ bitti infot, k sümbolit toovad k korda rohkem bitte infot. Kui näiteks kanalit läbib sekundis 125 sümbolit ja erinevaid võrdvõimalikke sümboleid on 256, siis läbib kanalit 1000 bitti sekundis. (Kontrollige!)

Valem (*) väärib natukene rohkem tähelepanu. Vaatleme olukorda õige kasti leidmisel kahe võimaliku (ja võrdtõenäose) kasti seast, mida illustreerib joonis 1a. Selle info saab kas või siis, kui

küsida, kas õige kast on vasakpoolne. Vastuseid saab olla kaks: *jah* või *ei*. Mõlema vastuse korral saadakse infot üks bitt, sest määramatus (valida oli kahest võrdsest võimalusest) kadus ära. Kui kaste on hoopis näiteks neli, nagu jooniselt 1b, on olukord keerukam. Õige kasti leidmiseks peab kaks korda küsima. Esiteks võib küsida, kas õige kast on ülemises reas. Nii saame teada õige rea. Selleks oleme ära kasutanud ühe küsimuse ja vastusega saanud ühe biti infot. Kui nüüd küsida edasi, kas õige kast on vasakpoolne, saame teada õige kasti, vajades kahe küsimuse peale kokku 2 bitti infot (kaks korda vastus *jah* või *ei*).

Joonis 1 sisaldab veel kirjeldusi selliste olukordade kohta, kus on vaja leida vastus 8 kasti ja 16 kasti kohta. Neil juhtudel läheb vaja vastavalt 3 ja 4 bitti infot. (Üsna lihtne on leida, kuidas tuleks küsida, mõelgu lugeja see ise välja.) Paneme nüüd tähele, et $2 = 2^1$, $4 = 2^2$, $8 = 2^3$, $16 = 2^4$. Seega on vajalik bittide arv kahe aste, mis annab kastide arvu. Kui nüüd võtta kahendlogaritm kõigi võrduste mõlematest pooltest, saame veel ilmsemal kujul valemit (*) kinnitavad näited. Kui arvestada ka erinevate võimaluste tõenäosusi, saadaksegi valemi (*) asemel entroopiat kirjeldav valem. Seda kõike teati info kohta enne tänapäeva arvutite teket.

Andmed

Meil on alust arvata, et materiaalses maailmas info säilib ja kandub edasi ainult materiaalsel kandjal. Võtame appi andmete mõiste. Oleme ju kõik andmetega kokku puutunud. Võime olukorda täpsustada öeldes, et info sisaldub andmetes mingil materiaalsel andmekandjal, kusjuures andmekandjateks võivad olla mis tahes märkidena kasutatavad objektid. Peale Juri Lotmani põhjanevaid töid semiootika vallas ei tohiks selline väide olla tänapäeva haritud inimesele üllatav.

Abstraktselt saame andmeid alati ümber kodeerida ning esitada neid nullide ja ühtede jadana ehk bitijadana, kasutades selleks nii palju bitte, kui osutub vajalikuks valemi (*) järgi. Praktikas võivad

andmete allikaks ja andmekandjaks, st infokandjaks olla tekstid, pildid, muusika, nelja tüüpi molekulidest (A, T, G, C) koosnevad jadad geenides, elektrilised ja keemilised signaalid jne. Üleminek füüsiliselt nähtuselt abstraktseteks andmeteks võib osutada üsna salapäraseks protsessiks. Peab märkima, et andmeallikaks võivad olla ka sotsiaalsed ja bioloogilised nähtused, mis hägustab veelgi andmete tekkeprotsessi. Meie vaatleme andmeid sellest hetkest alates, kui neid saab esitada bitijadana. Sellisteks andmeteks on kahtlemata tekstid, samuti pildid, helid ja nende jadad ning ka signaalid närvivõrgus.

Võib tunduda üllatavana, et andmete mõiste sai laialt kasutatavaks alles pärast arvutite tulekut. Seejuures muutus andmete kodeerimine bitijadadena üldlevinuks samuti alles arvutiajastul. Kodeerimine on lihtne, kui erinevaid võimalikke andmeid on lõplik hulk. Siis saame toimida järgnevalt. Võtame kahendlogaritmi arvust, mis näitab, kui palju erinevaid andmeid saab olla. Kui saadud logaritm on täisarv, ongi see kodeerimiseks vajalik bittide arv. Kui see pole täisarv, sobib kodeerimiseks kasutada bitti, mille arv on suurem kui saadud logaritm. Nüüd tuleb erinevatele andmetele seada vastavusse erinevad bitijadad, ja ongi kõik. Näiteks, kümnendnumbreid saame kodeerida nelja bitti kasutades, sest neid on kümme tükki ja $\log_2 10 = 3.1\dots$ Tavaline on võtta numbrite koodid nii, nagu seda näitab tabel 1. Teksti kodeeritakse tavaliselt selliselt, et iga sümboli jaoks kasutatakse 8 bitti. Selline bittide komplekt kannab nime bait.

Kahjuks pole teada, kuidas korraldada täpset kahendjadadega kodeerimist üldjuhul, kui võimalikke andmeid on lõpmatu hulk. Kas lugejal on teada mõni sellise lõpmatu hulga näide? Selline on näiteks pidevate funktsioonide hulk, sh ajas muutuvaid suurusi kirjeldavate funktsioonide hulk, millega puututakse kokku heli ja videosalvestusel. Kui me ei nõua kodeerimisel absoluutset täpsust, mis pole praktikas niikuinii iial teostatav, aitab meid kuulus Kotelnikovi teoreem.