

0. Sissejuhatus õppeainesse *Teoreetiline informaatika*

Alljärgnevas selgitame õppeaine *Teoreetiline informaatika* aluseks olevaid põhimõisteid *informaatika* ja *teoria* ning anname lühiülevaate õppeaine sisust.

0.1. Informaatika määratlus

Rahvusvaheliselt ei ole kokku lepitud mõiste *informaatika* ühtses definitsioonis. Kuna selle mõiste sisu on erakordsest lai, siis pakutud definitsioonid on väga üldsoinalised. Nii näiteks viitab Eesti andmekaitse ja infoturbe portaal AKIT (<https://akit.cyber.ee>) asjakohastesse ISO-standardide sõnastusele: *informaatika on teaduse ja tehnika haru, mis tegeleb arvutipõhise infotöötlustega*. Mõnevõrra konkreetsemal definitsiooni pakub Vikipeedia: *informaatika on info struktuuri, hankimist, töötlemist ja esitamist käsitlev teaduse ja tehnika haru*. Veelgi detailsemal definitsiooni esitas Eesti ülikoolide ja Teaduste Akadeemia liikmete ühine töörühm, kes 1991. aastal töötas välja informaatika ainekontseptsiooni: *Informaatika on süsteematiiline õpetus informatsiooni teisendamise algoritmilistest protsessidest, mis lisaks algoritmitooriale käsitleb nimetatud protsesside analüüsni, projekteerimise, efektiivsuse, reaalseerimise ja rakendamise küsimusi*.

Seega kõige üldisemalt võib informaatikat defineerida järgmiselt:

Informaatika on teaduse ja tehnika haru, mille objektiks on informatsioon ning selle töötlemiseks vajalikud vahendid ja meetodid.

Tänapäeval on informatsiooni töötlemise põhiliseks vahendiks arvuti. Arvuti töötab teatavasti mingi arvutiprogrammi järgi. Arvutiprogramm esitab mingis arvutile arusaadavas keelles kirja pandud ülesande lahendusalgoritmi. Seega sõltub informatsiooni töötlemine eelkõige sellest, millist algoritmi või meetodit töötlemiseks kasutatakse ning milline on kasutatava seadme (arvuti) võimekus seda algoritmi täita. Ühe ja sama ülesande lahendamiseks võib kasutada erinevaid algoritme, mis võivad üksteisest oluliselt erineda, seda nii algoritmi lihtsuse kui ka täitmiseks vajaliku ressursi – eelkõige vajaliku mälumahu ja lahendamiseks kuluva aja – pooltest. Vähe sellest, osutub, et teatud ülesanded polegi lahenduvad, või pole lahenduvad mõistliku aja jooksul. Seetõttu on arusaadav, et informaatika alased süsteematiilised uuringud said hoo sisse alles peale arvutite loomist ja nende kasutamisel tekkinud praktiliste probleemide ilmnemist. Süsteematiiline teoria arendamine oli tingitud eelkõige vajadusest jõuda selgusele üldistes seaduspärasustes ja põhimõtetes, mida on vaja arvestada nii infotöötluse seadmete – sh arvutite ja nende komponentide – konstruktsioonil kui ka nende seadmete tööd juhtiva tarkvara loomisel.

Sarnaselt informaatika üldisele definitsioonile, mille kohaselt see on teaduse ja tehnika haru, võib informaatika oma olemuselt jagada kahte põhiharju:

1) teoreetiliseks - informatsiooni töötlemisel aluseks olevate teoreetiliste lähteallustega ja vastavate formaalsete mudelite ja meetodite uurimine;

2) praktiliseks - informatsiooni töötlemise vahendite (nii riist- kui tarkvaralistele) loomine ja rakendamine.

Teoreetilise informaatika lätted on erinevates teadusharudes, mis rohkemal või vähemal määral käsitlevad informatsiooni töötlemist, eelkõige:

- bioloogia (näiteks neuronvõrkude/närvivõrkude mudelid, mis modelleerib inimaju toimuvate impulsidate liikumist)
- inseneriteadused (elektriliste vooluringide teoria)
- matemaatika (loogika ja algoritmitooria, diskreetne matemaatika)
- keeleteadused (formaalsed keeled ja grammatikad)
- infoteadused (informatsiooni kogumine, analüüsamine, korrasamine/struktureerimine jne).

Informaatika põhiharud jagunevad omakorda alavaldkondadeks. Mõnda alavaldkonda on lühidalt kirjeldatud käesoleva peatüki lõpuosas.

Informaatika mõiste kujunemist ning praktilise informaatika üht olulist aspekti - rakendusinformaatikat (st informaatika rakendamist teistes valdkondades) - käsitleme lühidalt ainekursuse ”DTI6001.DT Õppimine kõrgkoolis” raames.

0.2. Teoria määratlus

Sõna *teoreetiline* kursuse nimetus ”Teoreetiline informaatika” tähendab seda, et kursus tugineb vastavale teooriale. Mõiste *teoria* on mitmetähenduslik ning defineeritakse erinevates valdkondades erinevalt. Käesolevas kursuses lähtume teoria järgmisest määratlusest:

Mingi valdkonna teoria all mõistame selle valdkonna üldistatud objektide (st formaalse/abstraktsete objektide) ja nendevaheliste seoste kohta esitatud väidete süsteemi, mis on tuletatud teatud eeldustel formaal-loogiliste meetodite abil.

Kuna teooriad vaatlevad üldistatud (formaalseid) objekte, siis on need rohkem või vähem hüpoteetilised, täpselt sellisel kujul nad praktikas üldjuhul ei esine. Teorias nimelt defineeritakse uuritavad objektid vaid vastavaid reaalseid objekte kirjeldavate kõige olulisemate tunnustele abil, mistõttu teoria objekte võib käsitleda kui teatud reaalsete objektide **mudeleid** (mingi objekti mudeliks nimetatakse selle objekti mistahes lihtsustatud/üldistatud esitust). Seega ei käsitle käesolev kursus mitte niivõrd informatsiooni töötlemise vahendeid endid, kuivõrd nende lihtsustatud mudeleid.

Järgnevalt selgitame mudelite rolli keeruliste süsteemide uurimisel.

Reaalselt esinevad nähtused, objektid või nende süsteemid võivad koosneda väga paljudest – lausa tuhandetest – elementidest ning nende elementide koostöimet mõjutavatest seostest, mistõttu kõikide elementide ja seoste arvestamine on nende objektide/süsteemide uurimisel kas võimatu või vähemalt äärmiselt keeruline ja töömahukas. Samas sageli osutub, et uuritav objekt või selle mingi huvi pakkuv aspekt on suhteliselt hästi kirjeldatav vaid mingi väikese arvu elementide ja nende elementide vaheliste seoste abil. Seetõttu võib keskenduda vaid uuritava objekti kõige olulisematele elementidele/seostele, mis selle objekti olemust või käitumist kõige rohkem määrapavad. Teiste sõnadega: ei uurita mitte mingit nähtust ennast, vaid luuakse selle lihtsustatud, kuid samas piisavalt adekvaatne mudel ning uritakse loodud mudelit. Saadud uuringutulemused interpreteeritakse seejärel realses kontekstis. Piltlikult võib seda protsessi kujutada järgmiselt:



Olenevalt vaadeldava objekti olemusest ning soovitud eesmärgist kasutatakse mudelite loomiseks ja uurimiseks erinevaid esitusviise ja vahendeid. Näiteks kui objektiks on auto ja eesmärgiks on saavutada selle võimalikult väike õhutakistus, siis piisab uurida vaid selle auto kerekuju; kõik auto sees asuv, k.a. seal asuv ülikeeruline elektronika, ei mõjuta õhutakistust vähimalgi määral. Mudeliks võib sel juhul võtta näiteks auto väliskuju kopeeriva kujundi.

Juhul, kui mudel on esitatud mingi formaalse keele abil, räägime *formaalsest mudelist* (formaalseid mudeleid käsitlevat teooriat nimetatakse *formaalseks teooriaks*); kui mudel on kirjeldatud matemaatika keeles (võrrandite, võrratuste või avaldiste abil), räägime *matemaatilisest mudelist*.

Näide 1. Kui eesmärgiks on leida uuritava objekti kõrgus maapinnast, siis võime seda mõõta ajaga, mis kulub selle objekti langemisel maapinnale jõudmiseks; objekti kõrguse maapinnast matemaatiliseks mudeliks võime võtta vaid ühe tunnusega kirjeldatud valemi $s = gt^2/2$. Teame, et see (objekti kõrguse matemaatiline) mudel on enam-vähem adekvaatne vaid väikeste kõrguste korral; suuremate kõrguste korral tuleb mudel konstrueerida õhutakistust arvestava suurema arvu tunnuste abil.

Näide 2. Kinesioloogia on teadus inimese liigutustegevusest (motoorikast). Inimese liigutustegevuse kirjeldamiseks ei pea kasutama füüsolist inimest, vaid sageli piisab vaid "kriipsujukust". Seda viimast võib käsitleda kui inimese graafilist mudelit. Esitades "kriipsujuku" elementideks olevate sirglöökude algus- ja lõpp-punktid ruumikoordinaatide abil, saame luua inimese füüsilise kuju formaalse mudeli. See omakorda võimaldab koostada erinevate liigutustegevuste formaalseid mudeliteid ja neid analüüsida. Näiteks kiirjooksu üheks võimalikuks mudeliks on

$$v = 382 + 0,053x_{11}^2 + 0,0052x_5^2 - 5,203x_9 - 0,006x_{10}^2 + 2,55x_6 - 0,0014x_8^2 + 0,0085x_1x_2 + 0,000038x_1x_7^2 + 0,0000014x_3^2x_4^2 - 0,000017x_{11}^2x_{12},$$

kus muutujateks x_i on inimese kehal olevate teatud punktide abil arvutatud suurused (ülalolevas näites: x_1 - reiepiikkuse % kerepiikkusest, x_4 - tugijala põlvenurk 1. faasis, x_{11} - kere minimaalne kaldenurk jne).

Mudelite konstrueerimiseks on kasutusel suur hulk erinevaid vahendeid ja meetodeid; näiteks võib matemaatiline mudel olla loodud statistilise analüüsiga või funktsionaalse sõltuvuse kindlakstegemise teel.

Mudelite konstrueerimisel on oluline leida tasakaal mudeli lihtsuse ja mudeli adekvaatsuse vahel. Mida lihtsam on mudel, s.t. mida väiksema arvu tunnustega on see kirjeldatud ja mida väiksem on tunnustevaheliste seoste arv, seda lihtsam on mudelit uurida, kuid seda ebatäpselt kirjeldab mudel modelleeritavaid objekte ning seda problemaatilisem on mudeli uurimisel saadud tulemuste ülekandmine reaalsesse konteksti: näiteks näites 1 toodud valem ei ole rakendatav suurte kõrguste korral.

Tagamaks teoria võimalikult lai rakendatavus, käsitletakse teoria raames reeglina objekte/nähtusi ja nende üldkehtivaid omadusi, s.t. teoria raames ei vaadelda vaid mingitele antud klassi üksikutele objektidele või nähtustele omaseid tunnuseid.

Kokkuvõtteks: mistahes teoria uurimisobjektid on teatud mõttes abstraktsioonid, on reaalsete objektide mudelid. See asjaolu annab ka võtme vastamaks küsimusele teoria tähtsusest: **teoria tähtsus seisneb selles, et see lihtsustab käsitletava valdkonna uurimist ja võimaldab kindlaks teha uurimisvaldkonnas üldkehtivaid omadusi ja seaduspärasusti.**

0.3. Lühiülevaade kursuse *Teoreetiline informaatika sisust*

Käesoleva kursuse põhisisuks on informatsiooni töötlemise vahendite (alates lihtsast detektorist/kontrollerist kuni universaalsete arvutiteni ja arvutiprogrammideni) formaalsete mudelite uurimine; seetõttu kannab see õppaine ka teoreetilise informaatika nime. Keskendume eelkõige probleemidele, mis seonduvad järgmiste mõistetega: lõplikud automaadid (*sh* magasimnälu automaatid ja Turingi masinad), formaalsed keeled ja grammatikad ning algoritmide keerukus. Kirjeldame järgnevas neid mõisteid lähemalt.

Lõplikud automaadid ja regulaarsed keeled. Lõplik automaat on diskreetsete sisend- (ja väljund)andmetega lõpliku seisundite arvuga süsteemi formaalne mudel.



Selliseks on näiteks ukse koodluku kontroller, mis peab sisendsümbolitest sõltuvalt tagama ukse luku avamise või seda mitte tegema. Kõigepealt vaadeldi sarnaseid struktuure neuronvõrkude ja vooluringide uurimisel, edaspidi ka kompilaatorites (näiteks võtmesõnade ja muutujanimede äratundmine arvutiprogrammides), aga samuti tekstiredaktorite, otsiprogrammide jne loomisel.

Magasiniga automaadid ja kontekstivabad grammatikad. Magasiniga automaat on teatud tüüpi mälupesa (nn. magasinmälu) omav lõplik automaat.



Osutub, et iga magasiniga automaat on kirjeldatav teatud kontekstivaba grammatica abil (viimase aluseks on lõplikuelemendiline tähestik, mille abil moodustatakse teatud reeglite abil sõnad). Kontekstivabad grammatikad on omakorda vahenditeks näiteks programmeerimiskeelte spetsifikatsioonide ja kompilaatorite loomisel.

Algoritmide täitjad ja piiranguteta grammatikad. Lubades automaadil sisendsümboleid töö käigus muuta ja varustades automaadi potentsiaalselt lõpmatu suurusega mäluga, jõuame mistahes algoritme täitvate automaatideeni.



Viimased võib vaadelda kui arvutite formaalseid mudeliteid. Nagu eespool mainitud, on arvutiprogramm mingis programmeerimiskeeltes kirjutatud mingi ülesande lahendusalgoritm, s.t. mingist lõplikust arvust sammudest koosnev eeskiri, mis, rakendatuna teatud algandmetele, viib nende algandmete poolt üheselt määratud lõppitulemusele. Et nii ebatäpsese definitsioonile on raske ranget teooriat ehitada, siis ongi esitatud mitmeid (omavahel ekvivalentseid) algoritmide täitjate – st arvutite – formaalseid mudeliteid, millest käesolevas kursuses vaatleme Turingi masinaid. Osutub, et mistahes Turing masina poolt täidetav algoritm on kirjeldatav teatud piiranguteta grammatica abil.

Keerukuse teoria. Mistahes ülesande lahendamisel on olulise tähtsusega, kuipalju ressursse selle ülesande lahendamiseks kulub. Keerukuse juures vaadeldakse eraldi nn. ajalist ja mahulist keerukust. Ajaline keerukus näitab, kui palju samme (s.t. aega) ülesande lahendamiseks kulub, mahuline keerukus aga, kui palju on ülesande lahendamiseks vaja mäluruumi. Ülesandeks on antud probleemi lahendamiseks võimalikult vähe ressursse nõudva algoritmi leidmine.

0.4. Täiendav materjal

Mõnikord eristatakse teoreetilist informaatikat mõnevõrra laiemast mõistest *teoreetiline arvutiteadus* (*theoretical computer science*). Nimetusel ”teoreetiline informaatika” pole ühest ingliskeelset vastet; ülalnimetatud probleeme käsitleva valdkonna tähistamiseks sobivaim ingliskeelne nimetus oleks *theoretical computer science* või ka *theory of automata and computation* (automaatide ja arvutamise teoria). Vahel kasutatakse ülalnimetatud valdkonna tähistamiseks ka termineid *automata theory* või *theory of computation*. Kuigi kohati kasutatakse ka mõistet *theoretical informatics*, ei ole see inglise keeles üldkasutatav; rohkem on see levinud saksa (*Theoretische Informatik* ja *Computerwissenschaft* on kasutusel sisuliselt sünnonüümidena) ja mitmes teiseski keeles.

Teoreetilise informaatika kursuse õppimisel on sobivamateks lisamaterjalide allikaiks järgmised raamatud: 1) ”Introduction to Automata Theory, Languages and Computation” (John E.Hopcroft, Rajeev Motwani ja Jeffrey D.Ullman) ja 2) ”Introduction to the Theory of

Computation” (Michael Sipser). Käesolevas konspektis kasutatavad tähistused on kooskõlas nimetatud raamatutes olevatega; neist raamatutest pärineb ka osa konspektis kasutatavatest näidetest ja praktikumides lahendatavatest ülesannetest.

Ka on veebis vabalt kätesaadav suurel hulgal teemakohast õppematerjali. Näiteks võib kasutada sisuhaldussüsteemi Internet Archive (<https://archive.org/>), tehes seal sobivate märksõnadega otsinguid. Näiteks märksõna ”pumping lemma” sisestamisel kuvati 22 allikat, mille seas oli nii õpikuid, audiofaile kui ka teadusartikleid.

Teoreetilise informaatika skoobi määratlemisel on palju erinevaid seisukohti. Näiteks SIGACT (*ACM Special Interest Group on Algorithms and Computation Theory*), mis on oma missioonina sätestanud teoreetilise informaatika edendamise, nimetab selle olulisemate alamvaldkondadena järgmisi: automaatide teoria, algoritmid, keerukusteooria, paralleelarvutus, krüptograafia, andmestruktuurid, väga suurte integraalskeemide teoria, masinõpe, bioinformaatika (*computational biology*), arvutigeomeetria (*computational geometry*), informatsioniteooria, kvantarvutus, arvutialgebra ja arvuteooria, programmide semantika ja verifitseerimine, juhuslikkuse teoria (*theory of randomness*). Seega käsitleb käesolev kursus vaid ülimalt väikest osa teoreetilise informaatika skoobist, olles sisuliselt sellesse valdkonda sissejuhatavaks kursuseks.

Juhtivate rahvusvaheliste IKT-ühenduste IEEE (*Institute of Electrical and Electronic Engineers*) ja ACM (*Association for Computer Machinery*) koordineerimisel on välja töötatud seitse bakalaureusetaseme (*undergraduate degree*) IKT raamõppekava: SE2014 (Software Engineering, valmis aastal 2014), CE2016 (Computer Engineering, 2016), CS2023 (Computer Science, 2023), CSEC2017 (Cybersecurity, 2017), CCDS2021 (Data Science, 2021), IT2017 (Information Technology, 2017), IS2020 (Information Systems, 2020). Käesoleva kursuse teemad sisalduvad nendest kahes - SE2014 ja CS2023.

Vt <http://www.acm.org/education/curricula-recommendations>.

Kommentaar 1. Erinevad institutsioonid, klassifikaatorid ja käsitlused struktureerivad teoreetilise informaatika valdkonda erinevalt. Nii näiteks teadusartiklite alusel automaatselt genereeritav *Computer Science Ontology* jagab teoreetilise informaatika (*theoretical computer science*) vaid kolme alavaldkonda: 1) arvutuslik efektiivsus (*computational efficiency*), 2) formaalsed keeled ja automaadid (*formal languages and automata theory*) ja 3) graafiteooria (*graph theory*) (<http://cso.kmi.open.ac.uk>). Käesolev kursus keskendub valdkonnale 2).

Kommentaar 2. Hoopis erineva jaotuse pakkus 1990-ndate aastate alguses akadeemik Enn Tõugu initsiativil loodud eespool mainitud ülikoolide ja Teaduste Akadeemia esindajatest moodustatud töörühm, mis soovitas ülikoolide teoreetilise informaatika kursustega katta järgmised teemad: lahenduvus ja keerkus, termiteisendused, algebralised spetsifikatsioonid, programmide semantika esitusviisid, semantilised hulgad ja denotatsioonisemantika, funktsionaalprogrammide algebra, paralleelarvutused, andmemudelid ja teadmussüsteemid, programmide konstruktoorimine ja transformeerimine. Nendest teemadest katab käesolev kursus vaid väikese osa; osa teemasid käsitletakse aga teistes kursustes.

Teoreetilise informaatika probleemid kuuluvad suuremal või vähemal määral üheaegselt ka matemaatika valdkonda, mistõttu nõuab nende käsitlemine häid matemaatilisi oskusi. See on eriti mõistetav, kui pidada silmas valdkonna inglisekeelset nimetust ”theory of computation” (arvutamise teoria).

Järgnevas kirjeldame lühidalt teoreetilise informaatika mõnda alavaldkonda (see ei ole kaugetki ammendav), mille lähem käsitlemine ei mahu käesoleva kursuse raamidesse.

Programmide semantika. Programmeerimiskeelte (nagu ka kõikide muude keelte) korral eristatakse mõisteid ”süntaks” ja ”semantika”. **Süntaksi** all vaadeldakse programmeerimiske struktuuri, s.t. seda, kas programm on formaalselt korrektne. **Semantika** all vaadeldakse programmeerimiske tähendust, s.t. seda, mida programmeerimiskodeksid täitmisel sisuliselt teevad. Kui programmeerimiskele süntaks on praktiliselt alati antud formaalse reeglite abil (lausete üldkujud), siis semantika on enamasti antud mitteformaalse selgituse abil (”tsükkel on programmeerimiskodeksi, mida tädetakse kuni teatud tingimus on töene”). Selline mitteformaalne semantika selgitus on mittetäielik ning loob võimaluse programmeerimiskodeksi mitteüheseks mõistmiseks ja seega mittekorrektsete programmeerimiskodeksi kirjutamiseks. Enamlevinud on 3 semantika esitusviisi: 1) **denotatsioonsemantika** (ingl.k. *denotational semantics*) korral esitatakse keele iga lause tema tähendust väljendava mingi teise keele lause abil, s.t. programmeerimiskodeksi tähendust kirjeldatakse teatud (osalise) funktsiooni abil, mis teisendab lähtekeele lause mingi teise keele lauseks (analoog programmeerimiskodeksi kompileerimisega); 2) **operatsioonsemantika** (*operational semantics*) kirjeldab, kuidas esitada programmeerimiskodeksi järjestikku tädetavate sammude jadana; 3) **aksiomaatilise (deduktivse) semantika** (*axiomatic semantics*) korral esitatakse programmeerimiskodeksi matemaatilise loogika tehete ja tuletusreeglite abil elementaaroperaatorite kaudu; viimaste tähendus defineeritakse aksioomide abil.

Programmide konstruktsioon ja verifitseerimine. Kasutatavad meetodid on erinevad, vastavalt programmeerimiskele tüübile, näiteks funktsionaalprogrammeerimine või loogika programmeerimine.

Andmemudelid ja teadmussüsteemid (intelligentsed süsteemid). Andmemudelite all mõeldakse reaalse maailma abstraheerimisel (modelleerimisel) saadud teadmisi, mis kirjeldavad vastavate andmete struktuuri, andmetevahelisi seoseid ja nendega opereerimist. Rakendatakse mingitele teatud omadustega objektide hulgale. Üheks enamlevinud andmemudeliks on relatsiooniline andmemudel, kus iga objekt on määratud teatud seose elemendina. **Relatsiooniliste andmebaaside teooriat** võib vaadelda omaette valdkonnana; viimases püütakse organiseerida erineva struktuuriga andmehulki, nii et nende edaspidine töötlus oleks võimalikult efektiivne.

Termiteisendused. Käsitleetakse avaldiste arvutamist (ka lõpmatust arvust sammudest koosnevaid).

Arvutigeomeetria. Suhteliselt uus, kuid kiiresti arenev valdkond. Leiab rakendust arvutigraafikas, robootikas, väga suurte integraalskeemide konstruktsioonidel jne.

Robootika. Põhiprobleemiks on keeruliste komplekside (robotite) loomine, mis oleksid võimalised sensorite abil koguma ümbritsevast keskkonnast informatsiooni, seda analüüsima ja selle alusel planeerima ja täitma neile pandud ülesandeid.

Graafiteooria, arvuteooria jm algoritmid. Näiteks algoritmid, mis leiavad graafis lühima tee, neljavärviprobleemi lahendus jne.

Krüptoloogia. Probleem, kuidas šifreerida ja dešifreerida informatsiooni. Formaalselt vaadeldakse šifreerimisalgoritm $E(K, M)$, kus K on mingi võti ja M šifreeritav tekst ning dešifreerimisalgoritm $D(K, C)$, kus C on šifreeritud tekst. Seejuures šifreerimissüsteem on usaldusväärne, kui, teades šifreerimisalgoritm E , kuid mitte võtit, ei ole võimalik leida teksti M või selle osa. Klassikaliseks (enne 1970. aastat) oli üks levinumaid kasutuselolnud süsteeme järgmine: Võti K on sama pikk kui tekst M . Edastatakse $C = M \otimes K$, mis on saadud vastavatel kohtadel olevate bittide summast mod2 leidmisel. Et $M = C \otimes K$, siis vastuvõtjal (teades võtit K) on lihtne esialgne tekst dešifreerida.

Väga suурte integraalskeemide teoria. Suhteliselt uus valdkond. Põhilised probleemid: 1) mikroskeemi disain, selle funktsionaalne käitumine (lülituste kiirus, energiatarve, elementide omavaheline paigutus jne); 2) arhitektuur, mis tagaks töötluse kõrge parallelusastme.