

Friedrich A. Kittler

PROGRAMINĖ ĮRANGA NEEGZISTUOJA

The Eastern World is exploding (Rytų pasaulis sprogsta) – dainavo Barry's McGuire'as. Pirmąsyk jis tai darė pašėlusiam septintajame dešimtmetyje, norėdamas per vinilinę plokštelę ar kasetinį magnetofoną atkalbėti visus savo draugus nuo netikėjimo, kad gyvename žlugimo išvakarėse. Antrąsyk – po puikaus elektroninio perdurbimo, pavertusio jo seną vinilo dainą skaitmeniniu *Armed Forces Network* [ginkluotųjų pajėgų tinklo] Dharane hitu, bandydamas ultratrumposiomis bangomis vakariečius Operacijos „Audra dykumoje“ karius atkalbėti nuo netikėjimo, kad jie (arba mes) gyvena žlugimo išvakarėse¹.

McGuire'as (ar veikia skaitmeninis signalų procesorius, galėjęs be pėdsakų ištrinti jo neigimą, virtusį fonografiškai nemirtingu) antrąsyk tebebuvo teišus, bet tik todėl, kad sproginiai neturi jokios reikšmės. Nesvarbu, kad naftos grėžimo bokštai ar raketos *Scud*, tos tiesioginės V2 vaikaitės, išleikia į orą. Rytai gali ramiai sau sprogti, juk svarbu tik tai, kas šiuo metu vyksta Vakarų pasaulyje: visų pirma aukštųjų technologijų implozija, ir jos rezultatas – signifikantų erdvės, kuri kitu atveju turbūt dar tebesivadintų Pasauline dvasia (*Weltgeist*), implozija. Jei nebūtų kompiuterinės technikos, nebūtų ir dekonstrukcijos, sakė Derrida Zygene. Raštai ir tekstai nebeegzistuoja patiriamuose laikuose ir erdvėse, vien tik kompiuterių tranzistoriuose. O kadangi didvyriškais Silicio slėnio veiksmams pavyko juos sumažinti iki submikronų lygio (vadinasi, suspausti iki mažesnių nei mikrometras mastų), realią mūsų rašymo vietą galima aprašyti tik fraktalinės geometrijos sąvokomis:

¹ Ačiū Wolfgangui Hagenui (iš *Bremeno radijo*), transliavusio šių dviejų *Eve of Destruction* variantų tekstinį palyginimą savo radijo laidos klausytojams.

kaip raidžių tarpusavio panašumą, išlikusį per šesetą dešimtmečių, skiriančių patato dydžio įmonių reklaminius plakatus nuo tokių pačių raidžių taškinio dydžio tranzistoriuose. Alfabetinėje istorijos pradžioje tarp kupranugario (*Kamel*) ir jo hebrajiškos raidės *Gamel* tėra du su puse dešimtmečio, tuo tarpu sumažinus visus ženklus iki molekulinų mastų, rašymo aktas išnyko.

Kaip gerai žinome, tik nesakome, žmonės daugiau neberašo. Raštas, toji keista programinės įrangos (*Software*) rūšis, kamavosi nuo nepagydomo panaudojimo ir paminėjimo painiojimo. Iki pat Hölderlino *Himnų* laikų vien tik žaibo paminėjimas atrodė pakankamas pagrindas galimam poetiniam jo panaudojimui². Šiandien, šitam žaibui virtus elektra, žmonių rašymas vyksta mikroskopiniais įrašais, kurie ne tik elektroninės litografijos būdu įdeginami silicyje, bet ir skirtingai nei istoriniai rašymo reikmenys, patys pajėgia skaityti ir rašyti.

Vadinasi, paskutinis istorinis rašymo aktas galėjo būti tuomet, kai aštuntojo dešimtmečio pabaigoje *Intel* inžinierių komanda, vadovaujama dr. Marciano E. Hoffo, pasitiesė keletą tuzinų kvadratinų metrų³ braižybos popieriaus ant nuvalytų garažo grindų Santa Klaroje, norėdama nubraižyti pirmojo integruoto mikroprocesoriaus aparatinės įrangos (*Hardware*) architektūros planą. Šis rankinis dviejų tūkstančių tranzistorių ir jų jungčių maketas antruoju, jau mechaniniu, žingsniu buvo sumažintas iki miniatiūrinio lusto dydžio ir trečiuoju žingsniu, elektroninėmis-optinėmis mašinomis, įrašytas į silikono paviršių. O kai galutinis produktas – 4004 – nuo to laiko visų mikroprocesorių prototipas, ketvirtuoju žingsniu užėmė savo vietą naujajame staliniame skaičiuotuve, sukurtame *Intel* užsakovo iš Japonijos, galėjo atsiverti mūsų postmodernioji rašymo erdvė.

Taigi, turint omenyje aparatinės įrangos šiandieniuose mikroprocesoriuose sudėtingumą, rankinės projektavimo technikos nebeturi jokių galimybių. Norėdami kurti kitą kompiuterių kartą inžinieriai nebebraižo, bet projektuoja kompiuteriais: geometrinių naujausios kartos skaičiuotuvų pajėgumų pakanka jų įpėdinių topologijai projektuoti. Tad žingsniai tų, kurie ir tave išneš, vėl skamba ties durimis.

Tačiau primityvūs Marciano E. Hoffo techniniai piešiniai pateikė beveik tobulą Turingo mašinos pavyzdį. Nuo Turingo disertacijos 1937 m. kiekvieną skaičiavimo veiksmą – atliktą nesvarbu kieno, žmonių ar mašinų – galima formalizuoti kaip tam tikrą skaičių komandų, veikiančių be galo ilgoje popieriaus juostoje, pilnoje diskrečių ženklų. Turingo koncepcija⁴ aprašanti tokią popierinę mašiną, kurios operacijos apima tik rašymą ir skaitymą, judėjimą pirmyn bei atgal, pasirodė esanti

² Plg. Hafki 1993.

³ 1978 m., projektuojant procesorių *Intel 8086*, jo brėžiniai užėmė 64 kvadratinus metrus braižybos popieriaus. Plg. Schrödl 1990: 102.

⁴ Plg. Turing 1987 (1937): 40f.

matematinis visų apskaičiuojamų funkcijų ekvivalentas ir tikrąja, t. y. mašinine, prasme išstūmė nekaltą profesionalų „kompiuterio“ apibūdinimą⁵. Universaliooms Turingo mašinoms užtenka pateikti bet kurios kitos mašinos aprašymus (programą), kad jos galėtų efektyviai tą mašiną imituoti. O kadangi nuo Turingo laikų jau galima visiškai abstrahuotis nuo abiejų įrenginių aparatinės įrangos skirtumų, vadinamoji Churcho-Turingo hipotezė savo stipriausia, t. y. fizikine, forma skelbia, kad pati gamta yra Universalio Turingo Mašina.

Šis teiginys savaime turėjo pasekmių – jis padvigubino aparatinės įrangos imploziją programinės įrangos eksplozijos sąskaita. Nuo tos akimirkos, kai tapo įmanoma gaminti kompiuterius – nuo 1943 m. su vakuuminių vamzdžių technologijomis, nuo 1949 m. su tranzistoriais – egzistavo ir problema: kaip aprašyti ir perskaityti iš esmės neperskaitomas skaitymo-rašymo-mašinas. Kaip gerai žinoma, sprendimas vadinasi programinė įranga, kitaip, aukštesnių programavimo kalbų plėtotė. Senovinis monopolis, kai kasdienė kalba funkcionavo kaip savo pačios metakalba, nepriimdama nieko kito kaip kito, sugriuvo ir užleido vietą naujai programavimo kalbų hierarchijai. Šis postmodernus Babelio bokštas⁶ dabar stūkso visur – nuo paprasčiausių komandų kodų, kurių lingvistiniai plėtiniai tebėra aparatinės įrangos konfigūracijos, per assemblerius, kurie yra tų pačių komandos kodų plėtiniai, iki pat vadinamų standartinių kalbų, kurių plėtiniai, per visus galimus aplinkkelius – interpretatorius, kompiliatorius ir saistykles – taip pat vadinami „assembleriais“. Taigi, rašymas, kaip programinės įrangos kūrimas, šiandien tapo fraktalinės geometrijos atrasta begaline panašumų į save grandine. Skirtumas nuo matematinio modelio nebent toks, kad fiziškai-psichologiškai neįmanoma visų šių sluoksnių pasiekti. Moderniosios medijų technologijos, dar nuo kino juostos ir gramofono, iš esmės yra sukurtos juslinei percepcijai pakirsti. Mes tiesiog nebežinome, ką daro mūsų rašymas, o labiausiai tuomet, kai programuojame.

Šiai būklei iliustruoti pakanka vien kasdienių pavyzdžių, tokių kaip teksto apdorojimo programa, iš kurios liejasi mano žodžiai. Teatleidžia Palo Alto *genius*

⁵ Kiek toli pasistūmėjo šis proveržis, demonstruoja *Intel's Programmer's Reference Manual* autoriai (ar užrašytojai): pavyzdžiui, komanda $f2x-1$ – tai yra, įvesties kvadratas minus 1 – išversta į kasdienę anglų kalbą tampa ne „Compute 2^x-1 “ (*skaičiuok 2^x-1*), bet „Computer 2^x-1 “ (*kompiuteris 2^x-1*). Plg. Intel Corporation, 1989: 4–9, taip pat Intel Corporation, 1990: 26–72.

⁶ Plg. Hagen 1989: 221: „Lingvistinėje Neumanno mašininės logikos struktūroje jau įtvirtinta divergencija tarp programinės įrangos ir programinės įrangos vadovėlių, taip nuo 1945 m. kompiuterių veikimo kalbų Babelio bokštas kyla vis aukštyne, o jų vartojimas turi vis mažiau ką bendro su prasminga mašininės kalbos sandara. Šis programinės įrangos bokštas, pilnas nedokumentuotų klaidų, beviltiškai suveltų dialektų, ir daugybės kalbinių aktų, kurių jau niekas nebegali supaisyti“. UNIX ekspertas piešia mažiau tikslų, tačiau desperatiškesnį paveikslą: „Praėjus kuriam laikui visoms operacinėms sistemoms būdingas „tam tikras užterštumo lygis“. Jos išplinta įvairiomis kryptimis ir primena griuvėsius, kuriuos galima sulaukyti tik milžiniškais pastangomis“ (Drees 1988: 19). UNIX specialistas yra pernelyg mandagus, kad į šią sumaištį įpintų firmos pavadinimą, pvz., *Microsoft korporacija*.

loci, sukūręs pirmąsias, elegantiškiausias operacines sistemas, kad *Microsoft* korporacijos valdiny susiaurino jų pavyzdžius iki bukiausių [operacinių sistemų].

Norint apdoroti tekstus – tai yra, tapti popierine mašina [kompiuteryje] IBM AT, valdomame *Microsoft* DOS – pirmiausia reikia nusipirkti komercinį programinės įrangos paketą. Antra, kai kurie paketo failai turi turėti .EXE ir .COM plėtinius, kitaip tekstų apdorojimo nebus galima pradėti. Vykdomieji failai, ir tik jie, turi ypatingą ryšį su savo vardais. Viena vertus, jų vardai yra visiškai savireferenciniai, pavyzdžiui, *WordPerfect*, antra vertus, jie yra daugiau ar mažiau kriptiniai akronimai (nes neturi balsių), pavyzdžiui, *WP*. Pilni vardai pasitarnauja tik programinės įrangos gamintojų reklaminėms strategijoms, pastarosios vis dar naudoja kasdienę kalbą; o antroji galimybė reikalinga todėl, kad diskų operacinė sistema, dar žinoma kaip DOS, neperskaito ilgesnių nei aštuonių raidžių failų vardų. Štai todėl neištariamoms balsių neturinčios žodžių santrumpos – akronimai atsisakantys tos paprastos graikų inovacijos – yra ne tik būtini postmoderniajam rašymui, bet ir tobulai atitinka jo uždavinius. Dar daugiau, atrodo, kad pirmą kartą nuo alfabeto išradimo, šios santrumpos suteikė jam magiškų galių. Santrumpa *WP* daro tiksliai tai, ką ji sako. Skirtingai nei signifikantas *WordProcessor* ar tušti senieji europietiški vardai, tokie kaip „Dvasia“ ar netgi „Žodis“, vykdomieji kompiuterio failai apima visas programas ir duomenis, kurie būtini jų realizacijai. Rašymo aktas – W, P ir *Enter* paspaudimas – nepadaro žodžio tobulu, bet jis paleidžia *WordPerfect*. Štai kokių pergalių pasiekia programinė įranga.

Daugiau ar mažiau perteklinė popierinė įranga (*Paperware*), lydinti programinę įrangą, kad neatsiliktų nuo komandos eilutės, stebuklingas galias dar padvigubina. Kadangi programinės įrangos vadovėliai privalo peržengti prarają, žiojinčią tarp kasdienės kalbos, elektronikos ir literatūros, jie paprastai pristato savo programų paketus kaip lingvistinius veikėjus, kurių visagalybė tam tikro kompiuterio sistemos resursų, adreso laukų ir aparatinės įrangos parametrų atžvilgiu pasirodo absoliučiai būtina: *WP*, aktyvuotas komandos eilutės argumento X, perjungia ekraną iš būsenos A į B, pradeda taikyti C, galiausiai grįžta į D, ir t. t.⁷

Visgi, visi veiksmai, kuriuos pagal tai, kas įrašyta popierinėje įrangoje, atlieka veikėjas *WP*, yra visiškai virtualūs, kadangi kiekvienas atskiras veiksmas, kaip taikliai sakoma, vyksta „po“ (*unter*) DOS. Bet faktiškai veikia vien tik operacinė siste-

⁷ Nenuostabu, kad vienintelis man žinomas kontrapavyzdys yra Richardo Stallmano *Free Software Foundation*, paskelbusi didvyrišką, nors lygiai tiek pat beviltišką karą programinės įrangos autorinėms teisėms. Šis kontrapavyzdys skamba taip: „Kai sakome, kad „C-n leidžiasi vertikalčiai žemyn per vieną eilutę“, mes skelbiame skirtumą, kuris yra visiškai bereikšmis kasdienio vartojimo požiūriu, tačiau labai svarbus, norint suprasti, kaip tinkinti *Emac* tekstinius redaktorių. Taip suprogramuojama, kad funkcija *next-line* reikš judėjimą vertikalčiai žemyn. C-n veikia taip, kadangi yra susaistytas su šia funkcija. Persaisčius C-n su funkcija *forward-word*, C-n judės pirmyn per žodį“ (Stallman 1988: 19).

ma, ar tiksliau jos apvalkalas: COMMAND.COM ieško klaviatūros buferyje 8 baitų failo vardo, išverčia santykinį surastų duomenų failo adresą į absoliutų, užkrauna šią modifikuotą versiją iš išorinės masinės atminties į silicio RAM ir galiausiai paskiria programos vykdymą (ribotą laiką) kodo pirmosioms eilutėms, priklausančioms vergui vardu *WordPerfect*.

Tas pats komandos eilutės argumentas taip pat gali būti panaudotas prieš DOS, nes galiausiai paaiškėja, kad operacinė sistema veikia kaip pamatinės įvesties / išvesties sistemos, vadinamos BIOS, plėtotė. Jokia atskira taikomoji programa, net pamatinė mikroprocesorių sistema, negalėtų pradėti veikti, jei keletas elementarių funkcijų, saugumo sumetimais įdegtų silicyje ir taip virtusių neišdildoma programinės įrangos dalimi, neturėtų, taip sakant, Barono von Münchhauseno savybių ištraukti savęs pačių už plaukų⁸. Bet kokia materiali entropijos transformacija į informaciją, iš milijonų snaudžiančių tranzistorių į elektros įtampos skirtumus, būtinai numato materialų įvykį, vadinamą atkūrimu (*Reset*).

Iš principo šis nužengimas iš programinės į aparatinę įrangą, iš aukštesnio į žemesnį stebėjimo lygmenį, galėjo tęstis per tiek eilių, kiek panorėjus. Net elementarios kodų operacijos, nepaisant jų metaforiškų pažadų, pvz., kreiptis (*Call*) ar grįžti (*Return*), apsiriboja griežtai vietine manipuliacija ženklais, tai reiškia, (kaip gaila, Lacanai), elektrinio potencialo signifikantais. Bet kokia formalizacija, kaip ją apibrėžė Hilbertas, – efektyviai atsisako teorijos, paprasčiausiai todėl, kad teorija, su kuria čia susiduriame, „nebėra reikšmingų teiginių sistema, tačiau ją sudaro sakiniai kaip žodžių sekos, kurie savo ruožtu yra raidžių sekos. Vien per formą mes nustatome, kurios žodžių kombinacijos yra sakiniai, kurie sakiniai yra aksiomos, ir kurie sakiniai yra betarpiški kitų sakinių sekmenys.“⁹

Kai reikšmė susitraukia iki sakinių, sakiniai iki žodžių, o žodžiai – iki raidžių, tuomet neegzistuoja ir jokia programinė įranga. Arba dar daugiau: ji nebūtų egzistavusi, jei kompiuterinėms sistemoms nebūtų reikėję, bent jau iki dabar, koegzistuoti su kasdienių kalbų aplinka. Ši aplinka savo ruožtu buvo sudaryta, nuo garsiojo dvigubo graikų išradimo laikų, iš rašto ženklų ir monetų¹⁰, *letters and litters* (raidžių ir šiukšlių). Tuo metu ekonominių priežasčių spaudimas iš esmės įveikė kuklųjį Alaną Turingą, kuris technologinės eros akmenis amžiuje mieliau skaitė binarinę nei dešimtainę mašinos išvestį¹¹. Vadinamosios kompiuterininkų bendruomenės vadinamoji filosofija, priešingai, daro viską, ką gali, kad paslėptų aparatinę įrangą po programine, elektroninius signifikantus po žmogaus-mašinos sąsaja. Pro-

⁸ Toks galėtų būti termino *operacinės sistemos paleidimas* (*Booting*) laisvas vertimas.

⁹ Stephen C. Kleene, cituojamas Rosen 1988: 527.

¹⁰ Plg. Lohmann 1980: 174.

¹¹ Plg. Hodges 1983: 399.

gramavimo kalbų vadovėliai filantropiškai įspėja apie psichines negandas, kurios ištiks įrašius trigonometrines funkcijas į assemblerio kodus¹². BIOS procedūros (ir jų profesionalūs autoriai) kilniaširdiškai perima funkciją „paslėpti detales, kontroliuojančias aparatinę įrangą, jūsų programose“ (Barkalati 1989: 528). Iš to plaukianti loginė išvada nelabai skiriasi nuo viduramžiškų angelų hierarchijų – operacinės sistemos funkcijos, tokios kaip COMMAND.COM, visiškai paslėps BIOS, taikomosios programos, tokios kaip *WordPerfect*, visiškai paslėps operacines sistemas; ir taip toliau. Dabartiniai fundamentalūs kompiuterio dizaino pokyčiai (t. y. pagal tai, kaip Pentagonas supranta mokslą) išstobulino šią slėpimo sistemą.

Pirmiausia grafinės sąsajos buvo sukurtos naudojimuisi sąmoningai paviršutiniškame lygmenyje, kuris, paslėpdamas programavimui reikalingas rašymo operacijas, neprileisdavo vartotojų prie mašinos kaip visumos. IBM autorizuo- tas kompiuterinės grafikos kompendiumas net nebando apsimesti, kad jo vartotojų sąsajos padaro programavimą spartesnę ar efektyvesnę nei paprastos komandų eilutės¹³. Antra, kartu su Pentagono programavimo kalba ADA¹⁴, tačiau mikroskopi- niame pačios aparatinės įrangos lygmenyje, buvo sukurta nauja veikseną, vadinama „apsaugotas režimas“. Pasak Intelio *Microprocessor Programming Manual*, ji turi vienintelį tikslą – neleisti „nepatikimoms programoms“ ir „nepatikimiems vartoto- jams“ pasiekti sistemos resursų, tokių kaip įvesties / išvesties kanalai, ir operacinės sistemos branduolio. Tačiau techniniu požiūriu visi vartotojai yra nepatikimi, ap- saugotame režime (vyraujančiame UNIX operacinėje sistemoje) jiems neleidžiama kontroliuoti savo mašinų.

Nesulaikomas pergalingas programinės įrangos žygis keistai apverčia Turingo įrodymą, kad nėra jokių, matematine prasme, neapskaičiuojamų užduočių, kurių negalėtų išspręsti paprasčiausia mašina. Būtent tos mašinos buvimo vietoje fizinė Churcho-Turingo hipotezė, prilygindama fizinę aparatinę įrangą algoritmams, skir-

¹² Plg. *TOOL Praxis: Assembler Programmierung auf dem PC*. 1 leidimas, 1989. Würzburg: Vogel, p. 9.

¹³ Plg. Foley, Van Dam, Feiner, Hughes 1990: 397 f.: „tiesioginis [grafinis] manipuliavimas kartais pateikiamas kaip geriausia vartotojo sąsaja. Žinoma, ji veiksminga ir ja lengva išmokyti naudotis. Tačiau *Macintosh* sąsaja gali būti per lėta patyrusiems vartotojams, tuo tarpu tiesioginė komanda veikia greičiau. Norint rasti ir atspausdinti failą „Chapter 9“, reikia rasti jo piktogramą, tada ją pasirinkti ir pritaikyti komandą *spausdinti*. Piktogramos ieškant gali tekti slinkti per daugybę elementų. Jei vartotojas žino failo vardą, greičiau bus parašyti komandą „Print Chapter 9“. Analogiškai, norint ištrinti visus „.txt“ tipo failus, reikia visus juos surasti bei nutempti kiekvieną į šiukšliadėžę. Daug greičiau tai galima padaryti panaudojus UNIX komandą „rm *.txt“, kuri naudoja pakaitos simbolių * surasti visiems failams, kurie baigiasi „.txt“. Iš viso to galiausiai išplaukia gniuždantis pačių kompiuterių paskirties nuvertinimas: „kai kurios veiklos, pvz., programavimas, negali būti atliekamas tiesiogiai“.

¹⁴ Apie ryšį tarp Pentagono, ADA ir *Intel iAPX 432*, pirmojo mikroprocesoriaus, turinčio apsaugotą režimą, kurio ekonominė nesėkmė suteikė pagrindą industriniam standartui nuo 80286 iki 80486, plg. Myren 1982: 335–344 (už šią nuorodą dėkoju Ingo Ruhmannui iš Bonos). Kas norėtų suprasti tą nesėkmę, turėtų pamedituoti štai kokią sakinį: „432 galima apibūdinti kaip trijų adresų iš saugyklos į saugyklą (three-address-storage-to-storage) architektūrą, kurioje programoms matomų registrų nėra“ (Myren 1982: 342).

tiems jai apskaičiuoti, sukūrė tuštumą, kurią sėkmingai užimti galėjo programinė įranga ir kurios tamsumas jai [programinei įrangai] yra naudingas.

Galiausiai, kuo aukščiau kyla Babelio bokštas ir kuo kasdieniškesnės tampa programavimo kalbos, tuo labiau jos veikia taip pat, kaip vadinamosios vienusės funkcijos matematinėje kriptografijoje¹⁵. Savo standartine forma tokios funkcijos gali būti apskaičiuotos su pagrįstomis laiko sąnaudomis, pavyzdžiui, kai operacijų trukmė didėja tik daugianarėms funkcinio kompleksiško išraiškoms. Antra vertus, laiko sąnaudos atvirkštinei operacijai, tai yra, įvesties parametrų apskaičiavimui funkcijos rezultatų pagrindu, išaugtų eksponentiškai, vadinasi, nebevaldomai, palyginus su funkcijos kompleksišku. Kitaip tariant, vienusės funkcijos saugo algoritmus nuo jų pačių rezultatų.

Ši kriptografinė funkcija tarsi sukurta programinei įrangai (o taip ir yra). Ji siūlo patogų būdą išvengti to, ką rodo Turingo įrodymas: kad intelektinės nuosavybės sąvoka tapo nebeįmanoma – ypač ten, kur omenyje turimi algoritmai. Tačiau pats faktas, kad programinė įranga neegzistuoja atskirai nuo mašinų, tik padidino komercinius (arba amerikietiškus) reikalavimus medijai. Visos licencijos, aparatiniai saugumo raktai ar patentai, kurie buvo įregistruoti *WP*, arba *WordPerfect*, įrodė vienusių funkcijų funkcionalumą. Amerikos teismai, paniekindami visas matematinės garbės tradicijas, patvirtino net algoritmų autorines teises.

Taigi, nestebina, kad pastaruojų metu aukščiausiam lygyje, tai yra [kompagnijoje] IBM, prasidėjo medžioklė matematinių lygčių, kurios galėtų įvertinti algoritmų originalumą, remdamosi Shannon informacijos kiekiu ir Kolmogorovo informacijos turiniu. Senais gerais Shannon informacijos teorijos laikais maksimali informacija buvo proporcinga minimaliam triukšmo lygiui, o IBM kaltina Shannoną, kad net kauliukų metimo serija, jo manymu, matematiniu požiūriu nusipelnytų premijos. Tačiau jų netenkina ir Kolmogorovo pasiūlymas, premijuojantis trumpiausią galimą algoritmą išvesties generavimui. Pasak Kolmogorovo, visi trigonometrinių arba astronominių lentelių skaičiavimo sunkumai išnyktų lygtyse, kuriomis jie pagrįsti. Taigi naujasis IBM loginio gylio matas, supainiojęs sėdmenis ir atsitiktinumą, apibrėžiamas taip:

Pranešimo vertė [...] pasirodo glūdinti ne jo informacijoje (jo visiškai nenuspėjamos dalyse), ne jo perteklume (pažodiniuose pakartojimuose, nelygiuose skaitiniuose dažniuose), bet tame, ką galima būtų pavadinti jo palaidotu pertekliumu (*begrabene Redundanz*) – dalyse, kurios gali būti sunkiai numatomos, dalykuose, kuriuos gavėjas iš principo galėtų suvokti, nors jie jam nepasakomi, tačiau tik pasitelkęs nemenkas pinigų, laiko ar skaičiavimo sąnaudas. Kitaip tariant, pranešimo vertė proporcinga matema-

¹⁵ Apie tai žr. Horster 1982/1985: 23–27.

tiniam ar kitam darbui, kurį atliko jo iniciatorius, ir kurio gavėjas nebeturi pakartoti (Bennett 1988: 230).

IBM gylio matavimas matematiniu tikslumu galėtų taip pat pakeisti pasenusias ir visiškai netikslias kasdienes sąvokas – „originalumas“, „autorystė“ ir „autorių teisės“ – bei paversti jas teisiškai įpareigojančiomis. Deja, algoritmų originalumo apskaičiavimo algoritmas apskritai yra neapskaičiuojamas netgi Turingo metodais¹⁶.

Šioje tragiškoje situacijoje baudžiamoji teisė, bent jau Vokietijoje, atsisakė intelektualinės nuosavybės sąvokos programinei įrangai, nematerialiai kaip ir pati teisė, ir vietoj to apibrėžia programinę įrangą kaip „dalyką“ (*Sache*). Vokietijos aukščiausiojo teismo sprendimas, pasak kurio, jokia kompiuterinė programa neveiks nesant atitinkamos elektros srovės silikono grandinėje¹⁷, darsyk įrodo, kad virtualusis neapsisprendimas tarp programinės ir aparatinės įrangos, vargu, ar remiasi, kaip nuoširdžiai tiki sistemų teoretikai, stebėtojo perspektyvos pokyčiu¹⁸. Priešingai, nesunku rasti pagrįstų argumentų už aparatinės įrangos būtinumą, taigi ir pirmenybę.

Mašina su neribotais laiko ir erdvės ištekliais, nenutrūkstamu popieriaus tiekimu ir neribotu skaičiavimo greičiu egzistavo tik vienintelį kartą: Turingo straipsnyje „*On Computable Numbers with an Application to the Entscheidungsproblem*“ („Apie apskaičiuojamus skaičius, taikant juos sprendimo problemai“). Visos fiziškai pagaminamos mašinos, priešingai, yra griežtai apribotos parametrų, nustatytų jų pačių kodo. *Microsoft* DOS negebėjimas atpažinti pavadinimų, ilgesnių nei aštuoni simboliai (pvz., *WordPerfect*), trivialiu ir senstelėjusiu būdu iliustruoja problemą, dėl kurios atsiranda vis daugiau nesuderinamumo tarp skirtingų kartų 8 bitų, 16 bitų ir 32 bitų mikroprocesorių. Jis taip pat nurodo į principinę digitalizacijos negalimybę – tai yra, [negalimybę] apskaičiuoti realiųjų skaičių kūną, kuris anksčiau buvo vadinamas Gamta¹⁹.

Tačiau tai reiškia, pasak Los Alamos nacionalinės laboratorijos, kad

¹⁶ Už šį pastebėjimą dėkoju Oswaldui Wieneriui iš Doston Sičio.

¹⁷ Plg. König 1991: 73.

¹⁸ Veikiausiai galima, kaip 1991 m. balandžio 15 d. laiške rašo Dirkas Baeckeris, „manyti, kad skirtis tarp aparatinės ir programinės įrangos lydi skirties tarp programuojamumo ir neprogramuojamumo sugrįžimą į programavimo erdvę. Ji žymi, taip sakant, technologijos skaičiavimo galimybes, taigi, tam tikra prasme ir pačią technologiją. Tai įmanoma tik todėl, kad programos „vienybė“ pasiekiamą, jei lygtys ir skaičiavimai, atitinkamai, yra pasiskirstę abiejose pusėse, ir vienoje pusėje nuolat galima atlikti veiksmus, o kita išlieka nekintanti.“

¹⁹ Todėl aš nesuprantu, kaip žymiojo Turingo straipsnio pirmame sakinyje paskelbus, kad „apskaičiuojamus (*berechenbare*) skaičius“ „galima trumpai apibūdinti kaip tuos realiuosius skaičius“, „kurių dešimtainės išraiškos gali būti apskaičiuojamos (*errechnet*) baigtiniais dydžiais“ (Turing 1987 (1937): 19), apskaičiuojamų (*berechenbare*) skaičių aibę apibrėžti kaip skaičiuojamus (*abzählbar*), ir galiausiai π – kaip „apskaičiuojamos (*berechenbar*) konvergentinės sekos ribinę reikšmę“ – pavadinti apskaičiuojamu skaičiumi (Turing 1987 (1937): 49f.).

mes naudojame skaitmeninius kompiuterius, kurių architektūrą mums pateikia fizinės mašinos su visais savo dirbtiniais suvaržymais. Turime redukuoti nenutrūkstamą algoritminį aprašą iki vieno koduojamo įrenginio, kurio pagrindinės operacijos yra apskaičiuojamos, ir tai darome įvairiomis suskaidymo į dalis formomis, paprastai vadinamomis diskretizavimu. Kompiliatorius galiausiai toliau redukuoja šį modelį iki binarinės formos, kurią daugiausiai lemia mechaniniai apribojimai.

Rezultatas yra diskretiškas ir sintetinis pirminės problemos mikropasaulio vaizdas, kurio struktūra atsitiktinai fiksuojama diferencijavimo schemos ir atsitiktinai parinktos skaičiavimo architektūros. Vienintelis kontinuumo likutis yra šaknies aritmetikos naudojimas, kuris nustato nevienodą bitų svorį, o netiesinėse sistemose tai yra netikrų singularumų šaltinis.

Būtent tai mes išties darome apskaičiuodami fizinio pasaulio modelį fiziniiais įrenginiais. Tai anaipol nėra idealizuotas ir ramus procesas, kurį paprastai vaizduojamės ginčydamiesi dėl pagrindinių skaičiavimo struktūrų, be to, jis labai nutolęs nuo Turingo mašinos (Hasslacher 1988: 421f).²⁰

Kalba ne apie tolesnį sekimą Churcho-Turingo hipoteze ir „algoritminių santykių skiepijimą fizinio pasaulio elgesiui, nepatvirtintą jokiais duomenimis“ (Hasslacher 1988: 420). Jei pasaulis neatsirado Dievui žaidžiant kauliukais, algoritminis lietaus debesų ar jūros bangų elgesys ne atmeta, bet veikiau priima faktą, kad jų molekulės veikia kaip jų pačių veiklos kompiuteriai. Priešingai, tai tiesiog paties „programuojamumo kainos“ apskaičiavimo reikalas. Šis lemtingas kompiuterių pajėgumas akivaizdžiai nieko bendro neturi su programine įranga, jis priklauso tik nuo to, kiek tam tikra aparatinė įranga gali turėti savyje kažką panašaus į rašymo sistemą. 1937 m., kai Claude'as Shannonas „reikšmingiausiame kada nors parašytame magistro darbe“ (Hagemeyer 1979: 432) pateikė įrodymą, kad paprasčiausios telegrafo relės galėtų įgyvendinti Boolė'io algebrą kaip visumą, buvo sukurta tokia užrašymo sistema (*Aufschreibesystem*). O kai septintojo dešimtmečio pradžioje iš Williamo Shockley'o tranzistoriaus išvesta integruota grandinė viename luste sujungė silicį bei valdomą jo varžą su jo oksidu, kaip beveik tobulu izoliatoriumi, materijos programuojamumas galėjo, kaip pranašavo Turingas, „perimti kontrolę“ (Turing 1959 (1987): 15). Taigi, programinė įranga, jei ji apskritai egzistuoūtų, būtų tiesiog milijardų dolerių verslas, besisukantis aplink vieną pigiausių žemėje elementų. Sujungti luste silicis ir silicio oksidas sukuria beveik tobulą aparatinę įrangą. Viena vertus, milijonai grandinės elementų veikia tomis pačiomis fizinėmis sąlygomis – tai lemtinga visų pirma kritiniam lusto temperatūros parametru, be to, neleidžia eksponentiškai didėti tranzistoriaus varžos nuokrypams. Kita vertus,

²⁰ Kittleris netiksliai cituoja Hasslacherio straipsnį nurodydamas pavadinimą „Algorithms in the World of Bounded Resources“, iš tiesų Hasslacherio straipsnio pavadinimas – „Beyond the Turing Machine“, todėl literatūros sąrašė klaida atitaisyta. Straipsnio „Algorithms in the World of Bounded Resources“ (p. 407–416) autorius yra Yūrius Gurevichius (vert. past.).

šie milijonai perjungimo elementų lieka izoliuoti elektra vienas nuo kito. Tik šis paradoksalus ryšys tarp dviejų fizinių parametrų – šiluminio tęstinumo ir elektrinės diskretizacijos – leidžia integruotoms skaitmeninėms grandinėms būti ne tik baigtiniais automatais, kaip ir daugelis kitų žemėje esančių įrenginių, bet ir priartėti prie Universalios Diskrečiosios Mašinos, į kurią jau senai paniro jos išradėjo vardas – Turingas.

Šį struktūrinį skirtumą galima labai nesunkiai iliustruoti. Pavyzdžiui, „kombinuotas užraktas yra baigtinis automatas, tačiau jis paprastai neskaidomas į bazinį elementaraus tipo komponentų rinkinį, kurį galima perkonfigūruoti, kad būtų imituojama atsitiktinė fizinė sistema. Todėl jis nėra struktūriškai programuojamas, o šiuo atveju jis yra veiksmingai programuojamas tik ribota prasme, kad jo būseną galima būtų nustatyti tam tikro riboto tipo veikimui pasiekti.“ Priešingai, „skaitmeninis kompiuteris, naudojamas imituoti kombinuotą užraktą, yra struktūriškai programuojamas, nes jo veikimas pasiekiamas sintezuojant iš kanoninės primityvių perjungimo komponentų aibės“ (Conrad 1988: 289).

Perjungimo komponentai, nesvarbu, telegrafo relės, elektronų vamzdžiai ar galiausiai silicio tranzistoriai, sumoka kainą už savo skaidomumą arba diskretiškumą. Išskyrus trivialų (t. y. diskretų) tekstų apdorojimo atvejį, kuris nublanksta prieš visas kitas mokslo, karines ir pramonės sritis, kuriose naudojami kompiuteriai, skaitmeniniai skaičiuotuvai, kaip vieninteliai „viskas arba nieko organai“ tikrąja tų žodžių prasme²¹ stoja priešais nuolatinę debesų, bangų ir karų aplinką. Šių milžiniškų, be to, tikrų, skaičių laviną, kaip pasakytų Ianas Hackingas, galima įveikti tik pridėdant vis daugiau perjungimo elementų, kol 2000 *Intel 4004* tranzistorių pavirto dabartinio *Intel* flagmano 80486 [tranzistoriais]. Tačiau galima matematiškai įrodyti, kad galimų jungčių tarp šių elementų augimo tempo, t. y. skaičiavimo pajėgumų kaip tokių, viršutinė riba yra kvadratinės šaknies funkcija. Kitaip tariant, sistema „negali neatsilikti nuo polinominio problemos dydžio augimo“ (Conrad 1988: 293), jau nekalbant apie eksponentinę normą. Ta pati izoliacija tarp skaitmeninių ar diskrečių elementų, užtikrinančių jos gebėjimą funkcionuoti – bent jau ne tropinėmis ar arktinėmis sąlygomis – taip pat riboja galimų jungčių su tam tikro lusto vietine aplinka matmenis. Tačiau visuotinės sąveikos sąlygomis, kurias skaitmeniniai lustai patiria tik šiluminiu požiūriu, jų jungiamumas „pagal dabartinius jėgos dėsnius“ (Conrad 1988: 290) bei vadovaujantis kombinatorine logika, gali pakilti iki viršutinės ribos, kuri lygi visų susijusių elementų skaičiaus kvadratui.

Būtent toks optimalus ryšys, iš kitos, fizinės pusės, išskiria neprogramuojamas sistemas. Dėl savo sąveikų visuotinio tokios sistemos, nesvarbu, ar jos yra bangos,

²¹ Plg. Neumann 1951/1967: 150.

ar būtybės, gali demonstruoti polinomiškus kompleksiško augimo tempus; todėl juos apskaičiuoti galėtų tik mašinos, kurioms pačioms nereikėtų mokėti programuojamumo kainos. Akivaizdu, kad šitoks hipotetinis, bet labai reikalingas mašinų tipas būtų gryna aparatinė įranga: fizinis aparatas, veikiantis bet kokio fizinių įrenginių skaičiaus aplinkoje ir paklūstantis tik tokiems pat išteklių apribojimams, kurie taip pat taikomi ir pastariesiems. Programinės įrangos įprastine prasme, kaip abstrakcijos, kurią galima realizuoti, nebebūtų. Tokios mašinos procedūros, net jei jos ir liktų atviros algoritminiam scenarijui, iš esmės turėtų veikti remiantis materialiu pagrindu, kurio ryšys leistų nuolat perkonfigūruoti jo langelius. Ir nors „pagrindą taip pat galima aprašyti algoritminiais terminais, naudojant simuliaciją“, jo „charakteristika turi tokią didžiulę reikšmę [...] efektyvumui ir taip glaudžiai susijusi su aparatinės įrangos pasirinkimu“ (Conrad 1988: 304), kad jos programavimas turėtų mažai ką bendro su Turingo mašinų aproksimacijomis.

Tokios itin reikalingos ir ne per daug tolimos mašinos, apie kurias diskutuojama jau dabartiniame kompiuterių moksle ir kurias jau priartino lustų pramonė²², gali sugundyti kai kurių stebėtojų akis įžvelgti jose [mašinos] pažįstamą žmogišką veidą, tiek evoliuciškai užmaskuotą, tiek ir ne. Galbūt. Tačiau tuo pat metu mums žinoma silicio aparatinė įranga jau atitinka daugelį reikalavimų, keliamų įtinklinoms, neprogramuojamoms sistemoms. Tarp milijonų tranzistorių elementų nuolat vyksta milijonas kvadratu sąveikų: elektronų difuzija ir kvantinės mechanikos tunelio efektai pasireiškia visame luste. Nors šiuolaikinė gamybos technologija tokias sąveikas vis dar traktuoja kaip sistemos apribojimus, fizinius šalutinius poveikius, trukdžių šaltinius ir pan. Visą šį triukšmą, kurio negalima išvengti, reikia bent jau sumažinti: tokią kainą kompiuterių pramonė turi sumokėti už struktūriškai programuojamas mašinas. Atvirkštinė strategija – triukšmo didinimas – ne tik vestų nuo IBM atgal pas Shannoną; tai taip pat būtų vienintelis kelias į tą realiųjų skaičių kūną, kuris kažkada buvo vadinamas Chaosu.

Can't you understand what I'm tryin' to say (Argi nesupranti, ką bandau pasakyti), dainuojama originalioje dainos *Eve of Destruction* versijoje.

Vertė Rimantas Kmita

Versta iš Friedrich A. Kittler „Es gibt keine Software“, in Hans Ulrich Gumbrecht (Hrsg.) *Writing / Ecriture / Schrift*. München: Wilhelm Fink, 1993, S. 167–178.

Vertimą peržiūrėjo Kęstas Kirtiklis

²² Taip pirmasis integruotas neuroninis tinklas, kilęs iš diskrečiosios *Intel* lustų imperijos (antrąsyk šios įmonės istorijoje, po ganėtinau hibridiško i2920 signalų procesoriaus), sugrįžo prie analoginių stiprintuvų.

Literatūra

- Barkalati, N. 1989. *The Waite Group's Macroassembler Bible*. Indiana/IL.
- Bennett, Ch. H. 1988. "Logical Depth and Physical Complexity", in Herken, Rolf (ed.), *The Universal Turing Machine: a Half-Century Survey*. Oxford University Press, p. 227–257.
- Conrad, M. 1988. "The Prize of Programmability", in Herken, R. (ed.), *The Universal Turing Machine. A Half-Century Survey*. Oxford University Press, p. 285–307.
- Drees, H. 1988. *UNIX. Ein umfassendes Kompendium für Anwender und Systemspezialisten*. Haar bei München: Markt-u.-Technik-Verl.
- Foley, J. D.; Van Dam, A.; Feiner, S. K; Hughes, J. F. 1990. *Computer Graphics. Principles and Practice*. 2nd edition, Reading/MA.: Addison-Wesley Professional.
- Hafki, T. 1993. *Franklin – Frankenstein. Zum Verhältnis von Elektrizität und Literatur. 1750–1816*. Bochum. (Magistro darbas).
- Hagemeyer, F.-W. 1979. *Die Entstehung von Informationskonzepten in der Nachrichtentechnik. Eine Fallstudie zur Theoriebildung in der Technik in Industrie- und Kriegsforschung*. Diss. phil. FU Berlin.
- Hagen, W. 1989. „Die verlorene Schrift. Skizzen zu einer Theorie der Computer“, in Kittler, Friedrich A.; Tholen, Georg Christoph (Hrsg.), *Arsenale der Seele. Literatur- und Medienanalyse seit 1870*. München: Fink, S. 187–202.
- Hasslacher, B. 1988. "Beyond the Turing Machine", in Herken, R. (ed.), *The Universal Turing Machine. A Half-Century Survey*. Oxford University Press, p. 417–433.
- Hodges, A. 1983. *Alan Turing: The Enigma*. New York: Simon and Schuster.
- Horster, P. 1982/1985. *Kryptologie: eine Anwendung der Zahlentheorie und Komplexitätstheorie*. Mannheim–Wien–Zürich: BI-Wissenschaftsverlag, S. 23–27.
- Intel Corporation, 1989. *387 DX User's Manual. Programmer's Reference*. Santa Clara/CA.
- Intel Corporation, 1990. *i486 Microprocessor. Programmer's Reference Manual*. Santa Clara/CA.
- König, M. M. 1991. „Sachlich sehen. Probleme bei der Überlassung von Software“, *c't (Magazin für Computertechnik)*, 3: 70–75.
- Lohmann, J. 1980. „Die Geburt der Tragödie aus dem Geiste der Musik“, *Archiv für Musikwissenschaft*, 37: 167–186.
- Myren Jr., G. 1982. "Overview of the Intel iAPX 432 Microprocessor", *Advances in Computer Architecture*. New York, p. 335–344.
- Neumann, J. von. 1951/1967. „Allgemeine und logische Theorie der Automaten“, in *Hans Magnus Enzensberger (Hrsg.), Kursbuch 8 – Neue Mathematik, Grundlagenforschung, Theorie der Automaten*. Frankfurt am Main: Suhrkamp Verlag, S. 139–175.
- Rosen, R. 1988. "Effective Processes and Natural Law", in Herken, R. *The Universal Turing Machine – A Half-Century Survey*. Hamburg-Berlin: Kammerer & Unverzagt, p. 523–537.
- Schrödl, K. 1990. „Quantensprung“, *DOS*, 12: 102–110.
- Stallman, R. 1988. *GNU Emacs Manual*. 6th edition. Cambridge/MA.
- Turing, A. M. 1937 (1987). „Über berechenbare Zahlen. Mit einer Anwendung auf das Entscheidungsproblem“, in Dotzler, B.; Kittler, F. A. (Hrsg.), *Alan Turing Intelligence Service*. Berlin: Brinkmann und Bose, S. 17–60.
- Turing 1959 (1987). „Intelligente Maschinen. Eine häretische Theorie“, in Dotzler, B.; Kittler, F. A. (Hrsg.): *Alan Turing Intelligence Service*. Berlin: Brinkmann und Bose, S. 6–15.