Asupra Fundamentelor Informaticii

Neculai Andrei
Research Institute for Informatics,
Center for Advanced Modeling and Optimization
8-10, Averescu Avenue, Bucharest 1, Romania
E-mail: nandrei@ici.ro

Rezumat. În această prezentare încercăm să definim fundamentele informaticii. Vom începe cu o definiție a informaticii care după părerea noastră se plasează în sfera intelectualismului de mare altitudine și care constă în coborârea în computațional a conceptelor matematice, algoritmizarea conceptelor matematice, studiul algoritmilor din punctul de vedere al convergenței și al complexității lor.
Continuăm prezentarea cu fundamentele matematicii care ne arată sursa de unde provine problematica informaticii: calculabilitate, complexitate și matematica experimentală. Teoria calculabilității vine din două tradiții, care cu foarte mici excepții s-au dezvoltat în paralel, de cele mai multe ori ignorându-se reciproc. Pe de-o parte avem analiza numerică și calculele științifice, și pe de altă parte avem teoria calculului așa cum apare în logică și știința calculatoarelor. Ambele aceste tradiții utilizează noțiunea de **algoritm**. Ideea este de a construi o mașină (peste corpul numerelor reale) care combină cele două tradiții cu scopul de a înțelege natura și complexitatea calculului, ca fundament al informaticii.
În final discutăm matematica experimentală ca acea secțiune a informaticii care se ocupă cu codificarea și transmiterea, în comunitatea matematică, a rezultatelor prin utilizarea tehnologiilor avansate de calcul pentru a realiza un experiment matematic, cu scopul de a explora structuri și concepte matematice, de a testa conjecturi, credințe, de a sugera generalizări, precum și a analiza rezultatele obținute în urma acestor experimentări.

1. Definiția informaticii

Societățile dezvoltate, puternic industrializate, capabile să asigure un nivel de trai și de muncă ridicat membrilor lor își datorează acest statut nu numai faptului că acestea au înțeles ca în relația cu cetățeanul să aplice vechile concepte democratice grecești actualizate la nevoile timpului, ci au înțeles să implementeze cu perseverență și agresivitate dezvoltările științifice și tehnologice. Aceasta este esența societăților dezvoltate cultural și industrial, aplicarea științei și a tehnologiei în toate activitățile umane. Dar, dezvoltările științifice și tehnologice din orice domeniu de activitate sunt de sorginte matematică. Așa încât, matematica întotdeauna a avut o poziție specială în sistemul de gândire al umanității, dând sens, substanță și valoare acțiunilor acesteia.

La început, din timpurile străvechi până încoace, omul a fost sub **primatul existenței**. Descrierea sistemelor fizice sau artificiale era mai mult lingvistică, descriptivă, deseori nu așa cum acestea se prezentau, ci mai mult cum păreau ele să fie. Chiar matematicile în timpurile vechi, așa cum au fost sintetizate de Euclid (325-265 îChr.) sau de Diofantus din Alexandria (200-284 îChr.), erau exprimate sub formă de silogisme. Aproape 1500 de ani matematicile au rămas sub această formă silogistică. Lumea era văzută ca un Verb, ca o Zicere. Vechii filosofi, care băteau pietroasele pustii ale pământului predicând o filozofie a raționalismului, aveau credința că lumea în infinita sa varietate se supune unor legi, iar omul trebuie să trăiască sub autoritatea celor șapte inscripții Delphice: „*Tu ești*”, „*Zeului onoarea*”, „*Ascultă legilor*”, „*Cruță timpul*”, „*Cunoaște-te pe tine însuți*”, „*Nimic prea mult*”, „*Chezășia poartă nenorocire*” [Dumitriu, 1974, p. 145].

Atât oamenii de știință cât și filosofii, în general, fac o distincție clară între matematici și științele Naturii - științele inginerești, economie, biologie etc. În timp ce științele Naturii sunt construite pe baza invocării argumentului potrivirii experimentului cu

realitatea conform formulei „*adaequatio rei et intellectus*[1] “, matematicile permit o mult mai mare flexibilitate în lumea abstractă a minții. Această caracteristică care a definit matematicile de milenii a început să se schimbe odată cu apariția calculatoarelor. Acestea au schimbat peisajul intelectual oferind posibilitatea de a explora lumi cu totul şi cu totul noi, inimaginabile, care altfel fără utilizarea unei „*puteri de calcul*“ rămâneau inaccesibile minții noastre. Prețul plătit pentru accesul la aceste lumi, fiindcă întotdeauna trebuie să fie un preț, este că multe dintre acestea pot fi cunoscute numai pe cale experimentală.

Calculatoarele de astăzi promit accesul la lumi stranii, care deocamdată rămân doar performanțe intelectuale izolate. Într-adevăr, observăm o proliferare a tehnicilor euristice, a rezultatelor nedemonstrate, a conjecturilor, a opiniilor etc. Calculatoarele au oferit şi oferă din ce în ce mai multe resurse computaționale şi conceptuale pentru a explica câteva dintre aceste fenomene, dar acestea în esența lor au rămas simple curiozități izolate fără a fi sistematic studiate şi explicate.

Prima încercare de a introduce calculul matematic a fost făcută de Fibonacci (1170-1250) în cartea sa *Liber abaci*. Această carte extraordinară trata numerele arabe şi arăta cum se lucrează cu ele. Prezenta studii avansate de algebră referitoare la rezolvarea sistematică, prin aproximații, a ecuațiilor algebrice, precum şi diferite sisteme de numerație nezecimale, ca sistemul binar. Lucrarea conținea probleme privind comerțul şi calculul cursurilor valutare. Un capitol special era dedicat şirului numerelor Fibonacci. Apariția acestei lucrări a marcat un moment foarte important în dezvoltarea ştiinței. Cu toată importanța şi valoarea ei, lucrarea lui Fibonacci a trecut neobservată, lumea nefiind conştientă de puterea reprezentării matematice şi a calculului asociat acesteia. Abia după trei sute de ani s-a înțeles semnificația şi adevărata putere a calculului. René Descartes (1596-1650) a introdus *conceptul de metodă*, punând la bază *noțiunea de îndoială*. Galileo Galilei (1564-1642), unul dintre cei mai mari scriitori ai Italiei renascentiste, a fundamentat *experimentele fizice sistematice*, iar Sir Isaac Newton (1642-1727) şi Baron Gottfried Leibniz (1646-1716) au introdus *calculul diferențial și integral* realizând ceea ce se poate numi prima metamorfoză a ştiinței prin transformarea fundamentelor operaționale ale ştiinței, moştenire a lui Aristotel, în ceea ce se cunoaşte a fi ştiința modernă de astăzi. Această nouă atitudine constă în utilizarea *experimentelor fizice* şi a *modelelor matematice* exprimate ca ecuații diferențiale. Acest primat al matematicii este atât de activ încât matematicienii şi oamenii de ştiință din zilele noastre continuă să se refere la reprezentarea matematică a creației, evitând aproape complet să coboare în existență. Interesul principal este de a rezolva modelele matematice, de a găsi proprietățile soluțiilor acestora. Este important de remarcat faptul că toată construcția noastră intelectuală bazată pe modele matematice sub influența lui Newton şi Leibniz este de *natură localistă*. Altfel spus, modelele matematice, exprimate sub formă de ecuații diferențiale, reprezintă *comportarea locală*, în jurul unui punct, a porțiunii de creație modelate.

De-a lungul timpului am spus că **a face ştiință înseamnă a face matematică**, şi atât de mult s-a încetățenit această opinie încât în anumite domenii asistăm la un adevărat **primat al matematicii** asupra ştiințelor naturii. Această atitudine adoptată din ce în ce de mai mulți oameni de ştiință se datorează, în principal, superficialității şi suficienței în care a căzut filosofia secolului XX. Într-adevăr, filosofii contemporani practică o *filosofie garrula*, o filosofie flecară, vorbăreață, renunțând la conținutul fundamental al acesteia, aşa cum a fost el definit de la începuturi ca „*iubire de înțelepciune*“, „*iubire a Naturii*“.

Această atitudine de accentuare numai a formalismului matematic şi de abandonare a problemei descoperirii mecanismelor cauzale fundamentale care guvernează mişcarea în porțiunea de creație în care suntem interesați îşi are originea în ideile lui Pitagora. El a fost cel care a conştientizat ubicuitatea relațiilor matematice în Natură, privind matematica drept

[1] Sensul exact al cuvântului „*adaequatio*“ este „*potrivire*“. Să observăm că formula, lansată de scolastică, reprezintă totuşi o viziune simplistă şi limitată. Adică adecvarea, acordul dintre intelect şi real, când te referi la real, poate să limiteze realul. Cu alte cuvinte, este vorba de o identificare a omului cu realul constatat pe calea rațiunii sau sensibilă, altfel spus, cu realul pe care l-am „înțeles“ noi.

esența unui spațiu mistic supranatural care guvernează realitatea înconjurătoare. Mai târziu, Platon a fost cel care a adoptat misticismul Pitagoreic, soluția sa la problema conceptelor fiind construcția unei lumi supranaturale - *lumea Formelor* - prezentată în dialogul *Timaios*, în care se regăsesc abstracțiile matematice perfecte și eterne la care se referă conceptele noastre. Metafizic, lumea Formelor era privită ca reală, în timp ce entitățile fizice din lumea noastră nu erau nimic altceva decât umbre, copii imperfecte ale Formelor. Cunoașterea adevărată însemna deci cunoașterea Formelor imateriale și nu a lumii reale imperfecte, aflată într-o continuă schimbare. Ca atare, în științele Naturii cunoașterea adevărată este cunoașterea Formelor abstracte, adică a legilor matematicii din domeniul respectiv și nu a cauzelor fundamentale, proprii domeniului. Platonismul, care se manifestă cu putere și în zilele noastre, evită lumea fizică în favoarea unei lumi supranaturale a abstracțiilor pure în care există o ordine bine definită.

A doua versiune majoră a primatului matematicii este datorată acelor gânditori care au acceptat ideile platonice ale unei lumi materiale inerent pasive și haotice, dar care au rejectat existența unei lumi supranaturale, care impune ordine. Dar această poziție implică o lume care rămâne haotică - o reprezentare neacceptabilă, deoarece faptul că lumea are o ordine este evident și pentru cel mai sceptic dintre sceptici. Dilema *cum poate știința să explice ordinea, rânduiala, lumii fizice dacă ea rejectează noțiunea de ordine supranaturală* a fost soluționată de Kant, filosoful divin cu cea mai mare influență asupra oamenilor de știință din secolul XX. Soluția lui este: „*Noi suntem cei care facem ordine. Cunoașterea perceptuală nu este o achiziție directă a lumii înconjurătoare, așa cum este ea, ci o distorsiune a ei - ea constă în datele senzoriale care au fost procesate de conștiința noastră conceptuală, date adaptate pentru a se potrivi unor anumite categorii conceptuale înnăscute ce pun ordine în senzațiile noastre*“. Acest proces, pe care nu-l putem evita orice am face, se reafirmă prin faptul că noi nu percepem realitatea așa cum aceasta este în mod real, ci mai degrabă așa cum ne apare nouă, după ce am procesat-o.

Sub influența lui Kant oamenii de știință au renunțat la ideea de a înțelege lumea reală în profunzimile ei cauzale, ca fiind o idee fără sens, deoarece tot rămâne un rest care nici o dată nu va fi cunoscut - „*nunquam omnem veritatem sciemus*” - niciodată nu vom ști tot adevărul. Rămâne deci că modelele matematice descriu „*în mod corect*“ numai aparențele, adică înțelegerea noastră a lumii fizice în care trăim. Altfel spus, modelele matematice se găsesc mereu *în perspectiva infinitei asemănări* cu realitatea.

Observăm imediat premisele împărtășite de versiunea Platonistă și de cea Kantiană a primatului matematicii. Ambele privesc lumea materială a entităților ca nefiind reală, așa încât **ceva** din afara lumii materiale trebuie să răspundă de ordinea din lume. Platoniștii consideră că acest ceva aparține domeniului supranatural, în timp ce Kantienii cred că acesta se află în categoriile conceptuale ale minții noastre. Ambele concepții consideră că acest ceva se află în conștiință. Deci *primatul matematicii se reduce la primatul conștiinței* - idee conform căreia conștiința impune *identitate și structură* asupra unei lumi care altfel ar fi haotică[2] . Cele două concepte, totuși, nu cad de acord asupra purtătorului acestei conștiințe, care pune ordine în lume - Dumnezeu sau Omul.

Ca atare, vedem că în studiul științelor Naturii renunțarea la adresarea directă la lumea fizică reală în favoarea formalismului matematic vine direct din filosofie, sub autoritatea a doi dintre cei mai talentați și inspirați filosofi: Platon și Kant. Dar, odată cu apariția și dezvoltarea calculatoarelor s-a observat că și această lume ideală a Formelor Matematice are fisuri, are slăbiciuni în chiar natura ei.

Coborârea în computațional a modelelor matematice a determinat astfel *întoarcerea* la adresarea directă la lumea reală, în sensul potrivirii critice a rezultatelor experimentelor

[2] Este important să menționam aici concepția lui Roger Sperry, laureat în 1981, al Premiului Nobel pentru psihologie și medicină, care spune ca *dacă până acum câteva decenii, motivat de materialism și behaviourism, se considera că gândirea este produsul creierului, astazi trebuie să mărturisim că fenomenul interior, al conștiinței, nu este produsul, ci cauza creierului.*

computaționale cu observațiile. *Experimentele computaționale au reinstituit deci primatul existenței,* reafirmând concepția Galileană a identificării numai a concretului existent, identificare făcută în cadrul şi prin intermediul Formelor Matematice. Altfel spus, cauzalitatea este un fapt independent de conştiință - a noastră sau a lui Dumnezeu. Ordinea, legitatea, regularitatea nu derivă dintr-o conştiință cosmică şi nici dintr-o formă subiectivă de gândire, care se întâmplă să genereze mintea umană. *Cauzalitatea este o lege inerentă a lui a fi. A fi înseamnă a fi ceva, iar a fi ceva înseamnă a acționa conform naturii lucrului.*

Disciplina care realizeaza aceasta *reîntoarcere la lumea fizică reală a sensurilor* este ***informatica***. În acest sens, informatica nu se instituie sub nici o formă ca studiul substratului fizic al informației, al telecomunicațiilor şi teoriei sistemelor, sau teoria informației, al automaticii sau al aplicatiilor specifice din diverse domenii, aşa dupa cum s-a afirmat, ci este un ***mod concret de realizare a experimentelor computaționale, de coborâre în computațional a conceptelor matematice, de algoritmizare a acestor concepte, de studiul complexității acestor concepte***. Privită în profunzimi, ***informatica este algebră computațională,*** care în esența ei constă în ***transformarea unui concept într-un algoritm și apoi transformarea acestuia într-un program de calcul***. Evident că definirea conceptului matematic, elaborarea algoritmului şi elaborarea programului care „*întrupează*“ algoritmul sunt activități total diferite, care utilizează concepte şi principii diferite. Propriu informaticii este ***elaborarea algoritmului, studiul convergenței și a complexității acestuia, precum și transformarea lui într-un program de calcul operațional pe calculatoare***. Aceasta este substanța adevarată a informaticii, conținutul ei, care o plasează chiar în cadrul matematicii.
În acest context trebuie să clarificăm lucrurile şi să precizăm că:


Grigore Moisil
(1906-1973)

Adevaratul creator al informaticii în țara noastră, la nivelul conceptelor şi al fundamentelor, fără nici o îndoială, a fost Profesorul Moisil. Contribuțiile sale au fost în mai toate domeniile matematice, ilustrându-se cu precădere în logica matematică şi ştiința calculatoarelor. Profesorul Moisil instituie o concepție românească asupra teoriei automatelor, conform careia punctul de plecare al acesteia este algebra abstractă şi logica matematică. În anii 1950 Prof. Moisil a dezvoltat o nouă teorie a automatelor finite şi a propus ceea ce se cunoaşte astăzi a fi „algebra Lukaszewicz trivalentă“ pe care a aplicat-o la logica circuitelor de comutație, o contribuție importantă la dezvoltarea ştiinței calculatoarelor. Profesorul Moisil este singurul om de ştiință român care a primit, în 1996, adică la 23 de ani după plecarea lui dintre noi, premiul "*Computer Pioneer*" al IEEE Computer Society.

Prezentarea de mai sus reprezintă o analiză a *profunzimilor lumii spirituale*, a posibilităților noastre de înțelegere a lumii în care trăim. Nu cred că putem vorbi de un primat al matematicii sau de un primat al existenței. Acestea se instituie ca şi concepte de analiză şi de pătrundere în misterele lumilor spirituale şi materiale care caracterizează existența noastră. Ambele aceste concepte sunt importante definind ființa umană deodată ca *persoană* şi *libertate. Persoană în sensul cuplării la existență și făcând parte din existență, și libertate în sensul posibilității infinite de reprezentare a existenței în Forme Matematice.*

Adevăratul om de ştiință, pentru care intelectualismul, ca doctrină prin care cunoaşterea - în întregime sau parțial - este obținută prin rațiune pură reprezintă un mod de existență, întotdeauna *pendulează* între primatul matematicii şi cel al existenței. Trecerea de la o concepție la alta este dată de experimentele computaționale. Acestea „*aduc la realitate*“ dezvoltările matematice şi în acelaşi timp definesc noi „*proprietăți*“ ale existentului.

Am văzut că informatica se instituie ca o ramură a matematicii care *finalizează gândirea matematică în sensul precizării conținutului conceptelor matematicii din punct de vedere computațional*. Ca atare, pentru a ajunge la fundamentele informaticii să prezentăm fundamentele matematicii şi să vedem cum acestea au generat problematica informaticii.

2. Fundamentele matematicii

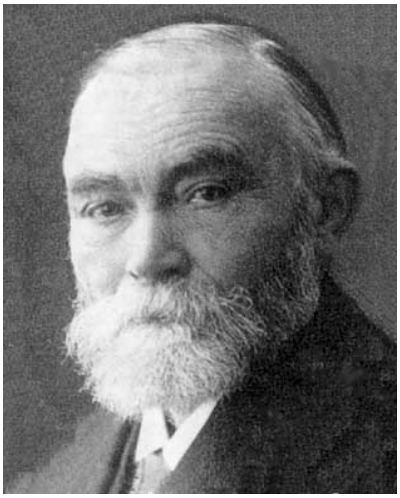
La începutul secolului trecut, într-o perioadă de timp relativ scurtă, cam de la începutul fizicii moderne (1905) şi până la descrierea lui Alan Turing a maşinii sale în 1938, o preocupare majoră a matematicienilor, şi a filosofilor deopotrivă, în teoria cunoaşterii a fost fundamentele matematicii. În acest sens s-au formulat o multitudine de opinii şi păreri care în final au dus la fundamentarea ştiinţei calculatoarelor şi a informaticii. *Fundamentele matematicii se definesc ca studiul logicii și a bazelor filosofice ale matematicii incluzând problema dacă axiomele unui sistem dat asigură consistența și completitudinea acestuia.* Evident că nu vom menţiona toate opiniile şi dezvoltările intelectuale asupra fundamentelor matematicii.

În vara anului 1900, în Paris s-au ţinut două conferinţe internaţionale importante. Prima a fost a filosofilor în 1-5 august. A doua a matematicienilor în 6-12 august. Ambele conferinţe au dezbătut starea disciplinei lor la nivelul fundamentelor, precum şi principalele direcţii de evoluţie ale acestora. Problema unui limbaj perfect şi a comunicării ştiinţifice internaţionale a fost punctul comun al ambelor conferinţe. Încă din timpul lui Galileo oamenii de ştiinţă au fost interesaţi în a defini şi construi un limbaj perfect, în sensul de a fi universal şi neambiguu, capabil de a susţine comunicarea atât între specialişti cât şi cu publicul larg, pentru reducerea efectului Turnului Babel. Încercările de a introduce un limbaj artificial internaţional făcute între alţii de: Descartes prin propunerea unor modificări ale gramaticii pentru a permite limbajului să suporte „*filosofia adevărată*" a ideilor clare şi distincte; Johann Martin Schleyer (1831-1912) care a inventat în 1879 limbajul *Volapük*, probabil primul limbaj artificial internaţional; Ludwik Zamenhof (1859-1917) care a inventat cel mai celebru limbaj artificial *Esperanto*; sau Peano cu al său *Latino sine flexione*, au fost toate un eşec. Totuşi aceste propuneri au avut ca efect introducerea unui sistem de notaţii logice şi simbolice standard, care s-au finalizat cu introducerea formalismului axiomatizat în matematică.

Principalele concepte şi direcţii de cercetare asupra naturii şi fundamentelor matematicii, precum şi cei mai iluştri actori care le-au formulat şi susţinut au fost după cum urmează.

Gottlob Frege

Fondatorul logicismului, poziţie care afirmă că întreaga matematică se reduce la un set de relaţii derivate una după alta numai prin mijloace logice, fără nici o referinţă la concepte matematice, cum ar fi de exemplu cel de număr. Ideea fundamentală a lui Frege este că matematicile sunt o ramură a logicii din care rezultă toate adevărurile matematice. Pentru a susţine această poziţie Frege a construit calculul propoziţional, pe care l-a axiomatizat, a elaborat logica predicatelor şi mai ales a definit funcţia logică ca un concept fundamental în logica predicatelor. Frege realizează prima construcţie a logicii matematice insistând asupra axiomatizării logicii ca un sistem formal. Pentru el modelul central l-a constituit geometria lui Euclid. El merge mai departe decât Euclid în sensul necesităţii ca „*toate mijloacele întrebuinţate pentru a obţine concluziile şi a efectua demonstraţii să fie date în prealabil*".


Gottlob Frege (1848-1925)

Deşi ideea de sistem formal fusese lansată de Leibniz (1646-1716), totuşi Frege a fost primul care a construit un sistem formal, anume *sistemul formal al calculului propoziţional*. El susţine că limba obişnuită se dovedeşte insuficientă, când este vorba să păzim gândirea de erori, şi de aceea este necesar să construim o limbă numai din semne din care să se elimine orice echivoc. Această idee a fost dezvoltată de Giuseppe Peano (1858-1932), iar mai târziu Ludwig Wittgenstein (1889-1951) a încercat să utilizeze conceptele matematice ale lui Frege în limbajul natural, fără însă a obţine rezultate de valoare. Frege a fost sursa de inspiraţie a

lui Rudolph Carnap (1891-1970) şi a diferitelor şcoli de pozitivism. Proiectul său a fost continuat şi dezvoltat de Bertrand Russell (1872-1970) şi Alan Whitehead (1861-1947).

Giuseppe Peano

Fondatorul simbolismului modern, întemeietorul primului sistem axiomatizat al numerelor naturale şi al logicii. Peano contează în istoria fundamentelor matematicii prin două proiecte fundamentale: *Formulario Mathematico* şi axiomatizarea sistemului numerelor naturale.

Giuseppe Peano (1858-1932)

Ambele aceste proiecte provin din dorinţa lui de a universaliza matematica, de a face inteligibil conţinutul, sistemul de notaţii şi limbajul matematicii. De fapt, Peano a formulat, dezvoltat şi consolidat următoarele şase proiecte. 1) *Formulario Mathematico* care concentra într-o singură carte toate principiile matematice. 2) Propunerea de construcţie a unui nou sistem notaţional în matematici. Acesta este cel mai faimos proiect deoarece formalismul iniţiat şi utilizat de Peano a fost adoptat de Russell şi apoi de întreaga comunitate ştiinţifică. 3) Axiomatizarea sistemului numerelor naturale prin propunerea a trei idei primitive şi a cinci axiome. Cu acestea Peano reuşeşte să reconstruiască aritmetica numerelor naturale şi apoi întreaga aritmetică. 4) Crearea unui nou limbaj: *latino sine flexione (interlingua)*, un limbaj perfect care a murit odată cu creatorul lui.

5) Crearea „*Vocabulario de Latino internationale, comparato cum Anglo, Franco, Germano, Italo, Russo, Graeco et Sanscrito*" ca răspuns la provocarea lui Vasiliev. 6) Construcţia unui calendar perpetuu valabil până în 2599 pentru *Academia delle Scienze*.

Formulario Mathematico, lucrare care a cunoscut cinci ediţii succesive, a fost proiectată pentru a conţine toate principiile matematice (evident cunoscute până la acel moment), cu toate propoziţiile, teoremele, demonstraţiile şi metodele lor. Lucrarea conţine 4200 de teoreme, precum şi detalii biografice a peste 300 de matematicieni. Întreaga lucrare este scrisă în *latino sine flexione.* Pentru a înţelege importanţa acestei lucrări, care a introdus un nou simbolism logic, este esenţial să menţionăm faptul că aceasta a fost prima dintr-o lungă serie de încercări similare care includ: „*Principia Mathematica*" a lui Russell (1913), „*The Foundations of Mathematics*" a lui Hilbert (1934) şi „*Elements of Mathematics*" a lui Bourbaki (1939-1967). Fără îndoială că această lucrare a influenţat definitiv gândirea matematicii din secolul al XX-lea şi este chiar mai importantă decât axiomatizarea sa a numerelor întregi.

Sfârşitul secolului al XIX-lea şi începutul secolului al XX-lea a găsit lumea matematicii în efortul aritmetizării matematicii. De la Gauss în geometrie, la Cauchy şi Weirstrass în analiză, efortul matematicienilor a fost de a reduce matematica la început la algebră şi apoi la aritmetică. De fapt aceasta era vechea idee lui Pitagora care asocia anumite „puteri" numerelor întregi. Aritmetica se considera nu numai ca suficientă, dar şi necesară pentru a fundamenta matematica. Leopold Kronecker (1823-1891), de exemplu, era convins că „*God made the natural numbers; all else is the work of man*".

Ideea fundamentală a lui Peano este de a juxtapune logica şi matematica în aşa fel încât acurateţea matematicii să fie extinsă la logică. Astfel, Peano reuşeşte să axiomatizeze logica bazându-se pe patru simboluri şi 12 propoziţii primitive. Peano ca şi Frege realizează trecerea de la logică, ca metodă, la logica matematică ca sistem matematic universal şi ca sistem axiomatic. De fapt, Peano a fost primul care a rezolvat în mod complet, cu ajutorul

unui număr mic de simboluri, problema pusă de Leibniz, şi anume, *să se descrie complet în simboluri, cu semnificații precis enunțate și supuse unui calcul analog calculului algebric, conceptele și propozițiile logice* [Dumitriu, vol. 4, 1998, pg. 86]. Altfel spus, Peano a realizat visul lui Leibniz de a construi o *Enciclopedie*, un rezumat sistematic al tuturor ştiinţelor umane, cu ajutorul unei notații ideografice, pe care o numea *Caracteristica universală* [Couturat, 1899]. Peano a introdus metoda axiomatică modernă în matematici. Axiomatizarea aritmeticii făcută de el reprezintă un moment crucial în fundamentele matematicii, iar expunerea geometriei în formalismul logic este prima în istoria fundamentelor acestei ştiinţe. Mai târziu Hilbert va finaliza axiomatizarea geometriei plane şi în spațiu.

David Hilbert

Fondatorul formalismului, poziție care afirmă că matematica constă numai în generarea unor combinații de simboluri conform unor reguli arbitrare şi aplicarea logicii. La fel ca Frege şi Russell, Hilbert pune la baza fundamentelor matematice logica simbolică, pe care el o numeşte *logica teoretică* sau *logica matematică*. Ideea lui este că „*gândirea logică își găsește imaginea într-un calcul algebric*". Hilbert face o distincție clară între „*limbajul în care vorbim într-o teorie matematică*" și „*limbajul în care vorbim despre această teorie*". În concepția lui Hilbert sarcina logicii este de a combina simboluri pure fără nici un interes asupra semnificației acestora. Altfel spus, logica nu este altceva decât un joc de simboluri, care se supune unor anumite reguli. La Hilbert logica devine o *logică a semnului*.


David Hilbert (1862-1943)

Semnul are o foarte mare importanță la Hilbert, purtând două caracteristici: „*pe de-o parte semnul depozitează o regulă formală și, pe de altă parte, semnul neavând nici un conținut, el are o mobilitate completă în domeniul sensibilului, o bogăție întreagă intuitivă, putând să i se substituie*" [Anton Dumitriu, 1998, vol. IV, pg. 153]. Pentru Hilbert semnul este totul: „**Am Anfang, so heisst es hier, ist das Zeichen**" – „*La început, așa se poate spune aici, este semnul*". Orice teorie matematică poate fi formalizată complet. „*Matematicile sunt libere de orice presupoziție, și pentru a le găsi fundamentele Hilbert nu are nevoie nici de „bunul Dumnezeu" – cum a avut Kronecker - nici de ipoteza unei dispoziții speciale a inteligenței noastre, cum este principiul inducției complete la Poincaré, nici de o intuiție originară ca Brouwer și nici de axiomele infinitului, a reductibilității etc. ca Russell și Whitehead*" [Anton Dumitriu, 1998, vol. IV, pg.153].

Hilbert a arătat importanța metodei axiomatice în care se pleacă de la axiome[3] sau postulate[4] şi utilizând deducția lui Aristotel se obțin teoreme valide. Această idee de a face matematică era foarte veche, fiind cunoscută de greci, în particular de Euclid şi geometria sa în care se vede un sistem matematic foarte clar şi de mare frumusețe. Intenția lui Hilbert a

[3] Termenul **axiomă** αξιωμα - ατοζ de la αξιοω, înseamnă adevăr fundamental care nu trebuie demonstrat, deoarece este evident prin el însuşi. Ştiința demonstrativă presupune cu necesitate axiome, care sunt „*premisele prime ale demonstrației*".

[4] Termenul **postulat** provine din limba latină: *postulare = a cere*, *postulatum = cerere*; aceasta fiind traducerea cuvântului grecesc *aitimata*. Distincția dintre axiome şi postulate o face Euclid în „*Elemente*". Proclus, în „*Comentariile*" sale face o distincție între axiomă şi postulat. Axiomele şi postulatele au în comun faptul că nu au nevoie de demonstrație, ci se consideră *evidente* ca *începutul celor ce urmează*. Proclus consideră că postulatele sunt specifice materiei geometrice, pe când axiomele materiei numerice, teoriei generale a cantității şi a mărimii.

fost de a fi foarte exact în definiții, în concepte matematice, în gramatică şi limbaj. Ceea ce a căutat el a fost de a construi un limbaj propriu matematicii în care **toate** aserțiunile să fie foarte clare, neambigue. De fapt, această concepție de a face matematică are o istorie lungă începând cu lucrările lui Leibniz, Boole, Frege şi Peano. Ceea ce a dorit Hilbert a fost de a depăşi toate realizările de până la el şi a formaliza toată matematica. Surpriza a fost că tot acest efort intelectual s-a finalizat cu un eşec, dar un eşec foarte productiv. În fapt, Hilbert prin întrebările pe care şi le-a pus a creat o nouă disciplină numită *metamatematica*, un domeniu de introspecție al matematicii în care se studiază ceea ce matematica poate să facă sau ceea ce ea nu poate face.

Hilbert a pus în practică ideile sale axiomatizând geometria Euclidiană fără a face apel la referințe spațiale sau intuiție. În 1905, şi apoi în 1918, el a încercat să stabilească un fundament pentru matematici prin demonstrarea consistenței, adică „*un raționament logic finit nu poate conduce la o contradicție*". Dar, în 1931 Kurt Gödel (1906-1978) a arătat că acest ideal nu este realizabil, adică în cadrul unei teorii axiomatizate se pot formula propoziții care sunt nedecidabile. Altfel spus, nu se poate cunoaşte cu siguranță că axiomele matematicii nu conduc la contradicții.

Luitzen Egbertus Brouwer


Luitzen Egbertus Brouwer (1881-1966)

Fondatorul intuiționismului, poziție care priveşte natura matematicii ca o construcție mentală guvernată de legi auto-evidente. Pentru Brouwer „*orice propoziție care are un conținut trebuie să indice una sau mai multe stări de lucruri bine determinate și accesibile experienței noastre*". Altfel spus, o propoziție are un conținut când este legată prin intuiția noastră imediată de anumite stări de lucruri. Intuiționismul a apărut în principal ca o necesitate de a da un răspuns problemei paradoxelor dezvoltându-se ca o încercare de a da o explicație asupra naturii şi fundamentelor matematicii [Anton Dumitriu, 1998, vol. IV, pp.160]. Intuiționismul afirmă că matematica este o funcție naturală a intelectului nostru, a gândirii, o creație a spiritului omenesc.

Limbajul în care se exprimă, fie cel obişnuit, fie formalizat, are funcția de comunicare a rezultatelor şi nu oferă altceva decât o imagine a matematicii. Matematica este cu mult mai mult decât această imagine şi limbajul în care exprimăm rezultatele matematicii [Heyting, 1934, pg. 106]. Altfel spus, matematica este independentă de limbaj şi de orice altă disciplină.

Brouwer este considerat fondatorul topologiei. În teza sa de doctorat din 1907 asupra Fundamentelor Matematicii, Brouwer a atacat fundamentele logice ale matematicii şi în 1908 în lucrarea „*On the Untrustworthiness of the Logical Principles*" el rejectează utilizarea demonstrațiilor matematice ale principiului terțului exclus – orice afirmație matematică este fie adevărată, fie falsă, nefiind posibilă nici o altă posibilitate. În 1918 Brouwer a publicat o teorie a mulțimilor, în anul următor o teorie a măsurii şi apoi în 1923 o teorie a funcțiilor, toate acestea dezvoltate *fără* a utiliza principiul terțului exclus.

Kurt Gödel

Acționând în sensul transformării logicii într-un aparat simbolic fără conținut, logicienii au fost puşi în fața unor probleme foarte serioase. Una dintre acestea a fost *problema deciziei* (*Entscheidungsproblem*). Adică, *dacă logica lucrează cu simboluri care se supun unor reguli foarte exacte care asigură deducției o rigoare matematică, cum este posibil ca operând numai cu simboluri să avem certitudinea că un proces deductiv ne conduce la adevăr* ?

Aceasta ne conduce la problema fundamentală a logicii: „*să se găsească un procedeu care să permită întotdeauna să recunoaștem dacă o propoziție este adevărată sau falsă într-o teorie dată*”. Altfel spus, problema deciziei este problema construcției unei științe deductive, și acesta a fost programul lansat de Hilbert la începutul secolului trecut și mai înainte de Frege în abordarea sa logicistă.

Kurt Gödel a considerat problema deciziei într-o formă mult mai generală – aceea a *nedecidabilității*. Într-un studiu publicat în 1931 în „*Monatshefte für Mathematik und Physik*”, și devenit celebru atât în lumea matematicii cât și a filosofiei, Gödel a ruinat speranța matematicienilor de a exprima întreaga matematică într-un mod formal este iluzorie și că există în sistemele logice formale principale, (ca acela al lui Russell și acela al lui Zermelo-Fraenkel dezvoltat de von Neumann), sau în sistemele înrudite probleme relativ simple care nu pot fi rezolvate. Altfel spus, Gödel arată că în sisteme logico-formale, utilizând semnele sistemului, axiomele și regulile de inferență ale acestuia, este posibil să construim o expresie corect formulată în sistem, care să fie nedecidabilă.

Kurt Gödel (1906-1978)

Înțelesul termenului de „*nedecidabil*” este că expresia respectivă nu este nici *demonstrabilă* și nici *nedemonstrabilă* în sistemul logic în care a fost formulată, conform regulilor acelui sistem [Anton Dumitriu, 1998, pg. 226]. Teorema lui Gödel zice că: „*în orice clasă de formule necontradictorii există propoziții nedecidabile (unentscheidbare)*”. Altfel spus, un sistem care îndeplinește anumite condiții generale și în primul rând de a nu fi contradictoriu și de a conține aritmetica (adică de a fi destul de general pentru ca aritmetica să fie formalizabilă în el) este limitat, adică în asemenea sisteme logico-formale se pot construi în mod corect anumite propoziții nedecidabile. *În acest sens o teorie axiomatizată este* ***consistentă*** *dacă este imposibil să demonstrăm simultan o afirmație și negația ei. Teorema lui Gödel zice că* ***orice teorie axiomatizată este incompletă****. O teorie axiomatizată este* ***completă*** *dacă orice afirmație corect formulată sintactic în acea teorie se poate demonstra că este fie adevărată fie falsă.* Aceste concepte sunt foarte importante fiindcă definesc conținutul teoriilor științifice axiomatizate.

Demonstrația teoremei de incompletitudine a lui Gödel este foarte ingenioasă cu multe detalii tehnice complicate. Dar, ceea ce este remarcabil în această demonstrație este faptul că se întâlnesc o multitudine de concepte proprii limbajului Lisp. Demonstrația lui Gödel implică definirea în mod recursiv a unui mare număr de funcții, funcții care tratează liste, exact cum se lucrează în Lisp. Așa încât, chiar dacă în 1931 nu exista nici un calculator și nici limbaje de programare, vedem cum ideile unui limbaj de programare stau la baza lucrării lui Gödel.

Acest rezultat a fost (și continuă să fie) foarte tare, tulburând lumea oamenilor de știință în sensul că acesta a lovit chiar în ceea ce aceștia credeau și considerau a fi baza lor, fundamentul lor – matematica – ruinând speranța posibilității predicției în Natură. Această teoremă a lui Gödel împreună cu alte descoperiri privind anumite limite fundamentale în ceea ce privește abilitatea noastră de a utiliza deducția în cadrul ciclului metodologic care definește metoda științifică a instituit *pierderea inocenței oamenilor de știință.* Într-adevăr, limitele în ceea ce privește rezolvarea analitică a problemei celor trei corpuri, apoi faptul că dicționarul care asociază informațiile fizice cu cele matematice și care astfel permite obținerea de predicții fizice, în cele mai multe cazuri nu se poate construi, decât numai în cazul utilizării unui înțeles mult mai limitat al noțiunii de predicție, faptul că partea inductivă

a ciclului metodologic este mult mai fragilă decât se spera, idealizarea că sistemele fizice se pot trata ca făpturi izolate din univers, aşa de frecvent utilizată în teorii, nu se poate utiliza pentru a stabili proprietăţi de stabilitate structurală ale acestor teorii a condus la cea de-a doua metamorfoză a ştiinţei. Aceasta consemnează tocmai acest lucru: *pierderea inocenţei deductivităţii, pierderea inocenţei determinismului, instabilitatea modelelor matematice a sistemelor închise, caracterul incomplet al sistemelor formale.* [Andrei, 2004b].

Rezultatul lui Gödel a însemnat un eşec relativ la încercarea lui Hilbert de a fundamenta ansamblul matematicilor clasice cu ajutorul teoriei demonstraţiei, dezvoltată în întregime într-un cadru finitist. Concluziile negative care apar din teoria lui Gödel precizează limitele sistemelor formale, şi anume [Ladrière 1957]:

- un sistem formal nu poate fi niciodată considerat ca o reprezentare adecvată a unei teorii matematice, în sensul că nu permite să decidem în mod efectiv validitatea unor enunţuri;
- există o dualitate între formal şi intuitiv, câmpul intuitiv nu poate fi niciodată complet formalizabil;
- limitarea formalismelor arată limitarea constructivităţii. Ladrière [1957, p.44] explică relaţia care există între formal şi constructiv arătând că formalismul este domeniul construcţiei. „*Gândirea matematică nu poate progresa în descoperirea obiectului său decât sprijinindu-se pe construcţii; numai în formalism îi apare ceea ce ea poate sesiza. Dar acest obiect depăşeşte întotdeauna cu mult formele în care el se prezintă. Entitatea matematică nu se epuizează în manifestările ei*".

Anton Dumitriu

Această filozofie a limitării sistemelor logico-formale a fost criticată de Profesorul nostru Anton Dumitriu într-o serie de lucrări, dintre care amintim: *Soluţia paradoxelor logico-matematice,* [1966] şi *Mecanismul logic al matematicilor*, [1968].


Anton Dumitriu (1905-1992)

Profesorul susţine că: „*Această filozofie a limitării sistemelor logico-formale se bazează pe presupuneri eronate. Ipoteza fundamentală a acestei filosofii presupune că matematica este un sistem formal şi numai atât, ceea ce este nedemonstrat. Dimpotrivă, toate eşecurile formalismelor (paradoxe, limitări, imposibilitatea de a demonstra necontradicţia formală a aritmeticii etc.) arată că această presupunere nu este justificată. Însăşi noţiunea de* ***decidabilitate*** *sau de* ***demonstrabilitate*** *într-un sistem formal este luată ca o idee în sine, fără a se vedea că această idee nu este un predicat, o proprietate a propoziţiilor unui sistem. Demonstraţia unei propoziţii într-o teorie este o* ***operaţie*** *şi nu o proprietate, care nu se referă numai la propoziţia avută în vedere, ci la toate propoziţiile care servesc la deducerea acelei propoziţii în cadrul sistemului. Demonstraţia are un caracter operaţional-constructiv şi nu se poate spune despre o propoziţie, luată în mod izolat, că ea are proprietatea de a fi demonstrabilă sau nu în cadrul unui sistem.*

Problema pusă de Gödel şi consecinţele ei se datoresc tocmai acestui fapt, al unei interpretări eronate a teoriilor matematice, natura lor reală constructivistă scăpând sistemului formal în care ele sunt traduse.

Actul matematic de gândire are două aspecte sau, cum spune Leibniz, se compune din două arte: ***ars inveniendi*** *şi* ***ars demonstrandi****. Prin arta de a inventa noi obiecte matematice procesul gândirii matematice este în continuu progres; prin arta de a demonstra, raţionamentul echivalează rezultatele obţinute cu cele de la care se pleacă şi le face necesare.*" (*op cit.* p. 231).

Teorema lui Gödel, şi în special acest rezultat privind limitarea sistemelor formale, a constituit punctul de plecare pentru numeroase studii care s-au concretizat în rezultate remarcabile, dintre care cele mai importante ni se par a fi:
- Contribuţiile lui Church în ceea ce priveşte problemele de decizie pentru un sistem formal.
- Teoria lui Turing, bazată pe noţiunea de *procedeu efectiv* şi dezvoltarea maşinii Turing. Un procedeu este efectiv dacă permite să se decidă dacă o proprietate aparţine sau nu unui obiect, adică dacă se poate construi o maşină care poate rezolva automat problema.

Gödel, care a fost un Kantian declarat, a făcut un mare serviciu comunităţii ştiinţifice dirijând cu precizie dezvoltările formale ale fundamentelor matematicii înapoi la chestiunea fundamentală a epistemologiei clasice. *Problema reală nu este aceea a construcţiei edificiilor logice elaborate, ci înţelegerea naturii şi a sursei axiomelor pe baza înţelegerii noţiunilor primitive.* Altfel spus, Gödel a schimbat definitiv perspectivele formalizării, adică perspectivele asupra aşa-ziselor *sisteme logico-matematice formalizate.* Aceste sisteme nu au nici o valoare în afară de una istorică. Ideea lui Hilbert şi a şcolii neopozitiviste că problema cea mai importantă a fundamentelor matematice este de a construi un **limbaj artificial** cu reguli sintactice precise şi că printre aceste limbaje ar exista un *limbaj universal*, perfect, care s-ar putea identifica cu matematicile, este considerată astăzi un eşec.

Odată cu înfrângerea, mai mult sau mai puţin definitivă, a şcolilor Logiciste şi Formaliste, şi abordarea lui Turing privind reducerea problemelor la chestiuni de programare, controversa privind fundamentele matematicii a luat sfârşit. Lucrările lui Turing au introdus noi concepte ale complexităţii limbajului care au format baza pentru dezvoltările lui Noam Chomsky şi fundamentele teoriei complexităţii. Gödel a arătat că comportarea sistemelor pur formale nu poate fi în mod complet descrisă prin logica formală, şi aceasta este de fapt fundamentul complexităţii, a nepredictibilităţii, elaborat şi consolidat de Turing, ilustrând încă odată bogăţia şi profunzimile lumii spirituale.

Alan Turing

Fondatorul ştiinţei calculatoarelor şi iniţiatorul cercetărilor în domeniul inteligenţei artificiale. Motivat de rezultatele lui Gödel, de a căuta o metodă algoritmică pentru a determina dacă sau nu o propoziţie dată (oarecare) este nedecidabilă, cu scopul final de a o elimina din matematică, Turing, în 1936, a demonstrat că nu poate exista o astfel de metodă universală şi deci matematica întotdeauna va conţine propoziţii nedecidabile. Pentru a ilustra aceasta Turing a imaginat o maşină care posedă proprietăţile minimale ale unui sistem de calcul modern – un program finit, o bază de date suficient de mare şi o procedură de a efectua operaţii matematice pas cu pas – *maşina Turing*. Utilizând metodele lui Hilbert, Turing a ruinat speranţele lui Hilbert şi Frege că toate propoziţiile matematice se pot exprima ca un set de axiome şi teoreme derivate din acestea.


Alan Mathison Turing (1912-1954)

În 1948 Turing a publicat articolul „*Rounding-off Errors in Matrix Processes*" în The Quarterly Journal of Mechanics and Applied Mathematics, vol. I, pp.287-308, în care prezintă preocupările lui privind rezolvarea sistemelor de ecuaţii algebrice liniare [Turing, 1948]. O problemă foarte veche şi aparent banală. Lucrarea este foarte bine cunoscută în lumea analiştilor calculului numeric, dar aproape necunoscută logicienilor şi celor interesaţi de ştiinţa calculatoarelor. În prima secţiune a acestei lucrări Turing tratează problema „*măsurii cantităţii de muncă depusă pentru realizarea unui calcul numeric*": „*It is convenient to have a measure of the amount of work involved in a computing process, even*

though it be a very crude one ... We might, for instance, count the number of additions, subtractions, multiplications, divisions, recordings of numbers,...". Să observăm că Turing consideră un ***model de calcul*** în care numerele reale sunt privite ca entități, iar operațiile algebrice, comparațiile, atribuirea de valori etc. sunt considerate ca unități ale cantității de muncă.

Aceasta a fost prima încercare de a defini o măsură a cantității de muncă depusă pentru a rezolva efectiv o problemă. Din punct de vedere matematic era suficient să se precizeze un rezultat privind existența soluției. Și de fapt aceasta este esența matematicii. Rezultatele cele mai tari din matematică sunt teoremele de existență și cele care precizează proprietățile soluțiilor anumitor obiecte matematice. Turing ridică problema efortului de calcul și prelungește astfel analiza întrebându-se dacă rezultatele matematice privind rezolvarea unei probleme, de altfel foarte frumoase fiindcă *theorema* în greaca veche însemna *spectacol,* au vre-o valoare efectivă.

3. Fundamentele informaticii

În tradiția lui Leibniz, Turing a fost preocupat de calcul și mai exact de calculabilitate. Ne reamintim că Hilbert și-a imaginat că trebuie să existe „*o procedură mecanică*" pentru a decide dacă o demonstrație este adevărată sau nu. Hilbert nu a clarificat niciodată ceea ce însemna pentru el procedură mecanică. Turing a fost acela care a precizat sensul afirmației lui Hilbert: „*procedură mecanică înseamnă o mașină*", o mașină de un anumit tip pe care acum o numim *mașina Turing*.

Teoria calculabilității vine din două mari tradiții, care cu foarte mici excepții s-au dezvoltat în paralel, de cele mai multe ori ignorându-se reciproc. Pe de-o parte avem analiza numerică și calculele științifice, și pe de altă parte avem teoria calculului așa cum apare în logică și știința calculatoarelor. Ambele aceste tradiții utilizează noțiunea de **algoritm**. Exemplul paradigmatic de algoritm, cel mai des invocat în textele de analiză numerică (și de programare matematică), este *algoritmul Newton*. Pe de altă parte, modelul de calcul prezent în textele de știința calculatoarelor asupra algoritmilor este *mașina Turing*. Curios este faptul că metoda Newton nu este discutată în textele de știința calculatoarelor, iar mașina Turing este total ignorată în cele de analiză numerică.

Dar, ceea ce este esențial și ceea ce caracterizează diferențele fundamentale între aceste două tradiții sunt: *spațiile în care lucrează fiecare dintre aceste tradiții; matematica utilizată și problemele considerate*. În analiza numerică și calculele științifice algoritmii sunt definiți peste corpul numerelor reale sau complexe și matematica utilizată este aceea a continuului. Este foarte interesant de menționat aici definiția articulată de Trefethen [SIAM News, Nov. 1992] ca: „*studiul algoritmilor pentru problemele matematice continue*". Cuvântul cheie în această definiție este algoritm, așa încât nucleul analizei numerice îl constituie: „*proiectarea și analiza algoritmilor pentru rezolvarea problemelor continue*". În definiția lui Trefethen *continuu* înseamnă că variabilele problemei aparțin corpului numerelor reale sau complexe, spre deosebire de cele discrete care constituie domeniul de preocupare a altor discipline. Pe de altă parte, în teoria calculabilității și știința calculatoarelor fundamentale sunt 0 și 1, iar matematica este discretă. Problemele de analiză numerică vin din tradiția clasică a rezolvării ecuațiilor și din calculul diferențial. Cele ale științei calculatoarelor au mai mult o origine combinatorială. Este important să notăm că teoria computabilității și teoria complexității din știința calculatoarelor în general nu este adecvată pentru analiza problemelor care sunt proprii analizei numerice și calculului științific. Totuși, remarcăm imediat faptul că analiza numerică în afară de reprezentarea în virgulă mobilă și analiza erorilor nu a generat o teorie a complexității comparabilă cu cea proprie științei calculatoarelor. *Problemele de computabilitate și cele de complexitate constituie fundamentele informaticii, ca disciplină privind* ***coborârea în computațional a conceptelor matematicii,*** *în care aceste probleme își găsesc locul.*

3.1. Problema complexității computaționale

Teoria complexității computaționale se concentrează asupra înțelegerii puterii calculului în sensul de a preciza semnificația întrebării „*de ce anumite probleme apar a fi mai dificil de rezolvat decât altele*" ? Deşi în ultimul timp s-au înregistrat progrese semnificative în acest domeniu, multe probleme fiind deja plasate în anumite clase de complexitate, totuşi foarte multe aspecte fundamentale ale complexității computaționale rămân nerezolvate.

Calculul se poate gândi ca o operație care determină o ieşire corespunzătoare unei intrări date. Deci o *problemă computațională* este specificată ca o relație între intrări şi ieşiri. Un algoritm care rezolvă problema consideră o intrare acceptabilă, numită instanțiere, şi calculează o ieşire care satisface relația intrare-ieşire. Un caz special, important, de problemă computațională este o *problemă de decizie*, în care pentru fiecare intrare ieşirea este restricționată numai la două a valori „yes" şi „no". Un alt tip de problemă computațională, foarte important, este *problema de optimizare*, în care este vorba de a efectua un calcul cu scopul de a minimiza sau maximiza un anumit criteriu.

Pentru precizarea calculului s-au propus trei modele computaționale, adică trei modele matematice ale unui calculator: *mașina Turing*, *mașina flow-chart logică* (logical circuit machine) şi *mașina cu acces aleatoriu* (random access machine – RAM). Deşi între aceste maşini sunt diferențe, acestea nu sunt de substanță, lucrul cu ele ținând mai mult de o convenienței şi tradiție.

Este foarte clar că motivația logicienilor de a dezvolta o teorie a computabilității prin anii 1930 nu avea nimic de-a face cu calculatoarele. Acele figuri ilustre ca Hilbert, Gödel, Turing şi alții erau interesați de „*ce înseamnă că o problemă este decidabilă*". Şi aceasta a fost motivată de problema a 10-a din lista lui Hilbert, pusă de el în 1900 [Hilbert, 1900].

Problema a 10-a a lui Hilbert.
Fie $X = \{ f \in Z[x_1, \ldots, x_n], n > 0 \}$ şi $S = \{ f \in X : \exists \xi \in Z^n, f(\xi) = 0 \}$.
Este S decidabilă ? Adică, dată o ecuație diofantică, se poate decide printr-un număr finit de operații dacă ea are sau nu o soluție întreagă ? (De fapt, Hilbert a fost mai categoric cerând să se elaboreze o procedură de determinare efectivă a soluției.)

(Vezi, de exemplu, lucrarea „*Probleme fundamentale*", [Andrei, 2005] unde se prezintă câteva detalii asupra celor 23 de probleme ale lui Hilbert cât şi asupra celor 18 probleme ale lui Smale.) Formalizarea noțiunii de decidabilitate, dată de logicieni, a avut consecințe profunde în dezvoltarea teoriei computabilității şi a ştiinței calculatoarelor. Vom da o definiție.

Definiție. *O mulțime* S $(S \subset X)$ *este decidabilă dacă funcția ei caracteristică* χ_S *care ia valoarea 1 pe* S *şi 0 pe* $X - S$*, este calculabilă în timp finit pe o mașină.*

Noutatea aici este faptul că decidabilitatea se defineşte în funcție de o maşină. O astfel de maşină 0-1 se numeşte *procedură de decizie pentru* S. La intrarea $x \in X$ maşina răspunde la întrebarea „Este x în S ?" cu o ieşire cu valoarea 1 dacă YES şi 0 dacă NO. În acest context $X = \Sigma^*$, mulțimea şirurilor nemărginite definite pe mulțimea finită Σ. Σ^* este numărabilă.

Aparent s-au propus multe maşini, dar surpriza a fost că toate dau *exact aceeaşi clasă* de funcții computabile. Astfel, clasa transformărilor intrare-ieşire a maşinilor Turing sunt exact funcțiile computabile aşa cum sunt derivate din sistemele de producție ale lui Post, sau din maşinile flow-chart, sau maşinile cu acces aleatoriu sau descrierea axiomatică a funcțiilor recursive etc. Aceasta a generat credința, cunoscută ca *ipoteza lui Church*, că funcțiile calculabile formează o clasă naturală şi că orice noțiune informală de procedură sau algoritm

se poate realiza pe oricare dintre aceste maşini. Următoarea teoremă precizează puterea modelelor computaţionale de mai sus.

Teorema 1. *Următoarele clase de probleme sunt identice:*
a) Clasa problemelor computaţionale rezolvabile pe o maşină Turing.
b) Clasa problemelor computaţionale rezolvabile pe o maşină cu acces aleatoriu.
c) Clasa problemelor computaţionale rezolvabile pe o familie de circuite logice care se pot genera pe maşini Turing.

Teorema următoare, care într-un anumit sens este complementara teoremei 1 de mai sus, este o consecinţă directă a faptului că există un număr nenumărabil de problema computaţionale, dar numai un număr numărabil de maşini Turing.

Teorema 2. *Nu toate problemele de decizie sunt rezolvabile pe o maşină Turing.*

În 1970 Yuri Matijasevich a dat un răspuns negativ pentru problema a 10-a a lui Hilbert arătând că funcţia caracteristică asociată χ_S nu este calculabilă Turing şi deci prin ipoteza lui Church nu există nici o procedură pentru a decide rezolvabilitatea în numere întregi a ecuaţiilor diofantice.

Înţelegerea acestei limitări intrinseci a puterii calculelor vine direct dintr-un rezultat de logică matematică. În particular, teorema de incompletitudine a lui Gödel [1931] conţine primul tip de rezultate de nedecidabilitate şi furnizează multe idei esenţiale care au fost mai târziu utilizate de Church, Post şi Turing în fundamentarea naturii calculului.

O preocupare constantă a logicienilor a fost aceea a clasificării problemelor ca probleme decidabile sau nedecidabile. Un efort considerabil a fost dedicat studiului ierarhizării problemelor nedecidabile. Problemele decidabile, în particular cele finite, au fost de un interes mai mic. Totuşi, prin anii 1960, -70 computeriştii au observat că nu toate problemele decidabile sunt la fel, şi că sunt unele care pun probleme când se încearcă determinarea unei soluţii admisibile.

Exemplu. Problema satisfiabilităţii (SAT)
$X = \{ f \mid f : Z_2^n \to Z_2 \}$ este mulţimea funcţiilor Booleene şi
$S = \{ f \in X : \exists \xi \in Z_2^n, f(\xi) = 0 \}$.

Se presupune că X este scufundat în $\{0,1\}^*$ utilizând o procedură de codificare naturală. Este foarte clar că testarea sistematică a tuturor celor 2^n argumente pentru o funcţie Booleană f constituie o procedură de decizie pentru S. Această procedură, în cel mai prost caz, consideră un număr exponenţial de operaţii. Nu se cunoaşte dacă SAT este *tratabilă*, adică dacă există o procedură de decizie în timp polinomial pentru S.

Definiţie. *Problema de decizie* (X,S) *este în clasa* P *dacă funcţia caracteristică* χ_S *este calculabilă în timp polinomial, adică calculabilă de o maşină Turing polinomială în timp. O maşină Turing polinomială în timp este una care se opreşte în* $c(\text{size } x)^k$ *operaţii pentru o constantă* c *fixă,* $k \geq 0$ *şi toate intrările* $x \in X$. *Prin* $\text{size } x$ *înţelegem lungimea secvenţei* x, *adică lungimea în biţi dacă* $\Sigma = \{0,1\}$.

Prin *calculabilitatea* unei funcții f înțelegem definirea unui algoritm care obține o ieşire $y = f(x)$ pentru o intrare x. Prin *verificabilitatea* unei funcții f înțelegem determinarea predicatului $R(x, y)$ care se presupune că primeşte valoarea TRUE dacă şi numai dacă $f(x) = y$.

> În anii 1970 Steve Cook şi Leonid Levin independent au formulat un răspuns la problema tratabilității lui SAT prin articularea chestiunii: $P = NP$?

O problemă de decizie $(X, S) \in class\ NP$ dacă pentru fiecare $x \in S$ există o verificare în timp polinomial a acestui fapt. Notăm că $SAT \in NP$. Dacă o funcție Booleană $f \in S$, atunci există un punct $\xi \in Z_2^n$ care permite o verificare în timp polinomial a faptului că calculul lui f în argumentul $\xi = [\xi_1, \ldots \xi_n]$ produce valoarea 0 pentru f. Dacă $f \notin S$, atunci nu există nici un astfel de punct care să permită această verificare.

Importanța chestiunii $P = NP$? a devenit clară când Karp [1972] a arătat că tratabilitatea fiecărei probleme dintr-o mulțime de 21 de probleme, aparent nelegate între ele, este echivalentă cu tratabilitatea lui SAT. De atunci pentru foarte multe probleme s-a putut demonstra echivalența cu SAT.

Aşa cum problema decidabilității a influențat teoria computabilității şi ştiința calculatoarelor, tot aşa chestiunea $P = NP$? a avut consecințe profunde. Aceasta este una din problemele faimoase nerezolvate atât în ştiința calculatoarelor cât şi în matematică. Dihotomia dintre clasele P şi NP a constituit fundamentul unor aplicații foarte importante ale teoriei complexității, cum ar fi criptografia şi securitatea transmisiei datelor.

Observăm că tradiția dată de maşina Turing şi deci acest model de calcul a condus la o *teorie a computabilității şi a complexității* foarte dezvoltată şi foarte bogată în rezultate, cu aplicații directe în calcul. De fapt maşina Turing a creat noi speranțe pentru a înțelege cât de dificile sunt problemele clasice (rezolvate prin algoritmul Newton).

Totuşi, deşi modelul computațional dat de maşina Turing a introdus principalele concepte cu care se lucrează în teoria complexității algoritmilor, şi deci a informaticii, s-a observat că acest model are anumite limite. Aceasta a determinat introducerea de către Lenore Blum, Michael Shub şi Stephen Smale, într-o serie de lucrări, a unui nou model computațional bazat pe o teorie a complexității şi a calculabilității peste inele sau câmpuri arbitrare R. Dacă $R = Z_2 = (\{0,1\}, +, \cdot)$, atunci modelul recuperează teoria clasică a ştiinței calculatoarelor (a calculabilității şi a complexității). Dacă R este chiar câmpul numerelor reale, atunci algoritmul Newton, cum am mai spus algoritmul paradigmatic al analizei numerice, se încadrează într-o manieră foarte naturală în noul model de calcul.

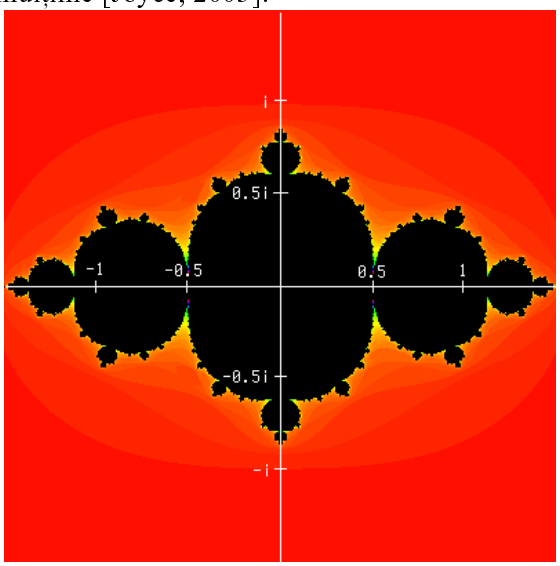
3.1.1. Motivații pentru un nou model computațional

Principalele motivații care au condus la introducerea unui nou model computațional se bazează pe încercarea de reconciliere a celor două tradiții de care vorbeam mai sus. Faptul că rezultatele din analiza numerică, foarte bogate de altfel, nu sunt de loc utilizate în teoria computabilității şi a ştiinței calculatoarelor a constituit o surpriză. *Depășirea limitelor mașinii Turing, introducerea conceptelor proprii analizei numerice în teoria computabilității și a complexității calculului au constituit principalele motive de dezvoltare a unui nou model computațional.* Acesta se fundamentează pe chestiunea decidabilității pe mulțimi continue (nenumărabile) şi pe rezultatele de analiză numerică (efort de calcul, erori etc.).

a) Decidabilitate pe mulțimi continue

Clasic, decidabilitatea este definită numai pentru mulțimi numărabile. Dar, în 1986 Benoit Mandelbrot, motivat de studiul dinamicii sistemelor neliniare a introdus o mulțime

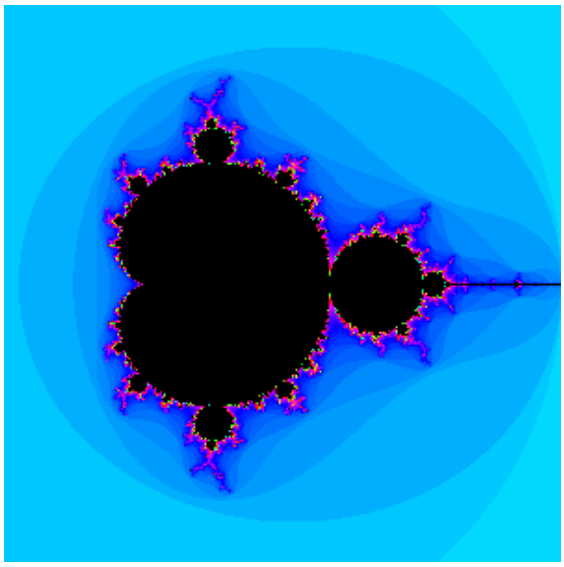
extraordinar de complexă, mulțimea Mandelbrot. În acest sens merită să introducem această mulțime [Joyce, 2003].



Mulțimea Julia pentru funcția $f(z) = z^2 - 0.75$.

Să considerăm o funcție $f: C \to R$, unde C este plnul complex, precum și un punct inițial z_0. Definim *iteratele funcției* f ca șirul $z_0, z_1 = f(z_0), z_2 = f(z_1), \ldots$ Evident că acest șir poate fi mărginit sau nemărginit. Mulțimea punctelor inițiale a căror iterate ramân mărginite se numește *mulțimea Julia* asociată funcției f.

Ca exemplu să consideram funcția $f(z) = z^2 - 0.75$. Figura alăturată arată mulțimea Julia din planul complex $z = x + iy$, unde $-1.5 \le x, y \le 1.5$. Zona neagră a planului conține acele puncte z_0 pentru care iteratele rămân mărginite. În zona mai deschisă iteratele tind mai mult sau mai puțin rapid la infinit. Observăm că mulțimea Julia pentru o funcție foarte simplă ca cea considerată aici este foarte complexă.



Mulțimea Mandelbrot pentru funcția $f(z) = z^2 - \mu$.

Să considerăm acum o familie de funcții parametrizate de un parametru μ. Deși se pot considera orice fel de funcții, totuși cea mai studiată familie este aceea a funcțiilor polinomiale pătratice: $f(z) = z^2 - \mu$, unde μ este un parametru complex. Când μ variază, atunci vom avea mai multe mulțimi Julia în planul complex. Unele dintre acestea vor fi conexe, altele vor fi disjuncte, așa încât acest caracter al mulțimilor Julia va partiționa planul complex al parametrului μ în două părți. Acele valori ale lui μ pentru care mulțimile Julia sunt conexe se numește *mulțimea Mandelbrot* în planul parametrului. În figura alăturată se arată mulțimea Mandelbrot pentru funcția de mai sus, unde $-1 \le x \le 2$ și $-1.5 \le y \le 1.5$ în planul parametrului, (zona neagră). Se observă complexitatea acestei mulțimi, chiar dacă ea este, în esență, foarte simplu de definit. Cea mai mare parte a mulțimii Mandelbrot este o cardioidă.

Observăm că pe ea sunt plasate un numar infinit de cercuri, fiecare dintre acestea având la rândul lor alte cercuri, și aceasta *ad infinitum*. La o privire mai atentă vedem că pe aceste cercuri sunt plasate alte cardioide care la rândul lor au cercuri, și aceasta într-o varietate nesfârșită de dimensiuni din ce în ce mai mici.

Deși regula de definiție a acestei mulțimi este foarte simplă, ca mulțimea tuturor parametrilor $\mu \in C$, astfel încât orbita lui 0 a transformării pătratice $f(z) = z^2 - \mu$ rămâne mărginită, totuși intrinsec mulțimea se prezintă ca o varietate fără sfârșit de structuri foarte elaborate. Mulțimea Mandelbrot este nenumărabilă. Deci, în formalismul clasic dat de modelul computațional Turing, mulțimea Mandelbrot nu este decidabilă. Altfel spus, nu

există un algoritm general de a decide dacă un element (sau punct) aparține sau nu mulțimii. Roger Penrose [1989] se întreabă: „*Could this be an example of a non-recursive [i.e. undecidable] set, truly exhibited before our mortal eyes ?*[5] ” Ca atare, modelul computațional trebuie rafinat pentru a cuprinde cazul mulțimilor nenumărabile.

Un alt aspect care trebuie considerat aici este ceea ce s-ar putea numi „*modelul numerelor raționale*”. Acesta nu este încă formalizat, dar se poate justifica prin faptul că mașinile de calcul sunt finite, apoi numerele reale sunt aproximate (finit) prin numere raționale, și deci problemele sunt definite peste corpul fracțiilor raționale. Totuși, utilizând acest model în teoria complexității, repede intrăm în dificultăți. De obicei computeriștii măsoară complexitatea ca o funcție de mărimea intrării, exprimată în biți. Dar, ceea ce este foarte important, mici perturbații în datele de intrare pot cauza diferențe foarte mari în mărimea intrării. De exemplu, o mică perturbație a lui 1 la $1+1/2^n$ determină o creștere a mărimii intrării de la 1 la $n+1$. Astfel, un algoritm care este polinomial în timp conform definiției care utilizează modelul numerelor raționale nu este de loc invariant la micile perturbații ale intrării. Dacă problema este bine condiționată (vom defini mai jos condiționarea), atunci acest model de computabilitate nu poate fi acceptat.

Un alt exemplu, care este întâlnit peste tot în matematici, este dat de

Problema Hilbert Nullstellensatz peste R (HN_R)

„Dat un sistem de ecuații polinomiale peste un inel R:

$$f_1(x_1,\ldots,x_n)=0,$$
$$\ldots$$
$$f_m(x_1,\ldots,x_n)=0,$$

există un $\xi \in R^n$, astfel încât $f_1(\xi)=0,\ldots,f_m(\xi)=0$?”

Această problemă de decizie peste R o numim HN_R. Dacă R este Z_2, Z sau corpul numerelor raționale Q, atunci HN_R se încadrează în formalismul lui Turing care utilizează codificarea pe biți a întregilor și a numerelor raționale. Problema de decizie peste Z_2 este problema SAT care este decidabilă, dar care nu se cunoaște a fi în P. Peste Z aceasta este problema a 10-a a lui Hilbert, care este nedecidabilă. Peste corpul Q nu se cunoaște dacă problema este decidabilă sau nedecidabilă. Acum, dacă R este corpul numerelor reale sau complexe, atunci HN_R nu se încadrează în mod natural în formalismul Turing.

Această problemă de decizie care utilizează operațiile aritmetice și comparații cu numere complexe furnizează o cale de a intra în amănuntele definirii unui nou model computațional. Într-adevăr, în loc de a utiliza mărimea problemei în funcție de codificarea problemei în biți, Blum, Shub și Smale [1989] propun un model computațional care calculează peste un inel (sau un corp) utilizând operațiile de bază ale inelului (sau corpului) respectiv împreună cu comparațiile pe elementele acestei structuri algebrice de inel (sau corp).

Toate aceste exemple, pe care le-am prezentat mai sus, furnizează o motivație foarte serioasă pentru a schimba modelul computațional clasic al lui Turing.

b) Algoritmi de analiză numerică

O altă motivație pentru adoptarea unui nou model computațional vine direct din tradiția dată de analiza numerică. Rezolvarea oricărei probleme de algebră liniară implică trei componente importante: *dezvoltarea și analiza algoritmilor numerici*, *analiza perturbațiilor*

[5] Roger Penrose, *Mintea noastră cea de toate zilele. Despre gândire, fizică și calculatoare*, Editura Tehnică, București, 1996, pg. 140. Traducere de Cornelia C. Rusu și Mircea V. Rusu a lucrării lui Roger Penrose: *The Emperor's New Mind: Concerning Computers, Minds, and The Laws of Physics*, Oxford University Press, 1989.

şi *implementarea algoritmilor în programe de calcul*. În general, pentru a obţine un răspuns de înaltă acurateţe şi de încredere trebuie să ne asigurăm că: *problema este bine condiţionată*, *algoritmul este numeric stabil în precizia finită a calculatoarelor* şi *algoritmul este bine implementat într-un software de calitate.*

O problemă este bine condiţionată dacă mici schimbări în datele de intrare ale acesteia, altfel spus mici variaţii în parametrii problemei, produc numai mici schimbări în soluţia ei. Dacă variaţii mici ale datelor de intrare au tendinţa de a produce mari variaţii în soluţie, atunci problema este prost condiţionată. Un algoritm este numeric stabil dacă nu introduce o sensibilitate mai mare la perturbaţii decât cea proprie problemei. Pentru mulţi algoritmi de algebră liniară se poate arăta *stabilitatea înapoi*, adică se poate arăta că soluţia calculată este lângă soluţia exactă a unei probleme care provine dintr-o uşoară perturbaţie a problemei originale. Soluţia problemei uşor perturbate se află lângă soluţia problemei originale dacă problema este bine condiţionată. Aceasta este legătura dintre bine condiţionare şi stabilitatea numerică a unui algoritm. Acum, pentru un algoritm stabil numeric nu ne putem aştepta să rezolve o problemă prost condiţionată cu o acurateţe mai mare decât cea garantată de datele (parametrii) problemei, dar un algoritm instabil numeric poate furniza o soluţie catastrofală chiar pentru probleme bine condiţionate. Un software de calitate ţine seama de toate aceste dezvoltări teoretice implicând în plus artă şi imaginaţie.

Aşa după cum am spus mai sus, Turing a fost preocupat de a defini o măsură a cantităţii de muncă depusă pentru a rezolva efectiv o problemă utilizând numerele reale ca entităţi bine definite, iar operaţiile algebrice şi comparaţiile cu aceste numere ca unităţi a cantităţii de muncă necesară rezolvării unei probleme. Algoritmul fundamental de analiză numerică este algoritmul Newton pentru rezolvarea unui sistem de ecuaţii algebrice neliniare în corpul numerelor reale sau complexe $f(z)=0$:

Maşina Newton		
1.	Iniţializare:	z, $\varepsilon > 0$ suficient de mic
2.	Calcul:	Se calculează $f(z)$
3.	Test de oprire a iteraţiilor, Branch:	Dacă $\lvert f(z)\rvert^2 \geq \varepsilon$, atunci se continuă cu pasul 4, altfel se execută pasul 5.
4.	Shift:	$z \leftarrow z - f(z)/f'(z)$ şi se continuă cu pasul 2
5.	Ieşire:	$z^* = z$

Observăm că această maşină a lui Newton se prezintă ca un graf orientat cu cinci noduri. Motivaţia acestui algoritm vine direct din rezultatul clasic al lui Evariste Galois (1811-1832) care zice că *pentru polinoame de grad* $n \geq 5$ *rădăcinile nu se pot calcula prin formule închise care utilizează întregi şi operaţiile algebrice:* $+, -, \cdot, \div$, *exponenţiere şi extragere de radicali de la ordinul doi la ordinul* n. Ca atare suntem nevoiţi a utiliza metode iterative pentru rezolvarea problemelor. *Principiul fundamental în construcţia algoritmilor de analiză numerică* este de a exprima formulele de actualizare a diverselor variabile sau parametri sub forma:

Valoarea nouă = valoarea veche + o mică corecţie,

unde corecţia este calculată cu mare acurateţe. Dar, se vede imediat că trecerea la operaţii pe biţi distruge complet structura naturală a acestui algoritm.

3.1.2. Maşini peste inele sau corpuri

Toate aceste aspecte prezentate mai sus argumentează introducerea unui nou model computaţional prin definirea unei maşini peste un inel sau un corp.

Ideea este de a introduce o maşină înzestrată cu o serie de concepte proprii analizei numerice care să ne permită să combinăm cele două tradiţii, adică instrumentele şi rezultatele din teoria computabilităţii şi ştiinţa calculatoarelor cu instrumentele şi rezultatele proprii analizei numerice cu scopul de a înţelege natura şi complexitatea calculelor.

Presupunem că R este un inel comutativ sau un corp (ordonat) cu unitate. Atunci o *maşină* M *peste* R are următoarele proprietăţi [Blum, 2001]:

Maşina M are un *spaţiu de intrare* şi un *spaţiu de ieşire*, ambele R^∞ (reuniunea disjunctă a lui R^n, pentru $n \geq 0$, adică $R^\infty = \bigcup_{n \geq 0} R^n$. La nivelul superior maşina M este similară unei maşini Turing. Adică M are o *bandă infinită la ambele capete* împărţită în *celule* şi un *cap de citire-scriere* care poate accesa în acelaşi timp un număr de k_M celule contigue. Structura internă a lui M este aceea a unui *program*, adică un *graf orientat cu cinci tipuri de noduri*, fiecare dintre acestea cu propria lui operaţie şi saltul la nodul următor:

- Operaţia g_i asociată cu nodul de intrare i ia elementele $x = (x_1, \ldots, x_k)$ din spaţiul de intrare şi pune fiecare x_i, ($i = 1, \ldots, k$) în celule succesive pe bandă, începând cu cea din stânga. Nodul i are un unic nod următor $i' \neq i$.
- Fiecare nod computaţional η are asociat o transformare polinomială sau raţională $g_\eta : R^n \to R^m$ cu $n, m \leq k_M$. Date elementele $x_1, \ldots, x_n$ în primele n celule ale lui M, atunci operaţia asociată g_η pune $g_\eta^j(x_1, \ldots, x_n)$ în celula a $j-$a a lui M, $j = 1, \ldots, m$. Nodul computaţional η are un unic nod următor η'.
- Fiecare nod de branch ξ are asociată operaţia identitate. Acest nod are două noduri următoare: ξ'_L şi ξ'_R, în funcţie de elementul x_1 plasat cel mai la stânga din punctul de vedere al maşinii M. Dacă $x_1 = 0$ ($x_1 \geq 0$ dacă R este ordonat), atunci $\xi' = \xi'_R$. Dacă $x_1 \neq 0$ ($x_1 < 0$), atunci $\xi' = \xi'_L$.
- Fiecare nod de shift (deplasare) σ are asociată operaţia identitate şi acest nod are un unic nod următor σ'. Nodul de shift σ_R deplasează banda cu o celulă la dreapta. Asemănător, σ_L deplasează banda cu o celulă la stânga.
- Operaţia g_N asociată nodului de ieşire N goleşte conţinutul benzii în R^∞. Nodul N nu are nici un nod următor.

Observăm că intern maşina M este de fapt o maşină Newton. *Funcţiile calculabile* peste R sunt *transformări intrare-ieşire* Φ_M ale maşinii M peste R. Astfel, pentru $x \in R^\infty$, $\Phi_M(x)$ este definit dacă nodul de ieşire N este accesibil prin programul lui M când aceasta are la intrare elementul x. În acest caz $\Phi_M(x)$ este ieşirea $y \in R^\infty$.

Deşi la orice moment de timp maşina poate accesa un număr finit de celule, nodurile de shift permite maşinii să citească şi să opereze pe intrări din R^n pentru toţi n, astfel încât pe maşina M se pot modela algoritmi care sunt uniform definiţi pe spaţiul intrării, de orice dimensiune.

Observăm imediat că dacă R este Z_2, atunci ne plasăm în teoria clasică a computabilităţii şi a complexităţii. De asemenea, vedem că maşina Newton este în mod natural implementată pe o maşina peste R.

Cu acestea să vedem problema decidabilităţii peste corpul real sau complex.

Definiție. *O problemă peste* R *este o pereche* (X, X_{yes}), *unde* $X_{yes} \subseteq X \subseteq R^{\infty}$. X *constă din instanțierile problemei, iar* X_{yes} *din instanțierile yes.*

De exemplu pentru problema HN_R (Hilbert's Nullstellensatz) mulțimile de mai sus sunt:

$$X = \{ f = (f_1, \dots f_m) : f_i \in R[x_1, \dots, x_n],\ m, n > 0 \},$$
$$X_{yes} = \{ f \in X : \exists \zeta \in R^n,\ f_i(\zeta_1, \dots, \zeta_n) = 0, i = 1, \dots, m \}.$$

Sistemele finite de polinoame peste R se pot codifica sub formă de elemente ale lui R^{∞}, deci X este un subspațiu al lui R^{∞}.

Definiție. *O problemă peste* R, (X, X_{yes}), *este decidabilă dacă funcția caracteristică a lui* X_{yes} *în* X *este calculabilă peste* R.

Deci , în acest cadru putem formula probleme peste corpul real sau complex şi ne putem pune problema decidabilității. Utilizând, de exemplu, metoda lui Grete Hermann [1926] problema HN_R se poate uşor converti la una de decizie peste corpul numerelor complexe C şi deci HN_C este decidabilă peste C. Analog, utilizând teoria eliminării a lui Seidenberg [1954], se arată că problema HN_R este decidabilă peste corpul numerelor reale R.

În acest moment putem da un răspuns formal la problema lui Penrose privind mulțimea Mandelbrot M. Acum $X = R^2$, iar X_{yes} este chiar mulțimea Mandelbrot M.

Teoremă (Blum, Smale [1993]) *Mulțimea Mandelbrot* M *este nedecidabilă peste* $\mathbf{R}$.

Utilizând maşina M peste R se pot defini clasele de complexitate P_R şi NP_R peste R, în tradiția clasică, şi evident se poate introduce maşina polinomială în timp, precum şi *principiul de transfer*. Blum, Shub şi Smale [1989] demonstrează următoarea

Teoremă. $P_R = NP_R \Leftrightarrow HN_R \in P_R$ *pentru* $R = Z_2$, *corpul numerelor reale sau complexe, sau orice câmp ordonat sau neordonat.*

Deci HN_R este o problemă $NP-$*completă universală.* Cunoaştem că $HN_R \in EXP_R$ pentru R corpul real sau complex. Adică există algoritmi exponențiali în timp care decid solvabilitatea sistemelor polinomiale peste corpul real sau complex [Renegar, 1991, 1992], dar nu se cunosc algoritmi polinomiali în timp. Ca atare, pe lângă chestiunea clasică $P = NP$?, apar şi următoarele două: $P_R = NP_R$? şi $P_C = NP_C$? Înțelegerea complexității problemei Hilbert Nullstellensatz joacă un rol central în teoria complexității peste R.

3.1.3. Condiționare şi teoria complexității

Condiționarea unei instanțieri a unei probleme măsoară cum micile perturbații ale intrării alterează ieşirea. Acest concept furnizează o *legătură importantă între cele două tradiții*: *analiza numerică şi calculele științifice* pe de-o parte şi *teoria calculului* aşa cum apare în logică şi ştiința calculatoarelor pe de altă parte.

Dacă avem o problemă φ care la intrarea x furnizează ieşirea $\varphi(x)$, iar la intrarea „uşor" perturbată $x + \Delta x$ are ieşirea $\varphi(x + \Delta x)$, atunci, într-o normă corespunzătoare, raportul

$$\frac{\|\varphi(x+\Delta x)-\varphi(x)\|}{\|\Delta x\|},$$

sau raportul relativ

$$\frac{\|\varphi(x+\Delta x)-\varphi(x)\|}{\|\varphi(x)\|}\frac{\|x\|}{\|\Delta x\|}$$

reprezintă *condiționarea unei instanțieri a lui* φ. Dacă acest cât este mare, atunci instanțierea este prost condiționată, ceea ce implică mai multe resurse de calcul pentru a obține o soluție de acurateț e și deci pentru a reduce erorile.

a) Condiționarea sistemelor liniare.

Facem observația că pe calculatoarele actuale toate calculele sunt executate într-o submulțime $F \subset R$ și nu în toată mulțimea numerelor reale R. O proprietate caracteristică a aritmeticii în virgulă mobilă este existența unui număr $0<u<1$, „*unitatea erorii de rotunjire*", și a unei funcții $r: R \to F$, „*funcția de rotunjire*", astfel încât pentru toți $x \in R$, $|r(x)-x| \le u|x|$. Operațiile aritmetice din R sunt deci înlocuite cu versiunea lor „*rotunjită*" din F. De exemplu, înmulțirea lui $x \in F$ și $y \in F$ este $r(xy) \in F$.

În timpul calculelor aceste erori se acumulează și rezultatul final poate fi foarte departe de ceea ce ar trebui să fie. Ca atare, o primă grijă în ceea ce privește proiectarea algoritmilor este de a minimiza aceste erori. În privința erorilor algoritmii sunt analizați și comparați la fel cum sunt comparați din punctul de vedere al timpului de calcul. De altfel, această practică era cunoscută încă de pe timpul lui Gauss [*Theoria Motus*] (vezi [Goldstine, 1977] în studiul său asupra istoriei analizei numerice).

Pentru a studia cum erorile se acumulează în timpul calculelor executate de un algoritm oarecare este foarte convenabil să considerăm o situație ceva mai simplă, anume aceea în care erorile apar numai în datele de intrare . Adică o intrare $a=(a_1,\ldots,a_n) \in R^n$ este rotunjită la $r(a)=(r(a_1),\ldots,r(a_n)) \in F^n$ și apoi algoritmul este executat cu precizie infinită utilizând intrarea $r(a)$.

Fiindcă rezolvarea sistemelor de ecuații algebrice liniare este o problemă fundamentală, să vedem cum lucrează aceste concepte în acest caz. Fie deci sistemul $Ax=b$ unde A este o matrice inversabilă din $R^{n\times n}$ și $b \in R^n$. Dorim să vedem cum *soluția* x *a acestui sistem este afectată de perturbații în datele de intrare* (A,b). von Neumann și Goldstine [1947] și Turing [1948] au arătat că în acest studiu conceptul cheie este

$$\kappa(A)=\|A\|\|A^{-1}\|,$$

unde $\|A\|=\max_{\|x\|=1}\|A(x)\|$, iar $\|.\|$ este norma Euclidiană în R^n. Turing a numit $\kappa(A)$ *numărul de condiționare* al matricei A. Principalul rezultat pentru $\kappa(A)$ este că

$$\textit{dacă } \kappa(A)\frac{\|\Delta A\|}{\|A\|}<1, \textit{ atunci } \frac{\|\Delta x\|}{\|x\|} \le \frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|}+\frac{\|\Delta b\|}{\|b\|}\right).$$

Observăm că factorul

$$\frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}}$$

tinde la $\kappa(A)$ când $\|\Delta A\| \to 0$. În plus vedem că $\kappa(A)$ este atins în sensul că nici un număr mai mic nu va satisface inegalitatea de mai sus pentru orice A şi b. Deci, $\kappa(A)$ *măsoară cât de mult eroarea relativă în datele de intrare este amplificată în soluție* şi $\log^+(\kappa(A))$ măsoară *pierderea acurateței*[6]. Astfel, $\log^+(\kappa(A))$ furnizează o margine inferioară a celei mai defavorabile situații de pierdere a acurateței în rezolvarea sistemului.

Această noțiune de număr de condiționare, introdusă de Turing, care s-a inspirat din conceptul de derivată a lui Newton şi Leibniz, leagă cele două tradiții de care vorbeam mai sus mai ales când ne referim la complexitate. Menționăm că $\log^+(\kappa(A))$ reprezintă un *parametru intrinsec,* care înlocuieşte mărimea oarecum arbitrară a unei instanțieri a problemei ca lungimea în biți a intrării, în care spațiile în care acesta lucrează sunt cel real R^n sau complex C^n. De aceea un aspect important în teoria complexității îl constituie formularea şi înțelegerea condiționării.

Numărul de condiționare al unei matrice apare practic în toate analizele erorilor de rotunjire din analiza numerică şi are un rol determinant în condiționarea problemelor de programare liniară [Higham, 1996]. Când A nu este inversabilă numărul ei de condiționare nu este definit. Totuşi, putem extinde definiția lui admițând $\kappa(A) = \infty$ când A este singulară. *Matricele cu* $\kappa(A)$ *mic se numesc matrice bine condiționate, cele cu* $\kappa(A)$ *mare se numesc prost condiționate, iar cele pentru care* $\kappa(A) = \infty$ *prost puse.*

Se poate demonstra că mulțimea Σ a matricelor prost puse are măsura Lebesgue nulă în spațiul R^{n^2}. Distanța unei matrice A la această mulțime este intim legată de $\kappa(A)$. Următorul rezultat, foarte important, este cunoscut ca:

Teorema numărului de condiționare [Eckart şi Young, 1936]:
Pentru orice matrice reală $n \times n -$ *dimensională* A *are loc*

$$\kappa(A) = \frac{\|A\|}{d_F(A, \Sigma)},$$

unde d_F *măsoară distanța în* R^{n^2} *în raport cu norma Frobenius* $\|A\|_F = \sqrt{\sum a_{ij}^2}$.

b) Numărul de condiționare în programare liniară.

Un al doi-lea exemplu care ilustrează foarte bine legătura dintre condiționare şi complexitate este dat de programarea liniară. Menționăm imediat că programarea liniară este o extensie foarte naturală a sistemelor de ecuații algebrice liniare. La rândul ei programarea liniară se extinde la programarea semi-definită şi conică, care realizează o unificare între teoria şi metodele de optimizare şi metodele de calcul din teoria sistemelor dinamice. De aceea clarificarea semnificației legăturii dintre condiționarea unei probleme de programare liniară şi complexitatea ei reprezintă un rezultat major.

O problemă de programare liniară este definită de tripletul (A, b, c), unde $A \in R^{m \times n}$, $b \in R^m$ şi $c \in R^n$ definite peste corpul numerelor reale, care constă în: $max\{c^T x : Ax \le b, x \ge 0\}$. Tradițional, analiza complexității celei mai defavorabile situații pentru rezolvarea problemelor de programare liniară s-a definit în termenii elementelor n, m şi L, unde L este cunoscut ca „lungimea în biți" a datelor care definesc problema (lungimea

[6] $\log^+(x) = \log(x)$ dacă $x \ge 1$, altfel $\log^+(x) = 0$.

intrării), care de obicei este numărul de biți necesari pentru a specifica problema de programare liniară. Pentru datele de intrate (A,b,c) <u>întregi,</u> L se definește proporțional cu:

$$\log(\text{ cea mai mare valoare absolută a determinatului oricărei submatrice din } A)$$
$$+\log(\|c\|_\infty)+\log(\|b\|_\infty)+\log(n+d).$$

Deci, în contextul mașinii Turing, complexitatea problemei de programare liniară se definește în funcție de acest L. Pentru programarea liniară principalele abordări care au condus la algoritmi au următoarele caracteristici:

a) Algoritmul simplex al lui Dantzig (1914-2005):
- are o bună performanță computațională practică,
- complexitate exponențială (în n și m), dovedită de exemplul lui Klee și Minty [1972].

b) Algoritmul elipsoidului al lui Khachiyan (1952-2005):
- numărul de iterații: $O(n^2L)$,
- complexitatea computațională: $O(n^4L)$,
- performanțe computaționale proaste.

c) Algoritmul de scalare proiectivă al lui Karmarkar:
- numărul de iterații: $O(nL)$,
- complexitatea computațională: $O(n^{3.5}L)$,
- performanțe practice mult mai bune decât complexitatea teoretică demonstrată. (KORBX)

d) Algoritmi de punct interior:
- numărul de iterații: $O(\sqrt{n}L)$,
- complexitatea computațională: $O(n^3L)$. ([Anstreicher [1999]: $O(n^3L/\log(n))$]),
- performanțe practice foarte bune obținute cu pachete profesionale: CPLEX, LOQO, PCx etc.

Deci, în cea mai defavorabilă situație, într-o aritmetică exactă (adică peste corpul numerelor reale), algoritmul simplex execută un număr exponențial (în n și m) de iterații. Pentru (A,b,c) numere raționale, în modelul pe biți, algoritmul simplex este de asemenea exponențial în funcție de lungimea intrării.
Pentru numere raționale, în modelul pe biți, algoritmul elipsoidului este polinomial în funcție de lungimea intrării. Dar, ca algoritm peste corpul numerelor reale, adică într-o aritmetică exactă, algoritmul elipsoidului nu este finit. Același lucru este valabil pentru orice algoritm de punct interior.

Aceasta recomandă ca foarte naturală analiza complexității algoritmilor de programare liniară, definite peste corpul numerelor reale, în raport cu o *mărime intrinsecă* a intrării. Pentru aceasta este necesară introducerea unei măsuri a condiționării unui program liniar peste corpul numerelor reale. Inspirat de teorema lui Eckart și Young, pentru problema (A,b,c) Renegar [1995] definește numărul de condiționare $C(A,b,c)$. Mai mult, acesta ia în considerare *acuratețea de calcul* a soluției și precizează un algoritm de punct interior a cărui complexitate este $O(n^3\log(C(A,b,c)/\varepsilon))$ ce poate rezolva problema de programare liniară cu o acuratețe relativă ε sau determină că problema este infezabilă sau nemărginită. La fel ca în cazul sistemelor liniare o problemă de programare liniară este *prost pusă* dacă problema poate fi făcută atât fezabilă cât și infezabilă prin schimbări arbitrar de mici ale datelor de intrare (A,b,c).

Pentru problema $max\{c^Tx : Ax \le b\}$ Renegar definește numărul de condiționare primal $C_P^1(A,b)$ ca inversa distanței normalizate la problemele prost puse. Distanța problemei (A,b) la problemele prost puse este distanța la mulțimea programelor liniare prost puse în raport cu norma Frobenius. La fel se definește numărul de condiționare

$C_D^1(A,c)$ pentru problema duală. Atunci numărul de condiționare al programului liniar (A,b,c), $C^1(A,b,c)$ se definește ca maximum dintre $C_P^1(A,b)$ și $C_D^1(A,c)$. Cu acestea putem introduce aceste numere de condiționare ca:

(a) Dacă $Ax \le b$ *este fezabil, atunci:*

$$C_P^1(A,b) = \frac{\|A,b\|_F}{\sup\left\{\delta : \|\Delta A, \Delta b\|_F \le \delta,\ (A+\Delta A)x \le (b+\Delta b)\ este\ \ fezabil\right\}},$$

(b) Dacă $Ax \le b$ *este infezabil, atunci*

$$C_P^1(A,b) = \frac{\|A,b\|_F}{\sup\left\{\delta : \|\Delta A, \Delta b\|_F \le \delta,\ (A+\Delta A)x \le (b+\Delta b)\ este\ infezabil\right\}},$$

Se vede că $C_P^1(A,b) \ge 1.$ La fel se definesc numerele de condiționare pentru problema duală sau pentru celelalte probleme canonice de programare liniară de mai sus.

Este interesant de notat că numărul de condiționare pentru problemele de analiză numerică se definește ca sensibilitatea ieșirii la perturbațiile intrării și deseori acesta este în legătură directă cu distanța la problemele prost puse. Renegar, inversează această schemă prin definirea numărului de condiționare ca distanța la problemele prost puse și apoi demonstrează că acest număr de condiționare furnizează margini ale sensibilității ieșirii la perturbații ale intrării. Mai târziu, Cucker ți Peña [2002] iau în considerare *erorile de rotunjire* și propun un algoritm de punct interior pentru care arată că dacă programul liniar este admisibil, atunci timpul de calcul pentru a obține o soluție admisibilă este de ordinul:

$$O\left((m+n)^{3,5}(\log(m+n) + \log^+ C(A,b) + |\log \varepsilon|)^3\right).$$

Vedem imediat că deși conceptele de număr de condiționare, acuratețe de calcul și erori de rotunjire, care sunt proprii analizei numerice, ne ajută să facem o legătură între modurile de abordare combinatorial și continuu pentru analiza și proiectarea de algoritmi de programare liniară, totuși chestiunile de complexitate ale acestora *rămân deschise*.

În particular, problema de programare liniară peste corpul numerelor reale este de clasă P_R ? Sau, mai important (și mai greu) problema de programare liniară este *tare polinomială*. Adică pentru problema definită peste corpul numerelor reale există un algoritm polinomial în timp peste acest corp care să fie de asemenea polinomial în timp în raport cu lungimea intrării în biți pentru problema cu date de intrare numere raționale ?

3.2. Matematica experimentală

Problema dacă cunoașterea matematică este a priori sau a posteriori continuă să ocupe o poziție centrală în filosofia matematicii. Dezvoltarea calculatoarelor și utilizarea lor din ce în ce în mai multe domenii ale matematicii pune această întrebare într-o lumină cu totul nouă. Se pare că în matematică calculatorul, altfel spus informatica, are două funcții principale: *execută calcule numerice* și *descoperă noi domenii de cercetare*. De exemplu, *studiul atractorilor*, a *sistemelor dinamice haotice*, a *fractalilor* sau a *anumitor proprietăți ale numerelor*, studiu care se face nemijlocit cu ajutorul calculatorului, pune într-o poziție nouă problema experimentului în matematică.

Faptul că matematica este a priori sau a posteriori depinde dacă aceasta se transformă în mod real într-o știință experimentală sau quasi-experimentală și dacă ea migrează de la o știință axiomatic-deductivă la una empirică. Problema este importantă fiindcă foarte multe rezultate matematice sunt obținute pe cale experimentală și încă nu au o demonstrație teoretică în sens aristotelic. De exemplu, Gauss în jurnalul său notează: „*am rezultatul, dar nu cunosc încă cum să ajung la el*“. Și Hadamard, unul dintre cei mai mari matematicieni care au luat foarte în serios problema cunoașterii în matematică, notează: „*obiectul rigorii*

matematice este de a sancționa și legitimiza realizările intuiției, și niciodată nu a fost altceva". Datorită experimentelor computaționale în matematică se pot obține informații care pot sugera formularea de teoreme şi chiar demonstrația lor. Mai mult, experimentele computaționale pot prezenta exemple sau contraexemple pentru o teorie sau ipoteză. Toate acestea arată o anumită similitudine între rolul experimentului în matematică şi rolul acestuia în ştiințele Naturii. Totuşi, există o mare deosebire între acestea. Rezultatele experimentelor în ştiințele Naturii îndeplinesc un rol esențial în faza de justificare a teoriei. Fiecare teorie ştiințifică se confruntă în mod continuu cu rezultatele observării şi a experimentării. Nu este suficient ca o teorie să fie logic consistentă, rezultatele experimentelor sunt acelea care decid dacă ea este acceptată sau rejectată. Aceasta este calea în care teoriile ştiințifice sunt permanent corectate. Spre deosebire de aceasta în matematică lucrurile stau cu totul şi cu totul altfel. Dacă se dispune de o demonstrație corectă a unei teoreme, atunci nu există nici un experiment (corect inițializat) a cărui rezultat să fie inconsistent cu tezele susținute de teorema respectivă. Mai mult, dacă o teorie matematică este consistentă, atunci experimentele matematice în cadrul acestei teorii nu pot produce rezultate inconsistente cu aceasta.

Menționăm că există o mare deosebire între matematici şi ştiințele Naturii în ceea ce priveşte poziția față de experiment. Într-adevăr, când ceva merge prost în matematică, adică se identifică un contra-exemplu sau o contradicție, atunci anumite construcții matematice sunt eliminate din cadrul teoriei sau sistemului considerat. Pe de altă parte, în cadrul ştiințelor Naturii, când ceva nu se potriveşte, în sensul că apar eşecuri experimentale, atunci în cadrul sistemului sau teoriei respective se adaugă anumite construcții care să explice eşecul.

Rezultatul experimentului în matematică are deci un rol *secundar*, el serveşte pentru a ilustra o teorie dată. Clar, foarte multe fapte obținute cu ajutorul calculatorului (deci prin experimente matematice) nu au încă o demonstrație deductivă, şi pentru moment trebuie să ne mulțumim cu această justificare experimentală. Un exemplu în acest sens îl constituie formula închisă de predicție a numărului de iterații pentru algoritmul simplex în implementarea MINOS pentru programarea liniară, formulă care se bazează pe o interpretare foarte profundă a esenței algoritmului simplex [Andrei, 2004c]. Totuşi, aceasta nu înseamnă că trebuie să acceptăm aceste rezultate şi să nu continuăm căutarea pentru o demonstrație deductivă a lor.

Trebuie să precizăm aici poziția experimentului critic de tip Kantian sau Galilean. În acest caz acesta nu are o poziție chiar secundară, ci una *definitorie*, în sensul de a stabili şi preciza conținutul şi mai ales „*puterea computațională*" a unui rezultat teoretic. Deseori este posibil să construim algoritmi pentru rezolvarea unor probleme, pentru care putem demonstra complexitatea polinomială, ceea ce reprezintă un rezultat foarte important. Totuşi, implementări foarte îngrijite ale acestor algoritmi se dovedesc catastrofale din punctul de vedere al convergenței către soluție, necesitând un număr excesiv de mare de iterații şi deci timp de calcul. Un exemplu în acest sens îl constituie algoritmul elipsoidului, pentru programarea liniară, propus de Khachiyan. Implementările algoritmului elipsoidului nu s-au dovedit competitive cu programele care implementează metoda simplex. Aceasta a dus la reconsiderarea dezvoltărilor teoretice a complexității algoritmilor în sensul de a se obține algoritmi polinomiali de grad mic. Dar această idee nu s-a încetățenit decât numai în urma unor foarte intense experimente computaționale, care au precizat importanța gradului polinomului de complexitate. Aşa au apărut şi s-au dezvoltat metodele barieră de punct interior pentru programarea matematică pentru care s-a putut demonstra complexitatea polinomială de grad mic.

Înțelegerea şi explicarea a ceea ce observăm pe ecranul calculatorului se face prin formularea şi demonstrarea de teoreme matematice. Rezultă deci că răspunsul la întrebarea pusă la începutul acestei secțiuni este că *cunoaşterea matematică este a priori, în sensul că pentru a justifica valoarea de adevăr a unei teoreme singurul lucru necesar este de a da o demonstrație corectă a acesteia în cadrul unei teorii*. Faptul că matematica pare să fie o

cunoaştere a priori nu înseamnă că dezvoltarea cunoştinţelor matematice se face într-o manieră precis axiomatic-deductivă. Din contră, descoperirile matematice sunt condiţionate de o multitudine de factori, printre care un rol deosebit de important, poate cel mai important imediat după intuiţie, îl are experimentul matematic, care se defineşte în cadrul informaticii.

În acest sens se defineşte *matematica experimentală ca acea secţiune a informaticii care se ocupă cu codificarea şi transmiterea, în comunitatea matematică, a rezultatelor prin utilizarea tehnologiilor avansate de calcul pentru a realiza un experiment matematic, în sens Aristotelian, Baconian, dar mai ales Kantian sau Galilean, cu scopul de a explora structuri matematice, de a testa conjecturi, credinţe, de a sugera generalizări, precum şi a analiza rezultatele obţinute în urma acestor experimentări*.

Este clar că rezultatele descoperite experimental suferă de o anumită lipsă de rigoare, rigoare care este întotdeauna asociată matematicii, dar acestea (experimentele) în general furnizează o anumită percepţie şi înţelegere a naturii conceptului sau fenomenului matematic asupra căruia se fac experimentările respective, înţelegere care mai degrabă sau mai târziu se poate formaliza în teoreme şi teorii. Astfel, putem spune că *creaţional matematica este cunoaştere a posteriori*.

3.2.1. Experiment şi experienţă

Chiar dacă în vorbirea curentă sunt utilizate fără prea mare discernământ, există totuşi o mare deosebire între experiment şi experienţă. Experiment vine din latinescul „*experimentum*“ - o încercare, un test („*experiri*“ - a încerca, a testa) şi reprezintă *o acţiune sau un proces considerate pentru a descoperi ceva care nu este încă cunoscut sau de a demonstra (ilustra) ceva cunoscut*. Cu alte cuvinte, experimentul reprezintă un procedeu de cercetare, care în esenţă constă în provocarea sau punerea în operă a unui fenomen, în anumite condiţii bine definite, care permite studierea acelui fenomen. Pe de altă parte, experienţa, care vine şi ea din latinescul „*experientia*“, are o conotaţie mult mai complexă, putându-i-se asocia două mari înţelesuri. În primul rând aceasta poate avea sensul şi încărcătura unui experiment ca încercare, ca test, dar poate avea şi un sens mai larg ca *totalitatea cunoştinţelor dobândite de-a lungul timpului despre un anumit fenomen sau concept*. Cu alte cuvinte, experienţa are un caracter integrator, dar în acelaşi timp acţionează ca un filtru, mereu modificând cercul ignoranţei noastre.

După cum am văzut se cunosc patru tipuri de experimente: experimentul Aristotelic, Baconian, Kantian şi Galilean. În general, cercetarea naturii se referă la investigarea acelor situaţii care nu sunt nici simple şi nici evidente. Mai mult, o cercetare serioasă a naturii implică o argumentare pe baza evidenţei empirice date de experimente.

În acest context, experimentul Baconian este acela în care se declanşează o anumită situaţie pentru a „vedea ce se întâmplă“. Ideea este că putem înţelege cum se desfăşoară fenomenele, în acea porţiune a creaţiei pe care dorim să o cunoaştem, prin simpla acumulare a evidenţei diferitelor comportări ale ei. Informaţia astfel obţinută este adăugată la corpul factual de informaţii care constituie astfel experienţa asupra fenomenului respectiv. Experimentul Aristotelic constă în demonstraţia adevărului unei idei sau teorii formulate pentru un anumit scop. În cadrul experimentului Aristotelic se creează anumite situaţii sau se selectează acele observaţii care furnizează rezultate care confirmă ideea sau teoria respectivă.

Aceste experimente nu sunt critice. Ele se bazează pe concepţia inductivistă conform căreia pentru a fundamenta idei sau teorii trebuie să găsim acele circumstanţe care le verifică.

Spre deosebire de aceste două tipuri de experimente există şi aşa numitele experimente critice. Experimentul Galilean investighează o idee sau o teorie în situaţii critice, delicate, în care aceasta poate fi rejectată. Altfel spus, se caută sau se organizează acele situaţii care pun ideea sau teoria în dificultăţi majore, care arată slăbiciunile ideei sau teoriei experimentate. Experimentul Kantian este un experiment gândit, probabil cel mai bine caracterizat ca un exerciţiu de modelare într-o formă mentală sau simulată pe calculator, cu scopul de a explora implicaţiile ideii sau teoriei considerate. Ambele aceste tipuri de

experiment, Galilean şi Kantian, implică deducerea consecinţelor ideei sau teoriei, precum şi examinarea critică, empiric sau raţional, a rezultatelor experimentului.

Experimentul ca metodă de cercetare a naturii, ca metodă de investigare a profunzimilor lumii materiale, a fost consolidat de arabi, de unde a trecut în Europa unde un Petru Peregrinus sau Roger Bacon (1214-1294) au încercat o teoretizare a lui. Termenul de „*scientia experimentalis*“ a fost introdus de Bacon pentru a arăta importanţa experimentului în demersul şi efortul de descoperire a legilor. De fapt, toată ştiinţa, începând cu grecii şi până în zilele noastre, trecând prin Descartes, Galilei, Newton, Faraday, Maxwell, Stokes, Einstein, Plank, Heisenberg, Cournot, Walras, Say, Keynes etc. este un efort de descoperire şi formalizare a legilor.

O analiză critică, foarte serioasă, a ştiinţei moderne, începând de la Galilei şi Newton, a fost făcută de filosoful român Lucian Blaga (1895-1961). În lucrarea „*Experimentul şi spiritul matematic*“, acesta arată că în ştiinţa modernă accentul cade pe descoperirea legilor prin intermediul unor *cupluri metodologice* ca: *observaţia empirică este folosită în cuplu cu matematica; experimentarea se face în spirit matematic* etc. Altfel spus, se ajunge la adevăr, adică se obţine cunoaştere, numai printr-o îmbinare de metode sub formă de cupluri metodologice, dintre care cel mai important este *cuplul experiment-matematică*. Aceasta arată *poziţia dublă* a experimentului, ca exponent al empirismului, deductivului, faţă de matematică, ca exponent al raţionalismului, inductivului, în zona de constituire a legilor.

3.2.2. Algebra ca experiment gândit

Întotdeauna matematica pune spiritul în arc, într-o stare de necontenită căutare, fiind singura dintre discipline care face acest lucru într-un sistem logic în care, raţionând corect, pe orice drum s-ar merge, se ajunge la adevăr. Fundamental pentru matematică este că aceasta creează instrumente universale care prin generalitatea şi rigoarea lor pot fi utilizate în orice domeniu al activităţii umane. Toate ramurile matematicii au sau vor avea cândva o aplicabilitate concretă şi aceasta este dată de poziţia specială şi rolul pe care-l are algebra în cadrul celorlalte discipline matematice. De aceea am şi spus că ***informatica este algebră computaţională***. Descartes a fost primul care a afirmat că „*ştiinţele matematice sunt matematice deoarece toate se fundamentează pe algebră*“. Filosoful care a luat cel mai în serios problema a ceea „*ce putem cunoaşte*“ a fost, fără nici o îndoială, Kant. El a încercat să răspundă la această întrebare dificilă prin împărţirea cunoştinţelor noastre în două categorii. Există metoda experimentală care ne dă cunoştinţe detaliate asupra fenomenelor analizate şi metoda matematică. Matematica nu ne dă cunoştinţe detaliate despre un obiect sau fenomen particular, ci condiţiile generale care guvernează clasa căruia obiectul sau fenomenul îi aparţine.

Succesul ştiinţelor inginereşti, al economiei, biologiei, ecologiei etc. în sens Galilean, ştiinţe profund empirice cu multă „inginerie“ şi fenomenologie, este în întregime datorat introducerii algebrei în rezolvarea problemelor fundamentale asociate acestor domenii. Este adevărat că în domeniile menţionate mai sus pentru descrierea matematică a fenomenelor proprii lor utilizăm capitole de calcul diferenţial şi integral, de ecuaţii diferenţiale, de geometrie diferenţială, topologie, analiză complexă etc. Toate acestea duc în final la construcţia modelelor matematice asociate fenomenelor studiate, modele care se află în perspectiva infinitei asemănări cu realitatea. Dar algebra este cea care face posibilă realizarea experimentelor numerice controlate, care duc la experienţă şi de aici la teorie.

Principala proprietate a algebrei este aceea de *închidere*, în sensul că aplicând o operaţie pe o mulţime de obiecte dintr-o anumită clasă, obiectul rezultat aparţine acelei clase. Descartes a fost primul care a arătat că geometria se poate studia într-un sistem algebric, fiind închisă la cele patru operaţii aritmetice.

Sistemul de obiecte care este închis în raport cu anumite operaţii este un sistem de obiecte inventate, fictive, care sunt reale numai dacă această lume fictivă are o reflexie în realitate. În algebră, tehnica de bază, aşa cum a fost descoperită de Descartes, constă în a găsi

operații şi mulțimi de obiecte care sunt închise la aceste operații. Pentru a asigura închiderea trebuie ca aceste obiecte algebrice să fie definite în mod artificial. De exemplu, numerele se definesc ca naturale, raționale, reale, complexe etc. Spațiile sunt calificate ca spații vectoriale, Banach, Hilbert, Hausdorff, varietăți diferențiale etc. În algebră putem face descoperiri (formale) asupra proprietăților obiectelor algebrice deoarece acestea sunt artificiale. Ca atare, *algebra este o ficțiune, adică un experiment gândit, pur gândit*. De exemplu, când Boole a propus algebra claselor, el a definit-o închisă în privința a două operații: reuniunea şi intersecția. Reuniunea a două clase poate forma o clasă care aparține algebrei claselor. Dar, ce înseamnă intersecția a două clase care nu au elemente comune ? Boole a propus că orice clasă are clasa nulă ca subclasă. Dacă acceptăm acest artificiu, atunci algebra Boole funcționează. Fără el nu există închidere şi deci nici algebra Boole. Este clasa nulă o ficțiune ?, este ea reală ? Clar, clasa nulă este o ficțiune. Pare, deci, că un sistem algebric, la fel ca un roman, ca o poveste, este o ficțiune, dar o ficțiune numai în privința obiectelor sistemului care o defineşte, nu în toate privințele. Chiar dacă sunt studiate cu scopul de a înțelege realitatea cea mai concretă, obiectele algebrice nu sunt deci reale, nici în mod necesar realiste, ci fictive, aparținând unor lumi total diferite de a noastră. Ca atare, aceste obiecte sunt cel mai bine descrise ca experimente gândite.

Algebra are legătură cu realitatea înconjurătoare datorită similitudinii care există între operațiile investigate în algebră, operații în raport cu care se defineşte închiderea, şi operațiile utilizate în experimentele computaționale din lumea fizică reală. Altfel spus, modul prin care un sistem algebric este contingent cu realitatea nu este datorat „realității“ obiectelor fictive inventate pentru a permite închiderea algebrică, ci similitudinii care există între operațiile algebrei şi operațiile utilizate în experimentele reale.

Succesul modelării matematice se datorează deci trecerii a două tipuri de teste. Primul constă într-o cercetare sceptică foarte minuțioasă a experimentelor modelistului, ca şi cunoscător al fenomenologiei respective. A doilea test se referă la cercetarea amănunțită din punct de vedere algebric, ca experiment gândit, care astfel pune anumite restricții teoretice asupra modelului ca reprezentare matematică a creației. Altfel spus, o teorie a unei porțiuni oarecare a creației nu trebuie să treacă numai testele experimentale în sensul *adaequatio rei et intellectus* din domeniul respectiv, ci în acelaşi timp trebuie să fie *interpretabilă algebric*.

Pentru experimentator, orice porțiune a creației este deodată mister și revelație, deodată taină și descoperire. Există un ezoterism esențial care se referă la acel aspect al naturii care nu este accesibil în general, care prin definiție este inaccesibil, un rest, care pune modelul, ca reprezentare matematică a acelei porțiuni a creației, în perspectiva infinitei asemănări cu realitatea. Despre acest rest vorbea Kant. Partea cunoscută, care se dezvăluie prin efortul de citire a creației, trebuie să fie interpretabilă algebric. Numai atunci modelul ia ceva din esența fenomenului.

În concluzia acestei secțiuni, cred că progresul se realizează prin interpretarea realistă a matematicii şi ştiințelor naturii, precum şi prin interpretarea instrumentală a algebrei ca experiment gândit. Adică, nu trebuie să clasificăm obiectele cunoaşterii ca empirice şi matematice, ca inductive şi deductive, nici să le interpretăm holistic, aşa cum propune Willard Quine (1908-2000), ci mai degrabă în sens Kantian, să împărțim aceste obiecte în „*ştiințele matematice empirice ale naturii*“ şi obiecte „*algebrice gândite*“, obiecte care participă nemijlocit la definirea *experimentelor computaționale controlate care formează fundamentul informaticii.*

4. Concluzii

Informatica este un mod concret de realizare a experimentelor computationale, de fapt a matematicii experimentale. Esenta ei consta în *coborârea în computational a conceptelor matematice, algoritmizarea acestor concepte, studiul complexitatii algoritmilor asociati acestor concepte.* Privita în profunzimi, *informatica este algebra computationala*, care în materia ei consta în *transformarea unui concept într-un algoritm* si apoi *transformarea acestuia într-un program de calcul.*

Problemele fundamentale ale informaticii, ca activitate de mare intelectualism în care creația (intelectuală) ocupă o poziție centrală, se referă la: algoritmizarea conceptelor matematice, studiul convergenței algoritmilor, studiul complexității algoritmilor și a naturii calculului, precum și la realizarea experimentelor computaționale.

Teoria clasică a computabilității (a lui Turing), care s-a dovedit extrem de productivă, a furnizat fundamentele şi cadrul dezvoltării ştiinţei calculatoarelor. Totuşi dependenţa acesteia de 0 şi 1 este inadecvată pentru a susţine fundamentele calculului ştiinţific modern, în care algoritmul prototip este algoritmul Newton. Efortul intelectual în a defini calculul peste corpul numerelor reale constă în extinderea maşinii Turing pentru a îngloba maşina Newton. Astfel, o serie de concepte majore proprii analizei numerice - număr de condiționare, aproximare, erori de rotunjire, probabilitate – utilizate în teoria complexității ne permit să combinăm instrumentele şi tradițiile din ştiinţa calculatoarelor cu cele din analiza numerică pentru a înțelege mai bine natura şi complexitatea calculului. Aceasta este de fapt problema fundamentală a informaticii; *natura și complexitatea calculului.*

Informatica se instituie ca un *instrument major al cunoaşterii prelungind analiza conceptelor matematice din punctul de vedere al puterii lor computaționale.* Aceasta reprezintă un pas important în procesul de îngustare a ignoranței noastre. Totuşi, trebuie să remarcăm imediat faptul că în ciuda rezultatelor excepționale oferite de aceasta, întotdeauna va rămâne un rest pe care nu-l vom cunoaşte niciodată. Adică conceptele matematice, algoritmii asociați acestora şi programele care întrupează algoritmii nu vor fi cunoscute niciodată. Aceasta este condiția noastră. Ceea ce putem face este să ameliorăm percepția noastră asupra acestor elemente. Observăm că în efortul de înțelegere a complexității computaționale, ca problemă fundamentală a informaticii, am distrus maşina Turing şi în locul ei am pus o altă maşină care include maşina Newton. Aici se întâlneşte Turing cu Newton. Și acesta este mersul natural, ontologic, al cunoaşterii. Progresul nu se realizează prin construcția de teorii elaborate, ci prin dărâmarea teoriilor, în bunul mers al Naturii: *ea distruge pentru a construi, și aceasta fără sfârșit.*

Există vre-o justificare că nu vom avea niciodată acces total la cunoaştere, inclusiv la cunoaşterea conceptelor matematice ? Da. Dumnezeu când l-a creat pe Om din lut, şi nu am nici o ezitare în a accepta această aserțiune, fiindcă nu găsesc nimic plauzibil care să o înlocuiască, a spus: „*Să facem om după chipul și după asemănarea noastră*” [Biblia, 1.26]. Observăm că s-a adresat la plural. Dar, în înțelepciunea Lui l-a făcut numai după chipul Lui. Aşa încât Omul *ab initio* se plasează în perspectiva infinitei asemănări cu divinitatea. Ontologic, Omul întotdeauna este pe drumul cunoaşterii. Aceasta face ca el să aibă un acces limitat la cunoaştere, de orice natură ar fi ea.

Referințe

Andrei, N., (2004a) *Convergența algoritmilor de optimizare.* Editura Tehnică, Bucureşti, 2004.

Andrei, N., (2004b) *Teorie versus empirism în analiza algoritmilor de optimizare.* Editura Tehnică, Bucureşti, 2004.

Andrei, N., (2004c) *On the complexity of MINOS package for linear programming.* Studies in Informatics and Control, vol.13, no.1, 35-46, 2004

Andrei, N., (2005) *Probleme fundamentale.* Scrieri matematice, Vol.1. (manuscris), 2005.

Anstreicher, K.M. (1999) *Linear programming in L([n^3/ln(n)]L) operations.* SIAM J. Optimization, 9, 803-812, 1999.

Blum, L., (2001) *Computing over the Reals: Where Turing meets Newton.* AWM Noether Lecture, Joint Mathematics Meetings in San Diego, January, 2001.

Blum, L., F. Cucker, M. Shub și Smale, S., (1998) *Complexity and Real Computation.* Springer-Verlag, 1998)

Blum, L., and Smale, S., (1993) *The Gödel incompleteness theorem and decidability over a ring.* From Topology to Computation: Proceedings of the Smalefest, Hirsch, M., Marsden, J. and Shub, M. (Eds.), 321-339, 1990.

Blum, L., M. Shub și Steve Smale (1989) *On a theory of computation and complexity over the real numbers: NP-completness, recursive functions and universal machines.* Bulletin of the AMS, vol.21, no. 1, 1-46, July 1989.

Cook, S., (1971) *The complexity of theorem proving procedures.* Proceedings of the 3rd Annual ACM Symposium on Theory of Computing. 151-158, 1971.

Couturat, L., (1899) *La logique mathématique de M. Peano.* Revue de métaphysique et de morale, Paris, 1899.

Cucker, F. ți Peña, J., (2002) *A primal-dual algorithm for solving polyhedral conic systems with a finite-precision machine.* SIAM J. opt., 12, 522-554, 2002.

Dumitriu, A., (1966) *Soluția paradoxelor logico-matematice*. Editura Științifică, București, 1966.

Dumitriu, A., (1968) *Mecanismul logic al matematicilor*. Editura Academiei Române, București, 1968.

Dumitriu, A., (1974) *Philosophia mirabilis,* Editura enciclopedică română, București, 1974.

Dumitriu (1986) *Eseuri – Știință și cunoaștere, Alétheia, Cartea întâlnirilor admirabile*. Editura Eminescu, 1986.

Dumitriu, A., (1998) *Istoria Logicii, Vol.IV*, Ediția a III-a revizuită și adăugită, Editura Tehnică, București, 1998.

Eckart, C. și Young, G., (1936) *The approximation of one matrix by another of lower rank.* Psychometrika, 1, 211-218, 1936.

Gödel, K., (1931) *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*. Monatshefte für Mathematik und Physik, vol.38, pp.173-199, 1931.

Goldstine, H., (1977) *A history of numerical analysis from the 16th through the 19th century.* Springer Verlag, 1977.

Hermann, G., (1926) *Die Frage der endlich vielen Schritte in der Theorie der Polynomideale.* Math. Ann., 95, 736-788, 1926.

Heyting, A., (1934) *Mathematische Grundlagenforshungen. Intuitionismus, Beweistheorie*, Berlin, 1934.

Higham, N.J., (1996) Accuracy and Stability of Numerical Algorithms. SIAM, 1996.

Hilbert, D., (1900) *Mathematische Probleme.* Nachr. Ges. Wiss., Göttingen, 253-297, 1900.

Jin-Yi Cai, (2004) *Lectures in Computational Complexity*. Electronic book, Wisconsin University, (ww.cs.wisc.edu/~jyc/book.pdf).

Joyce, D.E., (2003) *Julia and Mandelbrot sets.* http://alrph0,clarku.edu/~djoyce/julia/explorer.html.

Karp, R.M., (1972) *Reducibility among combinatorial problems.* Complexity of Computer Computations, in Miller, R.E. and Thatcher, J.W. (Eds.), Plenum Press, pp.85-103, 1972.

Klee, V. și Minty, G.J., (1972) *How good is the simplex algorithm ?* Inequalities III, O. Shisha (Ed.) 159-175, Academic Press, 1972.

Ladrière, J., (1957) *Les limitations internes des formalismes*. Louvain-Paris, 1957.

Levin, L., (1973) *Universal search problems.* Problems Inform. Transmission, 9, 265-266, 1973.

Renegar, J., (1991) *Recent progres son the complexity of the decision problems for the reals.* DIMACS series in discrete mathematics and theoretical computer science, 6, 287-308, 1991.

Renegar, J., (1992) *On the computational complexity and the first order theory of the reals, Parts I-III.* Journal of Symbolic Comp., vol.13, 255-352, 1992.

Renegar, J., (1995) *Incorporating condition numbers into the complexity theory of linear programming.* SIAM J. Opt., 5, 506-524, 1995.

Seidenberg, A., (1954) *A new decision method for elementary algebra.* Ann. Of Math., 60, 365-374, 1954.

Turing, A.M., (1936) *On computable numbers, with an application to the Entscheidungsproblem.* Proc. London Math. Soc., 42, 230-265, 1936.

Turing, A., (1948) *Rounding-of errors in matrix processes.* Quart. J. Mech. Appl. Math., 1, 287-308, 1948.

Von Neumann, J. și Goldstine, H., (1947) *Numerical inverting matrices of high order.* Bulletin of the AMS, 53, 1021-1099, 1947.

Noiembrie 11, 2005

----------ooooo0ooooo----------