Uvod u probleme teorijskog računarstva

Ivan Radiček

Uvod u probleme teorijskog računarstva

Ivan Radiček
Institut za kozmologiju i filozofiju prirode
ivan.radicek@icpn.hr

Danas su računala u nekom obliku dio gotovo svakog električnog uređaja, od običnih stolnih računala do mobitela, "pametnih" satova i automobila. No, još i važnije, računala se u raznim oblicima danas koriste u mnogim prirodnim znanostima — od dokazivanja teorema u matematici do računalnih simulacija u fizici, biologiji i kemiji. Posebice zbog tog svojeg značaja u znanosti, nužno je istraživanje i promišljanje o računalima.

Temeljnim pitanjima računala bavi se teorijsko računarstvo, grana matematike kojoj je cilj precizno definirati i analizirati pojmove računanja, algoritma, programskih jezika i sl. Na prvi pogled se može činiti čudno da matematika ima veze s računalima, no da bi se bilo koja ideja bolje razumjela potrebno je shvatiti kontekst njezina nastanka.

Naime, računarstvo je nastalo prije pojave digitalnih računala kao matematički alat potreban za rješavanje nekih kontradikcija u samim temeljima matematike, tj. preciznije kao odgovor na pitanje što se uopće može izračunati? No kako se kao dominantna struja matematike isprofilirala ona logičko-formalistička, tako je i samo računarstvo ostalo zakopano u mulju formalizma iz kojega se nikada nije izvuklo.

Usprkos takvim skromnim temeljima, danas postoji tendencija mistifikacije računala od strane raznih interesnih skupina, koje uključuju i znanstvenu zajednicu. To se najbolje vidi na primjeru tzv. umjetne inteligencije, odnosno ideje da računalo može simulirati ljudski um i misao. Stoga je važno promotriti teoriju računarstva kritički, bez znanstvenih dogmi i mistifikacija.

1 Kriza temelja matematike

S pregledom začetka teorijskog računarstva krećemo na početku 20. stoljeća, u vremenu kada je matematika tražila svoje temelje te ih se pokušalo postaviti i izučavati unutar same matematike. Kod pokušaja gradnje takvih temelja javili su

se razni paradoksi, među kojima je jedan od najpoznatijih Russellov paradoks [1], koji je dobio ime prema Bertrandu Russellu.

Russellov bi se paradoks mogao ilustrirati na primjeru brijača koji živi u nekom gradu i brije samo i jedino stanovnike toga grada koji se ne briju sami — paradoks se uviđa kada se postavi pitanje brije li brijač samoga sebe? Formalno, taj paradoks se izražava kroz definiciju skupa R kao skupa svih skupova koji nisu članovi sami sebe. Ako je R član samog sebe tada to po definicije nije, a ako pak nije onda bi to po definiciji trebao biti.

Važno je napomenuti da se u to vrijeme matematika izučavala kroz teoriju skupova, a teorija skupova je temelj dominantne formulacije matematike uopće i danas. Stoga nije čudno da su se i paradoksi, kao gore diskutirani Russellov, promatrali baš kroz tu teoriju.

Prirodno se nameće i pitanje: ako je teorija skupova i danas temelj matematike, kako se onda riješio Russellov paradoks? Paradoks se samo zaobišao tako da se u nekom obliku dodalo pravilo da ne postoji *skup svih skupova*, pa samim time više nije moguće definirati skup iz paradoksa. [1]

Oko problema temelja matematike sukobile su se različite škole mišljenja među kojima su najpoznatije bile intuicionistička, predvođena filozofom i matematičarom Luitzenom E. J. Brouwerom i formalistička predvođena matematičarom Davidom Hilbertom.[2] Ovdje ne možemo i nećemo ulaziti u detalje navedenih škola, no spomenut ćemo, pošto je važno za daljnji tijek priče, da formalisti smatraju da su matematički teoremi samo posljedica manipuliranja formulama na strogo definiran način, a odbijaju se zamarati bilo kakvim sadržajem, filozofijom ili razmišljanjem o značenju matematike mimo te *igre formula*.

Žalosno je da takav formalistički pogled na matematiku prevladava i u današnjem obrazovnom sustavu, skoro na svim razinama. Tako se učenicima bez ikakvog konteksta prezentiraju razne gotove formule s kojima treba računati prema zadanim pravilima.

Jedna od posljedica takvog učenja je nezainteresiranost mlađih učenika za matematiku, jer je vide samo kao niz napamet naučenih formula za koje samo treba pogađati koja formula je ispravna za koji zadatak na testu ili u domaćoj zadaći. Isto tako, te matematičke formule se prezentiraju kao neupitne istine koje su pale s neba, a određeni (povlašteni) matematičari su na njih naišli i otkrili ih. Istina je da većina matematičkih teorija nastaje iz praktičnih potreba drugih prirodnih

[1] Formalno gledano dodao se novi aksiom u teoriju skupova kojim je ograničena vrsta skupova koji se mogu definirati. O aksiomima će biti nešto više rečeno niže.

[2] Možemo spomenuti i ranije logiciste na čelu s matematičarom, logičarom, i filozofom Gottlobom Fregeom.

znanosti te kroz svoj razvoj neprestano prolazi kroz niz iteracija i transformacija, a taj put matematičke teorije je često zanimljiviji od trenutno važećih formula. Na kraju, posljedica toga je i da se nigdje ne diskutira o problemima i ograničenjima matematike kao alata u izučavanju prirodnih znanosti.

Da bi se razriješila spomenuta kriza matematike, Hilbert predlaže program [2] s nizom zahtjeva i ciljeva koji bi trebali čvrsto definirati temelje matematike. Osnovni zahtjev je da se svaka matematička tvrdnja mora zapisati pomoću formalnog jezika, odnosno niza precizno definiranih simbola, a tim nizovima simbola se može manipulirati samo strogo definiranim pravilima.

Tada se neka matematička teorija definira na način se postave njezini aksiomi, tzv. evidentne istine koje se ne dokazuju, a bilo koji teorem unutar te teorije se dokazuje izvođenjem iz aksioma manipulacijom formula definiranim pravilima. Matematička tvrdnja je općenito neka matematička formula koja može biti istinita ili neistinita u kontekstu neke matematičke teorije, a teorem je tvrdnja dokazana iz aksioma te se onda smatra istinitom u kontekstu te matematičke teorije.

Primjer matematičke teorije ćemo prodiskutirati na pojednostavljenom primjeru teorije prirodnih brojeva [3] koju je, između ostalih, definirao Giuseppe Peano. Neki od aksioma s kojima možemo definirati prirodne brojeve su: (i) 0 je prirodni broj, (ii) za svaki prirodni broj n, $n+1$ je isto prirodni broj, (iii) za sve prirodne brojeve n i m, $n = m$ vrijedi ako i samo ako vrijedi $n+1 = m+1$, (iv) za niti jedan prirodni broj n ne vrijedi $n+1 = 0$, i (v) ako neka tvrdnja vrijedi za 0 i ako pretpostavljajući da tvrdnja vrijedi za n možemo dokazati da vrijedi i za $n+1$, tada ona vrijedi za sve prirodne brojeve (princip matematičke indukcije). Primjeri tvrdnji koje možemo pokušati dokazati su $x+(y+z) = (x+y)+z$ (asocijativnost), $1+\cdots+n = \frac{n\cdot(n+1)}{2}$, a nakon što ih dokažemo te tvrdnje možemo smatrati i teoremima. Podsjećamo, da bi neku tvrdnju dokazali, potrebno je počevši od aksioma i koristeći samo dozvoljene logične korake doći do željene tvrdnje.

Matematička se teorija smatra *konzistentnom* kada se ne mogu dokazati kontradiktorni teoremi, kao što je npr. $2+2 = 5$ za cijele ili prirodne brojeve. S druge strane, teorija se smatra *potpunom* ako se svaki ispravan teorem može dokazati unutar teorije. Između ostaloga, važni ciljevi Hilbertovog programa su bili dokazati da su matematičke teorije konzistentne i potpune, unutar same matematike, odnosno koristeći upravo definirana formalna pravila.

Hilbert je u svojem programu definirao još jedan cilj, koji je i najzanimljiviji za naše razmatranje, a to je *odlučljivost* (u originalu *Entscheidungsproblem* [4]), odnosno definicija *algoritma* koji bi za bilo koju matematičku tvrdnju izračunao (odlučio) *da* ili *ne*, ovisno o tome li se ta tvrdnja može dokazati iz teorema. Napomi-

njemo da je to bilo vrijeme kada pojmovi algoritma i računala još nisu bili definirani.

2 Pojam algoritma i izračunljivosti

Prvi udarac Hilbertovom programu zadao je Kurt Gödel sa svoja dva poznata teorema o nepotpunosti (engl. incompletness) [5]. U oba se teorema pretpostavlja konzistentna matematička teorija koja uključuje barem dio osnovne aritmetike. Tada teoremi kažu[3] da takva teorija: (i) ne može biti potpuna, odnosno postoje tvrdnje za koje se ne može dokazati ni da su valjane (istinite) niti da su nevaljane unutar te teorije, i (ii) ne može dokazati svoju konzistentnost.

Zanimljivo je da je Gödelov inicijalni cilj bio dokaz Hilbertovog programa, ali, iako to ne prihvaća cijela matematička zajednica, ovi teoremi su pokazali da program, kada traži formalizaciju te potpunu i konzistentnu aksiomatizaciju sve matematike, nije moguć.

No, ovo još nije riješilo pitanje izračunljivosti, odnosno toga što se može efektivno (ručno, mehanički) izračunati, ponajviše jer nije postojao konsenzus što to uopće znači. Prvi je Gödel 1933., uz pomoć Jacquesa Herbranda, kao model izračunljivih funkcija definirao tzv. *rekurzivne funkcije*, gdje se pomoću jednostavnih funkcija i operacija grade kompleksnije funkcije. Iako nećemo ulaziti u detalje rekurzivnih funkcija, ovdje je važno napomenuti da se pojam izračunljivosti od početka promatrao kroz pitanje *koje matematičke funkcije nad prirodnim brojevima se mogu izračunati?*

Nakon rekurzivnih funkcija, 1936. Alonzo Church predlaže λ-račun [6] kao model izračunljivih funkcija. λ-račun opisuje formalan sustav računanja baziran na operacijama apstrakcije (definiranja funkcije) i supstitucije (aplikacije funkcije) te se taj sustav može vidjeti kao preteča računalnih programskih jezika funkcijskog tipa.

Iste godine, neovisno o Churchovom rezultatu, Alan Turing definira svoj univerzalni stroj [7], danas poznat kao Turingov stroj, kao još jedan model izračunljivosti. Turingov stroj je opisan kao mehanički stroj koji prima ulazne podatke, ispisuje izlazne podatke (rezultat), posjeduje svoje unutarnje stanje (konfiguraciju) te u svakom koraku svojega rada, ovisno o ulaznim podacima i trenutnom stanju, mijenja svoje stanje i po potrebi ispisuje rezultat. Zanimljivo je primijetiti da se upravo na ovakvom modelu računala temelji arhitektura današnjeg računala, ako

[3] Ovi opisi su neformalne naravi, pošto potpuni opis zahtjeva dosta tehničkog žargona, no dovoljni za naša razmatranja.

kao ulazne podatke uzmemo primjerice tipkovnicu i miš, kao izlazne podatke ispis na monitoru, unutarnje stanje stroja njegovu memoriju, a opis rada algoritam ili računalni program.

Church i Turing su pokazali, svatko u svojem modelu, da nije moguće definirati algoritam koji bi računao ispravnost matematičkih tvrdnji te su time zadali i konačan udarac Hilbertovom programu. No, može se postaviti pitanje koji je od ovih modela izračunljivosti onaj pravi i kako znamo da se neće pojaviti neki novi u kojem bi se mogla definirati odlučljivost matematičkih tvrdnji?

Već je Turing uvidio da su njegov stroj i λ-račun ekvivalentni, a John Barkley Rosser je formalno pokazao da su sva tri modela ekvivalentna, odnosno da rekurzivne funkcije, λ-račun i Turingov stroj računaju potpuno identične funkcije nad prirodnim brojevima.

Pošto su se ovi modeli pokazali ekvivalentnima, a nije postojao novi model koji bi računao neki drugi skup funkcija, Church-Turingova teza kaže da oni predstavljaju model izračunljivosti, odnosno da izračunljive funkcije jesu one i samo one koje se mogu računati s ovim modelima. Danas se ta teza smatra manje-više istinitom te je više nitko i ne propitkuje.

Usput možemo spomenuti da se ova teza pokušala preslikati na fiziku i filozofiju. Glavna ideja je da se Priroda i ljudski um mogu smatrati sofisticiranim računalnim procesima koje onda računala mogu simulirati te se čak uvode pojmovi kao *digitalna fizika* i *digitalna filozofija*.[4] Tako je, na primjer, britanski fizičar David Deutsch 1985. iznio tezu [8] da bi univerzalno računalo moglo simulirati svaki fizikalni proces.

Već je na prvu jasno da su ovakve teze neznanstvene, jer ni jedna znanost ne smije padati pred dogmama i mišljenjem da je neka teorija ili ideja apsolutno razvijena. Isto tako je ovakav način mišljenja nazadan i s filozofskog gledišta. Već je grčki filozof Heraklit smatrao, a dalje je razvijeno kod Hegela, da je Priroda u stalnoj promjeni te se spoznaje kroz jedinstvo suprotnosti, nasuprot ovakvom mehanicističkom shvaćanju Prirode kao nekakvog stroja. Problematično je i naopako idealističko shvaćanje da se Priroda pokorava nekakvom prirodnom zakonu, a k tome još i zakonu modeliranom računalom.

Ovakvo mišljenje se u današnje vrijeme manifestira u pokušajima kreiranja umjetne inteligencije, odnosno svjesnog računalnog programa koji simulira ljudski um. Takvo nešto se čini teško zamislivo čak i redukcionističkim promatranjem ljudskog uma samo kao skupa neurona kroz koje prolaze električni impulsi, a mišljenja kao nekakve manifestacije tog elektro-mehaničkog procesa. Naime, mo-

[4] Za nešto više informacija vidi: https://en.wikipedia.org/wiki/Digital_physics.

žemo se pitati kako bi taj skup računalno simuliranih neurona odjednom postao svjestan? Znanstvenici i popularizatori umjetne inteligencije smatraju da takav skup neurona samo treba *"trenirati"* na ogromnoj količini podataka koji opisuju željeno ponašanje, a onda će se svijest manifestirati sama od sebe — vjerojatno kao kod Skynet sustava u popularnom serijalu filmova Terminator.

3 Formalna analiza računalnih programa

Nakon što su se osnove računarstva razvile iz potreba matematike, samo računarstvo je postalo dio matematike. Cilj te grane je ispitivanje svojstva računala i računalnih programa koristeći matematičke modele. Pri tome se računalnim programom može smatrati bilo koji formalno matematički način opisa računanja ekvivalentan λ-računu ili Turingovom stroju. Gotovo svaki današnji računalni programski jezik, s precizno definiranom semantikom, može se smatrati takvim te se za takve jezike i kaže da su Turing-potpuni (engl. Turing-complete).

Primjerice, za neki program možemo definirati matematičku tvrdnju koja kaže da kada taj program završi s izvršavanjem (računanjem) izračunati će broj 42, odnosno da će rezultat toga programa biti 42. Ako je taj program modeliran pomoću Turingovog stroja tada bi ta tvrdnja iskazivala da će Turingov stroj završiti s radom u nekom trenutku i da će na izlazu biti broj 42. Kada tu tvrdnju dokažemo tada ćemo reći da program ima svojstvo "*da uvijek računa broj 42*".

U praksi se najčešće ispituju svojstva koja opisuju koje rezultate program izračuna za određene ulazne podatke (točnost izračuna), koje operacije obavlja tijekom računanja (da ne postoje tzv. "bugovi"), i koliko je vremena (ili memorije) potrebno za računanje (performanse).

Može se postaviti pitanje zašto nam je zanimljivo matematičko dokazivanje svojstva programa, tj. zar ne možemo jednostavno izvršiti program na računalu i vidjeti koji će biti rezultat? Jedan problem je što program može imati previše mogućih ulaznih podataka pa nije moguće izvršiti program za sve moguće ulazne podatke. Ovdje treba obratiti pažnju da koji god broj ulaznih podataka ispitali ne možemo biti sigurni da neki neispitani slučaj ne opovrgava svojstvo koje ispitujemo. Drugi problem je što bi htjeli svojstva programa znati i prije pokretanja, pošto pokretanje programa može biti nepraktično jer, na primjer, uključuje lansiranje rakete u svemir. S druge strane ako svojstvo dokažemo precizno matematički tada znamo da ono vrijedi za sve moguće ulazne podatke i bez pokretanja programa.[5]

[5] Preciznije bi bilo reći da tada znamo da to svojstvo vrijedi unutar apstraktnog matematičkog modela. Kako će to svojstvo odgovarati kod izvršavanja na stvarnom fizičkom računalu ovisi o tome koliko

Jedan od problema na koji ćemo naići kod ovakvoga razmatranja je da računalni programi mogu biti kompleksni te je stoga ručno (na papiru) matematičko dokazivanje praktično nemoguće. Tada se, slično kao i Hilbert, možemo zapitati da li nam računala tu mogu pomoći, tj. da li postoji algoritam koji bi izračunao da li neko svojstvo programa vrijedi ili ne (iako već znamo da to nije moguće za općenite matematičke tvrdnje)?

Turing je pokazao da to nije moguće za jedno specifično svojstvo koje se još naziva i *halting problem*, odnosno: "*program će se izračunati u konačnom broju koraka*", tj. neće se izračunavati beskonačno dugo. Slično je Church pokazao da ne postoji algoritam koji bi odredio da li dva programa računaju istu funkciju, odnosno da će uvijek za isti ulazni podatak izračunati isti rezultat.[6] Da li ova dva rezultata, koja govore o vrlo specifičnim svojstvima programa, govore išta o općenitom slučaju, tj. o općenitim svojstvima programa?

Na ovo pitanje potvrdno je odgovorio Henry Gordon Rice kada je pokazao da se algoritam za bilo koje netrivijalno svojstvo[7] računalnih programa svodi na halting problem te stoga na takva pitanja nije moguće odgovoriti algoritamski. [9]

Za kraj ćemo spomenuti jedno otvoreno (neriješeno) pitanje u matematici, a koje se isto tako može promatrati i kao problem u samom računarstvu. Ovaj primjer ilustrira kako naoko jednostavan problem kojeg može shvatiti i osnovnoškolac može zadavati probleme i matematičarima i računalima.

Problem se sastoji od računalnog programa (ili funkcije) koji transformira prirodne brojeve. Ako je broj paran tada ga program podijeli s dva. Ako je pak broj neparan tada ga program množi s tri i dodaje jedinicu. Ova se transformacija ponavlja sve dok rezultat nije jedan. Formalno to možemo zapisati kao:

$$f(n) = \begin{cases} 1 & \text{ako } n = 1 \\ f(\frac{n}{2}) & \text{ako je } n \text{ paran} \\ f(3n+1) & \text{ako je } n \text{ neparan} \end{cases}$$

Tada, na primjer, možemo računati $f(12) = f(6) = f(3) = f(10) = f(5) = f(16) = f(8) = f(4) = f(2) = f(1) = 1$. Svojstvo koje na zanima kod ovog programa je da li, počevši od bilo kojeg prirodnog broja, ovo računanje završi u konačnom broju koraka s rezultatom 1? Problem je 1937. postavio, i naslutio da je odgovor pozitivan, matematičar Lothar Collatz, po kojem se ovaj problem i naziva. [8]

dobro apstraktni model opisuje ponašanje računala na zadanom svojstvu.

[6] Church i Turing su pokazali nepostojanje algoritma za odlučljivost matematičkih (logičkih) tvrdnji svođenjem upravo na ove probleme.

[7] Ovdje nećemo ulaziti u preciznu definiciju netrivijalnog svojstava.

[8] Collatz nije ovaj problem postavio kao problem računalnog programa, ali je takav pogled na problem zanimljiv iz perspektive našega razmatranja.

Direktnim računanjem pokazano je da ovaj program uvijek završava u jedinici za 2^{68} brojeva. No, do danas, iako se time bavio niz matematičara dugo vremena, nije matematički dokazano da to svojstvo vrijedi za sve cijele brojeve, niti je pronađena iznimka koja bi opovrgnula to svojstvo.

Literatura

[1] Bertrand Russell, University Press, 1903, Principles of Mathematics

[2] Neubegründung der Mathematik. Erste Mitteilung, David R. Hilbert, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg,1, 157-177,1922

[3] Giuseppe Peano, Torino, 1889, Arithmetices principia, nova methodo exposita (The principles of arithmetic, presented by a new method)

[4] Grundzuge der Theoretischen Logik, David Hilbert and Wilhelm Ackermann, 1928, Springer Verlag

[5] 173–198, 1931, 38, 1, Monatshefte für Mathematik, Über Formal Unentscheidbare Sätze der Principia Mathematica Und Verwandter Systeme I, Springer, K. Gödel

[6] http://www.jstor.org/stable/1968337, Alonzo Church, Annals of Mathematics, 2, 346–366, Annals of Mathematics, A Set of Postulates for the Foundation of Logic, 33,1932

[7] Turing, Alan M., Proceedings of the London Mathematical Society, 42, 230–265, On Computable Numbers, with an Application to the Entscheidungsproblem, http://www.cs.helsinki.fi/u/gionis/cc05/OnComputableNumbers.pdf, 2, 1936

[8] Deutsch, David and Penrose, Roger, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400, 1818, 97-117, 1985,

[9] H. G. Rice, Transactions of the American Mathematical Society, 2, 358–366, American Mathematical Society, Classes of Recursively Enumerable Sets and Their Decision Problems, 74, 1953, http://www.jstor.org/stable/1990888