

KAKVI SU LJUDI KOMPJUTORI: NEKA OTVORENA PITANJA KOMPJUTACIJSKOG PRISTUPA U SUVREMENOJ KOGNITIVNOJ ZNANOSTI

Pavle Valerjev

Odsjek za psihologiju Filozofskog fakulteta u Zadru

Obala kralja Petra Krešimira IV. 2

23000 Zadar

Sažetak

Cilj ovog preglednog rada jest da se objasni kompjutacijska ideja o umu. Ukratko su iznesene osnove matematičke teorije kompjutacije. Nadalje, opisani su argumenti za i protiv teze da je ljudski um moguće opisati kompjutacijski, odnosno algoritmima. Među suvremenim nedoumicama i razmimoilaženjima u pokušajima da se postave temelji za teoriju kognicije, u ovom radu naglasak je stavljen na preispitivanje matematičkog argumenta protiv kompjutacijske hipoteze o umu, koji se temelji na GödelovU teoremu o ne cjelovitosti i "halting" problemu te su pokazane veze tih matematičkih formulacija s paradoksima i nerješivim problemima. Ukratko su opisani i alternativni pristupi izučavanju kognicije: konekcionistički i dinamistički. Konačno, dani su prijedlozi za buduća istraživanja kojima bi se trebale ispitati oni aspekti ljudske kognicije koji se tiču rješavanja takvih zadataka koji su kritični s obzirom na opisane argumente.

Ključne riječi: kognicija; kompjutacija; Turingov stroj; algoritam; Gödelov teorem; halting problem; paradoksi; metakognicija

UVOD

Što je kognitivna znanost?

Herbert A. Simon, jedan od pionira i vodećih stručnjaka u kognitivnoj znanosti, naveo je (1991) da, po njegovu mišljenju, postoje četiri velika pitanja na koja se pokušavaju pronaći odgovori u okviru prirodnih znanosti. To su pitanja o prirodi materije, postanku svemira, prirodi života i prirodi uma. Fizičke znanosti se bave prvim dvama pitanjima, biološke se bave trećim navedenim, a kognitivne znanosti četvrtim pitanjem. Kako vidimo, ova pitanja prate glavne evolucijske korake u povijesti svemira.

Cilj kognitivne znanosti je da postavi teoriju kognicije u najopćenitijem smislu koja bi mogla opisati i sve različite pojavnosti inteligencije. Naravno, pi-

tanje je da li će se ovako visoko postavljen cilj ikad moći postići. Ali to ne sprečava kognitivne znanstvenike da istražuju inteligenciju na različite načine: konstruiranjem apstraktnih teorija i modela inteligentnih procesa koji su odvojeni od specifičnih fizikalnih ili bioloških spoznaja; proučavanjem organizmičke inteligencije (ljudske i životinjske); te proučavanjem računalne inteligencije, odnosno razvijanjem takvih programa koji mogu izvoditi inteligentne aktivnosti. Na inteligenciju se, dakle, ne gleda kao da je to isključivo ljudska osobina. Prije bi se moglo reći da su ljudski umovi jedna od pojava inteligencije koja je rezultat stotina milijuna godina djelovanja procesa evolucije. To ne isključuje postojanje i alternativnih inteligencija koje mogu biti proizvedene umjetno, na stroju, ili pak, rezultatom prirodnih procesa na nekom drugom mjestu (npr. izvanzemaljske inteligencije).

Kognitivna znanost je višedisciplinarna znanost. Njene središnje discipline jesu kognitivna psihologija, računalna znanost i umjetna inteligencija, neuroznanost, lingvistika, filozofija i antropologija. Sve navedene discipline (kao i brojne međudiscipline poput psiholingvistike, računalne lingvistike, neuropsihologije, filozofije psihologije itd.) nastoje proučiti razne fenomene kognicije i to na različitim razinama organizacije.

Premda su postojale i postoje teškoće u komunikaciji među znanstvenicima specijaliziranim u različitim područjima kognitivne znanosti zbog specifične terminologije i metodologije, takve su prepreke sve manje zbog jačanja svijesti o važnosti višedisciplinarnog pristupa (što uključuje stjecanje znanja iz drugih disciplina te razvijanje i primjenu višedisciplinarnih istraživanja), a što je možda i važnije, zbog postojanja razvijene temeljne paradigme koja se može opisivati na formalnom matematičkom jeziku i pod koju se mogu grupirati temeljna istraživanja iz područja. Ta se paradigma naziva kompjutacijskom teorijom uma.

Danas je kognitivna znanost jedna od najbrže razvijajućih znanosti, o čemu mogu svjedočiti brojne publikacije, te postojanje odgovarajućih odsjeka i studija na većini zapadnih sveučilišta.

Teorija kognicije: kompjutacijska teorija uma

Temelji koje su postavili razvoj simboličke logike, matematičke teorije informacije i komunikacije, kibernetika, teorija kompjutacije, teorija algoritama, teorija formalnih jezika, o prirodi, prijenosu i obradi informacija i simbola, snažno su utjecali na ideju o kognitivnom sustavu kao komunikacijskom kanalu koji prima, pohranjuje, obrađuje i daje informacije. Tijekom 50-ih i 60-ih godina 20. stoljeća ovaj pristup je u psihologiju uveo paradigmu obrade informacija. Shvaćajući "unutarnje" procese ljudskog uma kao procese manipulacije informacijama, konačno se ukazao put k njihovoj objektivizaciji i demistifikaciji jer su za to postojale dobre matematičke osnove. Osim razvoja novih temeljnih postavki, kao dijela kognitivne revolucije, došlo je i do razvoja nove metodologije

koja je u skladu s tim postavkama, a područja kognitivne psihologije postala su među najviše istraživanim u okvirima temeljnih eksperimentalnih psihologijskih istraživanja.

Tijekom 70-ih razvija se koncept koji se najčešće naziva kompjutacijska teorija uma. Ni jedna ideja u povijesti proučavanja uma nije potakla toliki entuzijazam, niti izazvala tolike kritike kao ideja da je um kompjutor. "Kognitivizam", odnosno kompjutacijska teorija uma, ne pretpostavlja samo da ljudski um može biti modeliran na računalu, već da na odgovarajućoj razini apstrakcije ljudi i jesu računala.

U kognitivnoj znanosti kognitivizam se obično razumijeva kao načelo da su računala manipulatori formalnih simbola. U ovoj karakterizaciji termin "simbol" se odnosi na bilo koji uzročno-posljedični unutrašnji znak za koncept, ime, riječ, ideju, reprezentaciju, sliku, strukturu podataka, ili drugi sastojak koji reprezentira ili prenosi informaciju o nečemu drugom. Riječ "formalno" odnosi se na oblik, formu, ili sintaksu, ali isto tako i na nešto nezavisno od semantičkih svojstava kao što su odnosi ili istinitost. "Manipulacija" se odnosi na činjenicu da je kompjutacija aktivan proces – nešto što se odvija u vremenu. Sada možemo kazati da je kompjutacija kao uključenost u aktivnu manipulaciju semantičkih ili intencionalnih sastojaka na način da ovisi samo o njihovim formalnim svojstvima.

Gore napisano objašnjenje predstavlja najtemeljniju postavku o prirodi kompjutacije. No, može se ići i dalje. Izuzetno značajan utjecaj u kompjutacijskoj teoriji uma ima matematička teorija kompjutacije koja je preuzeta iz teorijske računalne znanosti. Načela ove teorije uglavnom se temelje na poznatoj konstrukciji Turingova stroja koji predstavlja univerzalno računalo na kojemu počiva Turing – Churchova teza, koja kaže da je Turingov stroj kadar izvesti bilo koju operaciju koja se može algoritamski opisati. Turing je u svom originalnom radu (1936) uveo ideju implementiranja virtualnih arhitektura na druge, postojeće, tako da spomenuta teza nije ograničena brojem arhitektura. Formalna teorija Turingovih strojeva je relativno apstraktna, i koristi se uglavnom u teoretske svrhe: da se pokaže što može, a što ne može biti podvrgnuto kompjutaciji, da se demonstrira kompleksnost nekog rezultata (tj. koliko je rada potrebno da se riješi neka klasa problema), da se dodijeli formalna semantika programskim jezicima itd. Uzevši Turingove strojeve kao univerzalne manipulatore simbolima koji su kadri (ako su adekvatno definirani, tj. programirani) izvesti bilo koji algoritam, Turing je pretpostavio u svome drugom klasičnom radu (1950) da se i inteligentna aktivnost može opisati algoritmima koju onda može izvoditi stroj koji je Turing nazvao "strojem koji misli". Tada je postavljen i kriterij za proglašenje stroja inteligentnim, tzv. Turingov test: Ukoliko čovjek koji razgovara preko konzole s računalom ne može reći komunicira li s čovjekom ili sa strojem, tada je stroj prošao taj test.

U kognitivnoj znanosti termin se "kompjutacijski" često povezivao sa specifičnom klasom arhitektura: serijskim sustavima koji se temelje na relativno fik-

snim, simboličkim, diskretnim reprezentacijama visoke razine, kao što su sustavi aksiomatskog zaključivanja, sustavi reprezentacije znanja i sustavi temeljeni na znanju.

Zastupnici ovakvih, klasičnih simboličkih arhitektura su npr. poznati teoretičari kognitivne znanosti Fodor i Pylyshyn (1988, prema Pylyshyn, 1996) koji ističu superiornost takvih arhitektura pri baratanju sa sustavnošću, produktivnošću i složenošću misli visokih razina. Oni su razradili ono što se danas naziva klasično gledište na kompjutaciju i kogniciju. Prema tom gledištu računala i ljudski umovi imaju barem sljedeće tri razine organizacije:

- 1) Semantička razina (ili razina znanja) – Na ovoj razini objašnjavamo zašto ljudi ili adekvatno programirana računala čine određene stvari za koje možemo reći da su povezane na neke racionalne načine tako da djeluje kao da znaju što rade i što su njihovi ciljevi.
- 2) Simbolička razina – Semantički sadržaj znanja i ciljeva je kodiran pomoću simboličkih izraza. Tako strukturirani izrazi imaju dijelove od kojih svaki kodira neki semantički sadržaj. Kodovi i njihova struktura, kao i pravila po kojima su manipulirani spadaju u ovu razinu organizacije sustava.
- 3) Fizička (ili biološka) razina – Da bi cijeli sustav mogao funkcionirati mora biti realiziran u nekoj fizičkoj formi. Struktura i načela po kojoj fizički objekti funkcioniraju korespondiraju s ovom razinom.

OSNOVE TEORIJE KOMPJUTACIJE

Tradicionalno, tri središnja područja teorije kompjutacije jesu: teorija automata, teorija izračunljivosti (computability) i teorija složenosti, a zajedno su povezana pitanjem: Koje su temeljne mogućnosti i ograničenja računala? (Sipser, 1997).

Teorija automata bavi se definicijama i osobinama matematičkih modela kompjutacije.

Teorija izračunljivosti postavlja kriterije pomoću kojih se problemi mogu klasificirati na one koji su rješivi i na one koji nisu. Nerješivi problemi su oni za koje se ne može naći algoritam koji ih može riješiti.

Teorija kompleksnosti pokušava dati odgovor na pitanje: Što čini neke probleme kompjutacijski teškim, a neke druge lakim? Ona također teži klasifikaciji problema prema njihovoj kompjutacijskoj težini.

Teorija automata je vrlo blisko povezana s formalnom teorijom jezika. Da bismo izbjegli nedoumice oko terminologije, vrlo sažeto ćemo se osvrnuti na osnovne pojmove formalne teorije jezika.

Abeceda je bilo koji konačni skup oznaka ili simbola. Tako npr. možemo imati abecedu koja sadrži samo dvije oznake: nulu i jedinicu; i tada pišemo

$$\Sigma = \{0, 1\}.$$

String je bilo koji konačni niz simbola iz određene abecede. Tako npr. iz gornje abecede možemo dobiti slijedeće stringove: 0, 1, 01, 11, 01010101010101111, itd.

Jezik je skup stringova nad određenom abecedom. Svi stringovi koji pripadaju jeziku nazivaju se rečenice tog jezika. Ne može bilo koji string nad određenom abecedom biti član nekog jezika. Isto tako, nad istom abecedom možemo imati različite jezike. Koji će stringovi biti rečenice jezika, određeno je gramatikom jezika. Gramatika je uređena četvorka

$G = (V, \Sigma, P, S)$, gdje je

- V - konačni skup varijabli, odnosno nezavršenih oznaka. Te oznake mogu biti zamijenjene drugim nezavršenim oznakama ili elementima abecede.
- Σ - abeceda jezika koju čini konačni skup završenih oznaka, tj. terminala.
- P - konačni skup pravila ili produkcija koje opisuju način pretvorbe ili izvođenja jednih oznaka u druge.
- S - je varijabla koja je početni simbol.

Tako npr. ako imamo sljedeću gramatiku koju čine dvije varijable $\{S, A\}$, abeceda od dvije završene oznake $\{0, 1\}$, skup pravila koji čini tri produkcije $\{S \rightarrow A1, A \rightarrow 0 \mid 1\}$ i početno stanje S pišemo ju

$$G = (\{S, A\}, \{0, 1\}, \{S \rightarrow A1, A \rightarrow 0 \mid 1\}, S)$$

i možemo dobiti rečenicu 01. Ako krenemo od varijable S, i zamijenimo je dopuštenom nizom A1, u njemu tada varijablu A možemo zamijeniti terminalom 1 ili terminalom 0. Kako vidimo ovo je vrlo jednostavna gramatika pomoću koje možemo dobiti samo dvije rečenice: 01 i 11.

Sada možemo definirati jezik. Dakle, jezik je skup svih onih stringova nad određenom abecedom do kojih možemo doći ako krenemo od startnog simbola S i koristimo se samo pravilima koja su određena gramatikom toga jezika. Jezik je podskup skupa svih mogućih stringova koji mogu biti formirani iz dane abecede. Formalno to pišemo

$$L = \{\omega \in \Sigma^* \mid S \Rightarrow \omega\}$$

gdje je ω oznaka za string, Σ^* je skup svih mogućih stringova koji se mogu dobiti iz dane abecede, a $\Rightarrow$ je oznaka izvođenja u jednom ili više koraka. Kako vidimo gramatika nekog jezika je formalni sustav za prihvaćanje ili odbacivanje jezika, odnosno kriterij članstva za jezik.

Ove pojmove morali smo razjasniti, jer matematičko shvaćanje jezika je, kako vidimo, ponešto šire od onog svakodnevnog. Ali i ljudski jezici također padaju pod ovdje navedene formalizme. Ono zanimljivo što bismo mogli ovdje reći jest da je i skup obrazaca neuralne aktivnosti kojima barata mozak također jezik. Gramatika tog jezika su sve postojeće transformacije tih obrazaca u druge. Tako npr. u vidnom korteksu imamo pravila za prepoznavanje obrazaca pa za njih možemo reći da su dio te gramatike.

Sada možemo prijeći na automate. Automat je kompjutacijski model, odnosno pojednostavljeni i formalizirani model računala. U teoriji automata postoje različite klase automata koje se razlikuju kompjutacijskom snagom, odnosno klasom jezika koju prepoznaju. Za gore navedeni primjer jezika i gramatike može se konstruirati jednostavni automat iz klase konačnih automata. Formalna definicija konačnog automata je

Konačni automat M je uređena petorka $(Q, \Sigma, \delta, q_0, F)$, gdje je

Q - konačni skup stanja,

Σ - abeceda, konačni skup oznaka

$\delta: Q \times \Sigma \rightarrow Q$ je tranzicijska ili preoblična funkcija

$q_0 \in Q$ je početno stanje

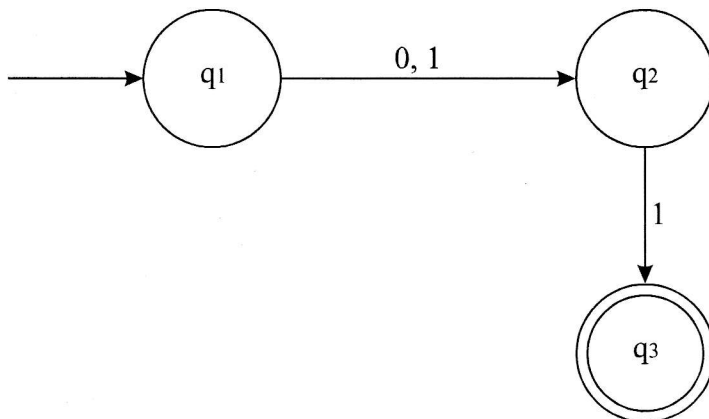
$F \subseteq Q$ je skup završnih stanja

Tada za jezik opisan u primjeru imamo konačni automat M_1 takav da je

$M_1 = (\{q_1, q_2, q_3\}, \{0, 1\}, \{(q_1, 0, q_2), (q_1, 1, q_2), (q_2, 1, q_3)\}, q_1, \{q_3\})$

Taj automat može se prikazati i dijagramom stanja koji je prikazan na Slici 1.

Slika 1. Dijagram stanja konačnog automata M_1



Na slici 1 vidimo da se opisani automat sastoji od tri stanja q_1 , q_2 , q_3 koja su označena kružnicama. Stanje q_1 je početno i to je označeno strelicom koja sa strane ulazi u njega. Završno stanje q_3 označeno je dvostrukom kružnicom. Kako se iz slike vidi, automat generira rečenicu jezika tako da prelaskom iz jednog stanja u drugo generira odgovarajući simbol jezika (npr. prelaskom automata iz stanja q_1 u stanje q_2 automat može generirati simbol 0 ili 1). Nadalje, prelaskom iz stanja q_2 u stanje q_3 može generirati jedino simbol 1. Kada se, konačno, automat nađe u završnom stanju q_3 , tada kompjtacija staje, a dobiveni string jezika može biti 01 ili 11.

Ovaj automat prepoznaje, odnosno prihvća stringove gore opisanog jezika L. Postupak koji automat obavlja naziva se kompjtacija. Tri uvjeta moraju biti zadovoljena za kompjtaciju:

- 1) Automat počinje u početnom stanju
- 2) Automat ide iz stanja u stanje prema tranzicijskoj funkciji
- 3) Automat prihvća svoj ulaz ako završava kompjtaciju u završnom stanju.

Jezici s pripadajućim gramatikama i automatima mogu se prema svojoj kompleksnosti klasificirati u Chomskyjevu hijerarhiju koja je dana u Tablici 1:

Tablica 1. Klasifikacija formalnih jezika s pripadajućim gramatikama i automatima.

Klasa jezika	Gramatika	Automat
Tip 0 Rekurzivno prebrojivi jezici	Neograničena gramatika $(V \cup \Sigma)^+ \rightarrow (V \cup \Sigma)^*$	Turingov stroj
Tip 1 Okolinski jezici	Okolinska gramatika $xVy \rightarrow x(V \cup \Sigma)^* y$	Linearno ograničen automat
Tip 2 Bezokolinski jezici	Bezokolinska gramatika $V \rightarrow (V \cup \Sigma)^*$	Potisni automat
Tip 3 Regularni jezici	Regularna gramatika $V \rightarrow V \Sigma^*$ $V \rightarrow \Sigma^*$	Konačni automati

Ovdje se vidi da svaka klasa jezika ima pridruženu, klasu gramatika koji ih generiraju i klasu automata koji ih prepoznaju. Na vrhu ove hijerarhije nalaze se jezici, automati i gramatike tipa 0, koji mogu obuhvatiti sve jezike koji su hijerarhijski niže. Svaka od ovih klasa ima određeni stupanj složenosti. Ta složenost može se prikazati dopuštenim produkcijama gramatike, odnosno tranzicijskim funkcijama automata i tipovima stringova koje oni prihvaćaju. Uz svaku od pripadnih gramatika dopisane su i karakteristične dopuštene produkcije, te se lijepo može vidjeti kako se njihova složenost povećava u svakoj novoj klasi.

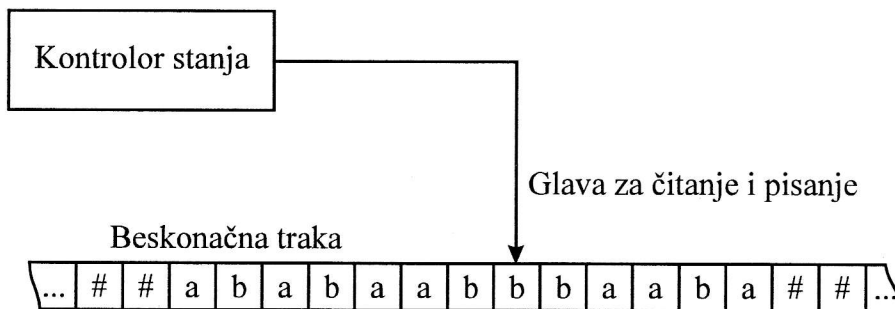
Na ovom mjestu neće biti detaljnije opisane sve karakteristike svake od ovih klasa jezika. Ono što je trebalo biti prikazano jest da postoje matematički formalizmi koji opisuju kompjutaciju, te da se kompjutacija može podijeliti u klase različite složenosti.

Inače, prikazani automati imaju svoju primjenu i u praksi. Većina jednostavnijih automatskih strojeva, poput automatskih vrata, mogu se prikazati konačnim automatima. Potisni automati i bezokolinske gramatike koriste se za opis sintakse programskih jezika, a okolinske gramatike koriste se za opis prirodnih jezika. Obje vrste imaju značajnu primjenu u lingvistici. Turingovi strojevi, koji su najsnažniji automati nad diskretnim simbolima, izuzetno su važni za teoriju algoritama, razumijevanje računala i umjetne inteligencije, matematike i logike, te ljudskog uma.

Detaljni pregled teorije automata može se naći u nekom od udžbenika (npr. Hopcroft i Ullman, 1979; Matuszek, 1996; Sipser, 1997).

Turingov stroj

Alan Turing je konstruirao zamišljeni automat u svom klasičnom radu (1936), koji je ubrzo nazvan Turingovim strojem. Neformalno se Turingov stroj opisuje kao automat koji se sastoji od konačnog kontrolora stanja, glave za čitanje simbola s trake, te trake koja je beskonačno duga i podijeljena na odsječke, takve da u svaki stane po jedan simbol. Na početku, traka sadrži samo ulazni string, a svugdje drugdje je prazna. Glava stroja može čitati simbol s trake, izbrisati simbol, upisati simbol, promijeniti unutarnje stanje i pomaknuti se za jedan odsječak lijevo ili desno. Na Slici 2 dan je shematski prikaz Turingova stroja.



Schema Turingova stroja

Kompjutacija se izvodi na sljedeći način. Stroj čita početni simbol stringa. Stroj se nalazi u početnom stanju. Nakon što je pročitao taj simbol, ovisno o svojoj tranzicijskoj funkciji on može ostaviti taj simbol, izbrisati taj simbol, upisati neki drugi simbol na njegovo mjesto, zatim promijeniti svoje stanje (ili ne), te konačno pomaknuti glavu lijevo ili desno. Tada ponavlja postupak s novim simbolom. Krećući se preko stringa na traci lijevo-desno neograničen, ali konačan broj puta, te mijenjajući taj string, stroj proizvodi svoj izlaz. Kompjutacija se završava kada se stroj nađe u završnom stanju. Postoje dva završna stanja: ono koje prihvaća ulazni string i ono koje ga odbacuje. Npr. možemo konstruirati Turingov stroj koji će ispitivati da li ulazni string ima podjednaki broj slova a i slova b . Tom stroju možemo zadati različite stringove koji se sastoje od ova dva slova. Stroj će uvijek stati u prihvatnom završnom stanju ako u stringu postoji jednak broj a i b , odnosno uvijek će stati u odbacujućem završnom stanju ukoliko taj broj nije jednak.

Formalna definicija Turingova stroja je:

Turingov stroj je uređena sedmorka $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{prihvatno}}, q_{\text{odbacujuće}})$, gdje je:

Q - konačni skup stanja,

Σ - ulazna abeceda

Γ - abeceda trake, takva da je $\#$ (prazni simbol) $\in \Gamma$, i da je $\Sigma \in \Gamma$,

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, D\}$ je tranzicijska funkcija, a L i D su oznake smjera

q_0 - početno stanje, $q_0 \in Q$

$q_{\text{prihvatno}}$ - završno prihvatno stanje, $q_{\text{prihvatno}} \in Q$

$q_{\text{odbacujuće}}$ - završno odbacujuće stanje, $q_{\text{odbacujuće}} \in Q$

U tranzicijskoj funkciji sadržane su sve dopuštene tranzicije, tj. preoblike. One se mogu zapisati kao uređena petorka (trenutno stanje, učitani simbol, simbol za upis, smjer pomaka, novo stanje).

Na primjer: $(q_1, a, \#, D, q_2)$

Sâm stroj, kao i mehanizam kompjudacije je vrlo jednostavan. Dakle, stroj u određenom stanju čita neki simbol. Zatim pronade odgovarajuću tranziciju, koja mu odredi simbol za upis, novo stanje i smjer pomaka glave. Skup svih ovih preoblika osim što čini tranzicijsku funkciju, čini i cjelokupni programski kod odgovarajućeg stroja. Ipak, treba odmah naglasiti Turingovu genijalnost, jer on je pomoću ovako jednostavnog mehanizma uspio pronaći hipotetski uređaj izuzetne kompjudacijske snage.

U stvarnosti Turingovi strojevi ne postoje u smislu da ih netko proizvodi. To su matematički formalizmi koji nam služe da proučavamo mogućnosti kompjudacije. Odgovarajuće programiran Turingov stroj ima dovoljnu kompjudacijsku snagu da može izvesti bilo koju operaciju koju može izvesti bilo koje računalo.

Algoritmi i teorija algoritama

Algoritam je recept, metoda ili tehnika da se nešto uradi. Osnovna karakteristika algoritma jest da je načinjen od *konačnog* skupa jednostavnih pravila ili operacija koje su nedvojbene i jednostavne za slijeđenje, odnosno definirane i efektivne. Tako određene operacije može izvoditi računalno. Ipak, veza s računalima nije neophodna. Ukoliko osoba opremljena olovkom i papirom može izvesti navedene operacije, tada one tvore algoritam.

Poznati primjer je Euklidov algoritam pomoću kojega se može naći najveći zajednički djelitelj dvaju broja m i n .

Korak 1. Za dva zadana pozitivna cijela broja, neka m bude veći od n ih, a n manji.

Korak 2. Podijeli m s n . Ostatak dijeljenja neka bude r .

Korak 3. Ako je $r = 0$, tada stani. Najveći zajednički djelitelj je n .

Korak 4. Inače, stara vrijednost od n postaje m , a vrijednost od r postaje n . Idi na korak 2.

Svi računalni programi su algoritmi. Pretpostavka kompjutacijske teorije uma je da je i mišljenje algoritam ili još bolje, rezultat mnogih algoritama koji simultano rade.

Preciznija definicija algoritma od gore navedene jest da je algoritam nedvosmislena, precizna lista jednostavnih operacija koje su mehanički i sustavno primijenjene na skup simbola, znakova ili objekata (Dietrich, 1999). Početno stanje algoritma je ulaz, a završno je izlaz. Operacije korespondiraju s *prijelazima stanja*, gdje su stanja konfiguracije simbola, koje se mijenjaju u skladu s primjenom operacija na njih. Skoro sve kod algoritama je konačno: lista operacija, svaki znak, ulaz, izlaz te skupovi simbola između ulaza i izlaza, premda teoretski, ulazni, izlazni, te među-skupovi simbola mogu biti i kontinuirani. Algoritam, dakle, opisuje proces u kojemu sudjeluju znakovi i taj proces se naziva kompjutacija. Taj proces je u svakom trenutku u određenom stanju, te prelazi (ili ne prelazi) u neko drugo stanje.

Algoritam je također i unutarnja definicija izračunljive funkcije. Vanjska definicija izračunljive funkcije dana je skupom ulaza i izlaza. Unutarnja definicija opisuje i *kako* je funkcija izračunata, a ne samo *što* je ta funkcija. Ova veza s izračunljivom funkcijom je ključna. Funkcija F je izračunljiva ako, i samo ako se može opisati kao algoritam. Također je i zanimljiva veza između vanjske i unutarnje definicije funkcije: za svaku vanjsku definiciju funkcije postoji beskonačno različitih unutarnjih definicija te funkcije, odnosno, beskonačno algoritama. Dokaz za to je činjenica da se uvijek može konstruirati novi, duži algoritam dodavanjem operacija koje, u načelu, ne čine ništa. Naravno, u praksi se najčešće traži kanonički algoritam, tj. najkraći i najefikasniji.

Izračunljiva je ona funkcija čiji su izlazi producirani Turingovim strojem na temelju njezinih ulaza. Prema Churchovoj tezi sve izračunljive funkcije mogu biti izračunate Turingovim strojem. Dobar način da se ovo razumije jest da kažemo da Turingov stroj iscrpljuje značenje pojma izračunljivosti. Nisu sve funkcije izračunljive, što znači da se sve funkcije ne mogu opisati algoritmima. Ovo je bilo jedno od najvećih otkrića matematike 20. stoljeća. Do njega su praktički istovremeno i nezavisno došli Turing, Church i Kleene 1936. godine (Dietrich, 1999).

U vezi s algoritmima treba se osvrnuti na još nekoliko stvari:

- 1) Matematičari i informatičari ponekad ograničavaju definiciju pojma algoritam. Oni ga uže definiraju - kao efektivnu proceduru koja uvijek stane (što znači da daje određeni izlaz).
- 2) Veza između algoritma i kompjutacije je vrlo bliska. Na algoritam se može gledati i kao na opis kompjutacijskog uređaja. Kada je algoritam implementiran na standardno računalo, tada on postaje stvarni kompjutacijski uređaj koji se naziva i virtualni stroj. Virtualni stroj je stroj koji čini ono što algoritam specificira. On postoji na nekoj višoj razini od stroja na kojemu je algoritam implementiran. Npr. obrađivač teksta je virtualni stroj koji egzistira na hardverskom stroju na kojemu je implementiran. Notacija virtualnog stroja je vrlo važna u kognitivnoj znanosti jer nam omogućava da podijelimo proučavanje uma na razine, s neurokemijom u temelju (ili blizu njega) i s kognitivnom psihologijom blizu vrha. Na svakoj razini koriste se različite metode i terminologija. Jedna od važnih činjenica u svezi s virtualnim strojevima je da ni jedan stroj nije važniji od drugih u toj hijerarhiji. Teorije svih razina potrebne su za cjelokupno i istinsko razumijevanje uma.
- 3) Prema kompjutacijskoj teoriji kognicije, kognitivni se procesi mogu opisati algoritmima. To znači sljedeće: ti algoritmi ne moraju biti kanonički, tj. najjednostavniji. Važno je da su efikasni. Naime, um nije a priori dizajniran prema nekim racionalnim pravilima, već je rezultat procesa evolucije.
- 4) Kognicija se ponekad preusko shvaća kao skup procesa koji se odnose na to kako i o čemu mislimo. Drugim riječima, kao ono o čemu smo svjesni dok mislimo. No, to je samo vrh ledene sante kognicije. Kogniciju čine svi oni automatski kompjutacijski procesi (algoritmi) čijim pozadinskim djelovanjem nastaju, između ostalog, i svjesne reprezentacije. Zbog istog se razloga npr. percepciju neopravdano smatra "nižim" kognitivnim procesom. Percepcija je naizgled jednostavna jer se odvija (na našu sreću) uglavnom automatski i bez napora. U stvarnosti, u podlozi percepcije nalaze se vrlo složeni kompjutacijski procesi.

Ono što je važno napomenuti u svezi s Turingovim strojevima jest pojam univerzalnog Turingova stroja. To je stroj koji može kao ulaz uzeti bilo koji

drugi Turingov stroj i ulaz za taj stroj, te obaviti kompjutaciju i dati izlaz koji bi dao taj Turingov stroj. Ovo je moguće zato što svaki Turingov stroj mora imati definirane parametre (stanja, abecedu, tranzicijsku funkciju), a oni se mogu upisati u obliku stringa. Sada dolazimo do pojava koje postaju vrlo zanimljive. Takav Turingov stroj može uzeti i samog sebe kao ulaz i ulaz za takav stroj (opet samog sebe itd. ... ili neki drugi Turingov stroj i njegov ulaz) i to ne jednom, nego bilo koji željeni, ali konačni broj puta.

Ovo i nije tako neobično. S ovakvim se pojavama profesionalni programeri susreću svakodnevno. Npr. moguće je napraviti takav algoritam koji kao dio svoje kompjutacije uzima i koristi samog sebe. Ili, nerijetka je pojava da se kompajler (prevodilac) za neki programski jezik programira u tom programskom jeziku samom.

Opisana se pojava zove rekurzija. Rekurzija u najopćenitijem smislu jest samo-referentost, odnosno kada nešto poziva u svojoj strukturi samog sebe, ili dijelove sebe. Douglas R. Hofstadter u svojoj vrlo inspirativnoj knjizi *Gödel Echer, Bach: An Eternal Golden Braid* (1980), koja je izvršila veliki utjecaj na proučavanje prirode uma, piše da je "... (rekurzija) gniježđenje i varijacije na gniježđenje. Ovaj pojam je vrlo širok. (Priče unutar priča, filmovi unutar filmova, slike unutar slika, ruske lutke unutar ruskih lutki (pa čak i komentari u zagradama unutar komentara u zagradama!) ...)." (str. 127). To su samo neki od jednostavnijih primjera. Rekurzije se javljaju posvuda: u matematici imamo teoriju rekurzivnih funkcija, u umjetnosti imamo Echerove slike i Bachove kontrapunktove. Rekurzije su česte i u prirodi. Od elementarnih čestica kada kod kretanja jednog elektrona nastaju i nestaju virtualni fotoni i antielektroni, a njihovim kretanjem nastaju i nestaju opet novi, pa do genetskog koda koji rekurzivno, koristeći sebe i bjelančevine gradi cijeli organizam, a ako uključimo i razmnožavanje - i vrstu.

Rekurzivno prebrojivi skupovi su oni skupovi koji mogu biti generirani iz početnih točki (aksioma) ako na njih ponavljano primjenjujemo neka pravila. Skup prirodnih brojeva je rekurzivno prebrojiv skup. Početna točka od koje se kreće je brojka 1, a primjenom funkcije neposrednog sljedbenika dobivamo sve ostale brojeve. Takvi skupovi su beskonačni, ali prebrojivi, u smislu da se do bilo kojeg elementa može doći u određenom broju koraka (za razliku od skupa realnih brojeva).

Cijela ova digresija bila je potrebna da se objasne rekurzivno prebrojivi jezici. Svaki rekurzivno prebrojiv jezik ima beskonačan broj stringova koji se mogu generirati ponavljanom primjenom pravila na početni string.

Turing - Churchova teza

Matematičari Alan Turing i Alonzo Church dokazali su 1936. godine nezavisno jedan od drugoga da su algoritmi rekurzivne funkcije i da Turingov stroj može izvoditi bilo koji algoritam. Dakle, Turingov stroj može izvoditi bilo

koju kompjutaciju koja se temelji na manipulaciji diskretnih simbola. Rezultat takve kompjutacije su rekurzivno prebrojivi jezici.

Ova teza imala je golem utjecaj na teoretičare kognicije i uma. Ona je potakla brojne rasprave i polemike koje i danas traju. Osnovno pitanje je: Je li ljudski um Turingov stroj? Naravno, ovo ne treba shvatiti doslovno. Nitko u glavi ne nosi beskonačnu traku. Preformulirano, pitanje bi bilo: Mogu li se operacije uma izvesti (simulirati) Turingovim strojem?

Konačnog odgovora na ovo pitanje još nema. Kasnije ćemo razmotriti argumente i protuargumente. Za sada možemo reći da ako se kompjutacija uma može opisati algoritmima, onda nedvojbeno znači da ju može izvoditi i Turingov stroj.

U vezi s mogućnošću simuliranja ljudskog uma na stroju, u umjetnoj inteligenciji postoje dva pristupa (Russell i Norvig, 1995). To su:

- 1) Jaka umjetna inteligencija – zastupnici ovog pristupa smatraju da strojevi mogu biti svjesna inteligentna bića u punom smislu.
- 2) Slaba umjetna inteligencija – prema ovom pristupu, strojevi mogu samo uspješno simulirati ljudske inteligentne aktivnosti i to ne u potpunosti. Uvijek će ostati stvari koje strojevi neće moći napraviti.

Ograničenja kompjutacije: paradoksi

Jedan od prvih logičkih paradoksa bio je Empedoklov paradoks koji je kasnije postao poznat i kao sofizam lašca, a on glasi:

Svi Krećani lažu.

Paradoks je u tome što je i Empedoklo sam bio s Krete. Navedena se rečenica može pojednostaviti, a da ne izgubi smisao, rečenicom: "Ja lažem" ili još općenitijom: "Ova rečenica je lažna."

Ako ja govorim istinu kada kažem da ja lažem, znači da ja lažem, a to onda opet znači da govorim istinu, što opet znači da lažem... i tako unedogled. Koje je rješenje? Kako ustanoviti da li je ta rečenica istinita ili lažna? Postoji priča da je neki grčki filozof umro bezuspješno pokušavajući riješiti ovaj problem.

Drugi poznati primjer je Russelov paradoks. Russel je bio logičar i matematičar, Paradoks ide ovako: Sve skupove možemo podijeliti na one koji sadrže sebe kao element i one koji ne sadrže sebe kao element. Tako sad imamo dva skupa: skup svih skupova koji ne sadrže sebe kao element i skup svih skupova koji sadrže sebe kao element. U koji od ta dva skupa bismo onda stavili skup skupova koji ne sadrže sebe kao element? Ako ga stavimo u skup skupova koji ne sadrže sebe, onda će automatski sadržavati sebe!

Ovi problemi su dugo mučili logičare. Postojali su razni bezuspješni pokušaji da se taj problem riješi. U oba tipa problema imamo zajedničku osnovu:

Postoji struktura koja je samo-referentna, tj. rekurzivna, ali problem je u tome što ona unutar sebe same pokušava objasniti svoju negaciju. Dokaz o nerješivosti ovih problema ustvari je njihovo "rješenje".

Prvo je matematičar i logičar Kurt Gödel izveo 1931. svoje poznate teoreme necjelovitosti. On je dokazao da formalni aksiomatski sustavi nisu cjeloviti. U pokušaju da se stvori formalna teorija temeljne razine za prirodne brojeve koja će biti konzistentna i kompletna te na temelju koje će se tada moći izvesti cjelokupna matematika, Gödel je stvorio poseban način zapisivanja rečenica takve teorije – korištenjem brojeva samih. Taj se postupak naziva i Gödelov kôd, ili Gödelovo numeriranje i on omogućuje da se svaki izraz teorije može prikazati cijelim brojem na takav način da se ne mogu dobiti isti brojevi za različite izraze. Budući da je tragao za formalnim sustavom prirodnih brojeva kojeg je prikazivao prirodnim brojevima samim, dobio je situaciju da se sustav pokušava objasniti samim sobom. Tada je izveo dokaz o postojanju barem jedne tvrdnje čija se istinitost ili neistinitost ne mogu dokazati, odnosno formule koja tvrdi da tu istu formulu nije moguće dokazati jer za nju ne postoji Gödelov broj. U njegovu originalnom radu (Gödel, 1931, prema Hofstadter, 1980) piše da "za svaku -konzistentnu rekurzivnu klasu formula korespondira rekurzivna klasa oznaka r , takva da ni $G(v, r)$ niti $G(v, r)$ ne pripada u $Flg()$, (gdje je v slobodna varijabla od r)." Prevedeno na "normalniji" jezik, to znači da "sve konzistentne aksiomatske formulacije teorije brojeva uključuju neodlučive sudove". Gödel je dokazao da ovakav teorem vrijedi za sve formalne teorije koje se bave prirodnim brojevima. (Napomena: Za detaljni izvod teorema pogledati Ćirović, 1996 ili Podnieks, 1992). U početku se mislilo da je ovaj teorem ograničen na teorije ovog tipa zbog nekih njihovih specifičnosti.

No, kada se Turing-Churchovom tezom dokazalo da nema razlike između rekurzivnih funkcija i algoritama (a formalne teorije i nisu ništa drugo doli rekurzivno prebrojivi jezici) postalo je jasno da zbog "halting" problema teorem necjelovitosti vrijedi za bilo koju formalnu teoriju.

"Halting" problem doslovno znači "problem stajanja". Za Turingov stroj kažemo da prepoznaje jezik ako može izvoditi kompjutaciju nad njime. Ali to i nije dovoljan uvjet da će stroj izvesti kompjutaciju do kraja i stati u završnom stanju i time prihvatiti ili odbaciti ulazni string (te na taj način dokazati da je odgovarajući postupak rješiv, odnosno odlučiv). Zamislimo da imamo univerzalni Turingov stroj. Takav stroj može čitati bilo koji drugi Turingov stroj i njemu pripadajući ulaz, te će stati u prihvatnom stanju svaki put kada u taj Turingov stroj koji čita stane u prihvatnom stanju za određeni string. Takav stroj može dobiti i sebe kao ulaz, te izvršiti kompjutaciju. Isto tako može se konstruirati i drugi univerzalni Turingov stroj koji radi obrnuto: koji će stati u odbacujućem završnom stanju ako Turingov stroj koji on čita staje u prihvatnom stanju za njegov ulaz. Sada konačno dolazimo do problema. Što se događa kada takav stroj do-

bije samog sebe kao ulaz? Ako stane u prihvatnom stanju, tada mora stati u odbacujućem stanju, što znači da mora stati u prihvatnom... itd. Stroj ne može odlučiti u kojem stanju da stane. Doveo se do paradoksa i ovime je dokazano da postoji klasa nerješivih problema za bilo koji algoritam. Tek s ovim dokazom "riješeni" su paradoksi tipa Empedoklova i Russelova paradoksa, na taj način što se objasnila i dokazala njihova nerješivost. Osnovna karakteristika ovakvih paradoksa jest samo-referentnost (pozivanje na sebe) koja uključuje i negiranje sebe.

Ubrzo su i ovi dokazi našli svoju primjenu i u teoretiziranju o temeljnoj prirodi uma. No pokazalo se da tu treba biti vrlo oprezan da ne dođe do krivih tumačenja.

ARGUMENTI ZA I PROTIV KOMPJUTACIJSKE TEORIJE UMA

Jedan od prvih, a ujedno i najcitiranijih radova na tu temu je onaj Turingov (1950). U tom radu Turing je prvo dao operacionálnu definiciju mišljenja. Po toj definiciji za stroj možemo reći da misli ako uspješno simulira mišljenje, odnosno ako u komunikaciji sa strojem preko tipkovnice ne možemo razlikovati stroj od čovjeka. Ovaj kriterij je poslije nazvan "Turingov test". Turing je bio uvjeren u mogućnost konstrukcije takvog stroja.

U istom radu Turing navodi i argumente protiv mogućnosti da strojevi uspješno simuliraju mišljenje. U nastavku su svi navedeni zato što je većina njih i danas vrlo aktualna:

- 1) Teološki argument – Mišljenje je funkcija čovjekove besmrtné duše. Bog je dao besmrtnu dušu ljudima, ali ne životinjama ili strojevima. Otud strojevi ne mogu misliti.
- 2) Argument "glave u pijesku" – Posljedice strojnog mišljenja bile bi previše opasne. Zato se nadajmo i vjerujemo da strojevi ne mogu misliti.

Ova dva argumenta Turing ne smatra previše vrijednima da se o njima puno diskutira.

- 3) Matematički argument – Razni dokazi ukazali su na ograničenja snage diskretnih strojeva. To su rezultati Gödela, Churcha, Turinga i drugih. Nedvojbeno je, dakle, da postoje stvari koje strojevi ne mogu učiniti. Drugim riječima, za svaki stroj može se pronaći problem koji on ne može riješiti. Ali zato se može konstruirati drugi stroj koji taj problem može riješiti, ali ne može riješiti neki drugi itd. Osim toga, ovaj argument temelji se na nedokazanoj činjenici da ograničenja koja vrijede za strojeve ne vrijede za ljudski um. Je li to tako, još se ne može reći, ali i ljudi sami često griješe i daju pogrešne odgovore.

- 4) Argument svijesti – Po ovom argumentu strojevi ne mogu pisati sonete ili komponirati, jer nemaju emocije i nemaju znanje o tome da su nešto napisali. Ni jedan mehanizam ne može osjetiti radost uspjeha, ljutnju, tugu, itd.

Po Turingu, doslovno gledajući, solipsistički, ne možemo biti ni sigurni da opisane doživljaje može itko imati osim nas samih. O iskustvima tuđih doživljaja znamo samo ono što čujemo u komunikaciji. Nije nemoguće napraviti stroj koji će nas konzistentno izvještavati o "svojim doživljajima". Postoji podosta nejasnoća oko problema svijesti i taj problem se još istražuje. No, ne treba zato unaprijed tvrditi da je ona osnovni preduvjet za mišljenje.

- 5) Argument o različitim nemogućnostima – Ako se i napravi stroj koji radi sve spomenute stvari, on nikad neće moći raditi X. Pod X se uklapa čitav niz aktivnosti poput: ljubaznosti, prijateljstva, imati inicijativu, imati smisao za humor, činiti greške, zaljubiti se, uživati u jagodama s tučenim vrhnjem, itd.

Turing prvo obrazlaže da ljudi imaju predodžbe o strojevima sudeći prema onima s kojima su se susretali. Vjerojatno bi se mogli napraviti strojevi koji bi obavljali ili simulirali navedene aktivnosti. Ali, npr. što će nam stroj koji uživa u jagodama s tučenim vrhnjem. To što stroj ne može pokazati ili simulirati cjelokupnu raznovrsnost ljudskog ponašanja nije problem apsolutne nemogućnosti stroja, već, po Turingu, ograničenja kapaciteta stroja poput memorije i brzine, ali i nedostatka potrebe da se tako složen i dugotrajan posao uopće počne raditi.

- 6) Argument Lady Lovelace – Lady Lovelace bila je kći lorda Byrona i prvi programer namijenjen za Babbageov Analitički Stroj. Ona je rekla da takav stroj može izvesti sve što znamo odrediti kako da se izvede. Dakle, on ne može napraviti ništa novo, već samo ono što mu je određeno da radi.

Turing smatra da bi se ovaj problem mogao riješiti ako na pravilan način uspijemo programirati strojeve da uče. Osim toga, takvu vrstu programiranja Turing smatra izuzetno važnom u nastojanju da se naprave programi koji su kadri izvoditi inteligentne aktivnosti.

- 7) Argument o kontinuiranosti živčanog sustava – Istina je da stroj diskretnih stanja mora biti različit od kontinuiranog stroja i po tome je različit od ljudskog mozga. Ali, po Turingu, diskretni strojevi mogu simulirati odgovore, tj. davati vrlo približne odgovore onima koje bi davao diferencijalni analizator.
- 8) Argument o neformalnosti ponašanja – Ne postoje formalna pravila ljudskog ponašanja. Kada bi se ljudsko ponašanje moglo opisati pravilima, značilo bi da smo mi neka vrsta strojeva. Budući da to nije tako, mi smo

različiti od strojeva. Po Turingu, to što još nisu nađena pravila ponašanja ne znače da ona ne postoje. Možda su presložena, a možda se i ne istražuju na sve adekvatne načine.

- 9) Argument iz ekstrasenzorne percepcije (ako ona postoji) – Kako bi stroj simulirao aktivnost osobe obdarene telepatijom? Ovo je jedini argument na koji Turing ne daje zadovoljavajući odgovor.

Poznati primjer suprotnog stajališta od Turingova je onaj iznesen u Lucasovu (1961) radu i ustvari se odnosi na treći Turingov argument – matematički argument. On rezonira na sljedeći način: Gödelov teorem mora se primijeniti na strojeve zato što se odnosi na samu bit stroja koji je konkretna izvedba formalnog sustava (za koje teorem ustvari vrijedi). Slijedi da za bilo koji dani stroj postoji formula koja je nedokaziva u sustavu. Koliko god složeni stroj konstruiramo, davanjem novih pravila uvijek će postojati nešto što stroj neće moći napraviti, a ljudski um može. Otud Lucas zaključuje da ni jedan stroj ne može biti adekvatan model uma, i da su umovi po svojoj biti različiti od strojeva. Nadalje, izneseno je i da stroj ne može dokazati istinitost o vlastitoj konzistenciji, dok um koji nije stroj to može. Kao primjer za to Lucas navodi ljudsku samo-svjesnost kao dokaz o konzistentnosti ljudskog uma, te tvrdi da stroj nikad neće moći postići samo-svjesnost. Na kraju, navodi da je moguće konstruirati mehaničke modele uma, ali ni jedan neće biti kompletno adekvatan i dorečen.

Na ovom mjestu mogu se iznijeti i Hofstadterovi (1980) protuargumenti na ovu Lucasovu tezu:

- 1) Naime, ako tvrdimo da ne možemo napisati program koji će moći uvijek primijeniti operaciju "gödeliziranja" (tj. primijeniti Gödelovu metodu), znači da ni mi sami to ne možemo napraviti u svakom slučaju. Po Hofstadteru, Lucas čini pogrešku izjednačavajući apstraktnu mogućnost "gödeliziranja", i znanja kako da se to učini u svakom pojedinačnom slučaju. Ako ni mi ne možemo eksplicitno reći što je uključeno u primjeni Gödelove metode u svim slučajevima, tada će za svakog od nas vjerojatno doći do tako složenog slučaja da nećemo znati kako da primijenimo tu metodu.
- 2) Drugi, možda jasniji argument koji navodi Hofstadter jest da su ljudi podjednako podložni Gödelova teoremu kada se on prevede u riječi. Radi se o varijacijama Empedoklovog paradoksa. Imamo jedan lijepi primjer, a to je sljedeća rečenica:

"Lucas ne može konzistentno potvrditi OVU rečenicu."

Ako pažljivo promotrimo ovu rečenicu, možemo vidjeti da:

- a) ona je istinita, a ipak
- b) Lucas ju ne može konzistentno potvrditi.

Dakle, i Lucas i bilo koji drugi čovjek (jer se za svakog može naći odgovarajuća rečenica) nisu cjeloviti sustavi, jer postoji istinita rečenica čiju istinitost on ne može potvrditi.

- 3) Lucas također pretpostavlja da je stroj konkretna izvedba formalnog sustava. To doista i jest tako na razini hardvera. Međutim, on čini pogrešku smatrajući da je to jedina razina na kojoj se odvija manipulacija simbolima. No, to nije tako. Štoviše, istraživanja o umjetnoj inteligenciji razvila su brojne programe koji mogu manipulirati slikama, formulirati analogije, zaboravljati, miješati pojmove, itd. Ovakvi programi mogu biti nelogični i mogu lagati. Drugim riječima funkcioniranje viših razina takvih programa ne temelji se na logičkoj dedukciji, što znači da to nisu formalni aksiomatski sustavi, a opet nisu u kontradikciji s funkcioniranjem njima podložnog hardvera.

S Hofstadterovim protuargumentima ova rasprava nije završila. Ponovno ju je aktualizirao Roger Penrose (1991) knjigom *The Emperor's New Mind*, gdje, između ostalog, razmatra sposobnosti matematičara koje oni koriste u generiranju novih matematičkih sudova i njihovih dokaza. Po Penroseu, postupak po kojemu matematičar odlučuje o istinitosti novog suda (teorema, formule) ne može se temeljiti na algoritmu ili formalnom sustavu, jer tada bi se odmah mogla Gödelovom metodom konstruirati Gödelova rečenica i sasvim lako znati da je taj sud istinit i prije njegova izvođenja. Dakle, matematički uvid nije algoritamski. Također, po njemu ni gödelijanski uvid se ne može formalizirati.

Zanimljivi su i zaključci koje Penrose dalje izvlači iz navedene postavke: razmatranjem važnosti fizike za funkcioniranje mozga, on zaključuje da se po našem fizičkom razumijevanju mozgovnih operacija ne može utvrditi ne-algoritamski aspekt uma, pa je dakle, prema modernoj fizici, moguće simulirati ljudski um računalom. Otud Penrose zaključuje da moderna fizika mora biti u krivu, tj. da mozak koristi fizičke principe koji još nisu otkriveni, koji moraju biti ne-algoritamski po karakteru te da su vjerojatno povezani s interakcijom kvantne teorije i gravitacije. (Napomena: Penrose je po svojem obrazovanju fizičar koji je dao značajne rezultate u teoretskoj fizici – znatno je doprinio otkriću crnih rupa.)

Tu bi bilo zanimljivo spomenuti i jednu vrlo inventivnu ideju - da se u mozgu odvija kompjutacija i na kvantnoj razini. Ovakvi nalazi još su uvijek samo teoretski jer još ne postoji eksperimentalna metoda da se oni provjere. Jibu i Yasue (1997a; 1997b) dokazuju pomoću kvantnih jednadžbi da se u uvjetima određenih dijelova živčane stanice (naročito u mikrotubulima) moraju odvijati kvantni procesi koji su povezani sa spontanom emitiranjem fotona nalik onim kod lasera. Nadalje, oni sugeriraju da ako je to tako, tada već na razini jedne živčane stanice imamo uređaj velike kompjutacijske snage koji, među ostalim, može utjecati na

plastičnost staničnog prijenosa impulsa. Osim toga oni sugeriraju da su kvantne pojave odgovorne za prirodu svijesti.

Ipak, ako se zadržimo na matematičkom aspektu Penroseovih argumenata, mogu im se uputiti kritike: prvo, u matematici postoji duga povijest nekonzistentnosti, krivih teorija, što ukazuje da su umovi matematičara također podložni ograničenjima. Skoro svaka važna osoba u povijesti logike je barem jedanput objavila nekonzistentan skup aksioma (Russel i Norvig, 1995). Drugo, Penrose ne objašnjava na koji se način gödelijanski uvid ne može formalizirati: možda samim time što je čovjek uspio dokazati necjelovitost formalnih sustava on je superioran stroju koji to sigurno ne bi mogao izvesti. Ali, čini se da to nije tako i da je "gödelizacija" ustvari formalizirana. Štoviše, razvijeni su programi koji automatski dokazuju Gödelov teorem iz skupa temeljnih aksioma, na sličan način kao što je učinio i sâm Gödel. Takav je npr. program SHUNYATA (vidjeti Ammon, 1997).

ALTERNATIVNI PRISTUPI KOGNICIJI

Konekcionizam

Sredinom 80-ih došlo je do razvoja pristupa kogniciji koji predstavljaju alternativu klasičnim simboličkim pristupima. To su konekcionističke ili neuralne arhitekture koje pomiču naglasak s kompjutacijskih modela na kognitivnu neuroznanost. Takve arhitekture istražuju različite forme ovisnosti sustava o okolin-skom kontekstu, složene adaptivne sustave i modele umjetnog života (tzv. artificial life, Alife ili AL) koji su motivirani fenomenima slučajne mutacije i evolucijske selekcije, te sustave interaktivne robotike.

Temeljna strategija konekcionističkog pristupa je da se uzme takva temeljna procesirajuća jedinica koja je bliska apstraktnom neuronu (Rumelhart, 1996). Kompjutacija se tada izvodi paralelno, kroz jednostavne interakcije među takvim neuronima. Ideja je da takvi procesirajući elementi komuniciraju slanjem brojčanih vrijednosti duž linija koje povezuju te elemente. Jedna od osnovnih razlika konekcionističkih modela od onih simboličkih je u ograničenju da se cjelokupno znanje nalazi u vezama među jedinicama, tako da se vrste veza razlikuju od jednog modela do drugoga. Dakle, znanje je implicitno u strukturi modela koji ga koristi, za razliku od eksplicitnog znanja u simboličkim arhitekturama.

Korištenje ovakvih kompjutacijskih sustava u sustavima nalik onim mozgovnim ne nudi samo nadu da se opišu načini na koje mozak ustvari izvodi zadatke obrade informacija. Ovakav tip arhitektura nudi i jednostavnije solucije za određene klase problema koje se vrlo teško rješavaju na klasičan, serijski način. Takvi su na primjer problemi najboljeg slaganja ili optimizacije.

Postoji sedam glavnih komponenti svakog konekcionističkog sustava koji se zajedno nazivaju konekcionistički okvir. To su:

- 1) Skup procesirajućih jedinica.
- 2) Stanje aktivacije definirano nad procesirajućim jedinicama.
- 3) Izlazna funkcija za svaku jedinicu koja mapira svoje stanje aktivacije u izlaz.
- 4) Obrazac povezanosti među jedinicama.
- 5) Aktivacijsko pravilo za kombiniranje ulaza koji djeluju na jedinicu s njezinim trenutnim stanjem da proizvedu novu razinu aktivacije za tu jedinicu.
- 6) Pravilo učenja po kojemu se obrasci povezanosti mijenjaju s iskustvom.
- 7) Okolina unutar koje sustav mora operirati.

Konekcionistički modeli razlikuju se od klasičnih u brojnim dimenzijama (Smith, 1999):

- 1) Najčešće su paralelni, a ne serijski.
- 2) Njihovim sastavnim strukturama, naročito u stadiju dizajniranja, najčešće nije određen nikakav reprezentacijski ili semantički sadržaj.
- 3) Takav sadržaj ili interpretacija koja je na kraju određena obično se temelji na korištenju iskustva, prije nego na čistoj denotaciji ili opisu.
- 4) Ove arhitekture najčešće se koriste za modeliranje akcije, percepcije, navigacije, motoričke kontrole, pokreta tijela i drugih oblika vezivanja sa svijetom, dok su klasične arhitekture povezane s odvojenim, često deduktivnim zadacima koji se tiču raznih oblika rješavanja problema i baratanja znanjem.
- 5) Konekcionističke arhitekture nastoje da budu "u realnom vremenu" u smislu zahtjeva za bliskim povezivanjem trajanja kompjutacijskih procesa i trajanja subjektivne domene.

Visoki prioritet je obično više posvećen i fleksibilnosti i mogućnosti konekcionističke arhitekture da se snalazi s neočekivanim varijacijama u okolini nego dubokom rezoniranju i čistim deduktivnim vještinama.

Dakle, ove promjene označavaju i promjene u teoretskoj orijentaciji unutar kognitivne znanosti - koje su često opisane u kontrastu s tradicionalnim "kompjutacijskim" pristupima. Ipak, činjenica je da danas ne postoji dovoljno bogata i šire prihvaćena teorija kompjudacije koja može objasniti svu raznolikost u kompjudaciji "stvarnoga svijeta". Prema tome, nije ni jasno mogu li se doista novopredloženi sustavi odrediti kao ne-kompjudacijski. Npr. nitko ne bi mogao reći da ti novi prijedlozi izbjegavaju granice kompjudacijske kompleksnosti koje su postavljene matematičkom teorijom računalne znanosti. Najvjerojatnije se može

očekivati da će se pojam "kompjutacije" tako adaptirati da u svoj doseg uvrsti i te novije pristupe (Smith, 1999).

Dinamička hipoteza u kognitivnoj znanosti

Kognitivizam koji na kogniciju gleda kao na kompjutaciju u smislu manipulacije diskretnim simbolima potakao je nastanak reakcija na takav pristup koji sebe nazivaju ne- ili čak anti-kompjutacijskim. Takva je dinamička hipoteza o prirodi uma.

Središnja ideja kompjutacijske hipoteze jest da su kognitivni agenti u svojoj osnovi diskretna računala. Prema dinamičkoj hipotezi, kognitivni agenti su dinamički sustavi. Izvjestan utjecaj na ovu hipotezu izvršio je napredak u teoriji dinamičkih sustava, naročito u teoriji nelinearnih sustava i teoriji kaosa. Prema van Gelderovu radu (1998), u kojemu objašnjava i argumentira za dinamičku hipotezu, to znači da kognicija pokazuje određena svojstva dinamičkih sustava. To znači da koriste numeričke varijable, koja imaju svojstva vektora u sustavu.

Dinamička hipoteza uključuje dvije glavne komponente: prirodnu hipotezu i hipotezu znanja.

Prirodna hipoteza je zahtjev o prirodi kognitivnih agenata. Kognitivni su agenti doslovno identični s dinamičkim sustavima. Hipoteza znanja odnosi se na teoriju kognitivne znanosti. Po njoj bi trebalo cijelu kogniciju shvaćati dinamički. Prema van Gelderu, dinamička hipoteza je empirijska hipoteza i kao takva stoji kao čvrsta alternativa kompjutacijskoj hipotezi.

Pokazalo se za rekurentne (povratne) neuralne mreže koje imaju kontinuirani ulaz, dakle realne, a ne cijele ili racionalne brojeve, da mogu implementirati dinamičke sustave. One se u nekim radovima nazivaju i super-Turingovi automati (npr. Kilian i Siegelmann, 1996; Siegelmann, 1995). Radi se o formalnim strojevima s kompjutacijskom snagom većom od snage Turingovih strojeva, u tom smislu što prepoznaju širu klasu jezika i mogu izračunati bilo koju rekurzivnu (Turingovu) funkciju. Dakle, ti su strojevi u svojoj osnovi analogni, a ne diskretni.

Je li pristup koji shvaća kogniciju i adaptivno ponašanje kao dinamički sustav potpuno u suprotnosti s reprezentacijsko-kompjutacijskim shvaćanjima? Van Gelder (1998) smatra da je dinamizam jedini pravi način proučavanja kognicije i odbacuje kompjutacijski pristup. Clarck (1997) naziva takvu, suparničku tezu tezom radikalnog utjelovljenja kognicije, te smatra da se dinamizam lako može pogrešno tumačiti. Nadalje, u istom radu, on obrazlaže da nije zastupnik takve teze, te pokušava pronaći središnju opciju u kojoj se dinamička i kompjutacijska shvaćanja mogu postaviti komplementarno, radije nego suparnički. Konačno, daje zanimljiv prijedlog. Postavlja razliku između tzv. on-line i off-line kognicije i rezoniranja. On-line kognicija se odnosi na akcijski orijentirane karakteristike, u realnom vremenu, u interakciji s okolinom. U to spadaju operacije poput vidne

percepcije, motorne kontrole i one su pogodne za opisivanje dinamičkim, dakle i konekcionističkim modelima. S druge strane, off-line kognicija nije u realnom vremenu i nije u nužnoj interakciji s okolinom. Ona je reprezentacijska u smislu baratanja znanjem. To su procesi poput dedukcije i oni su pogodni za opisivanje simboličkim, dakle klasičnim kompjutacijskim modelima.

O sličnim idejama o mogućnosti kompatibilnosti dinamičkih pristupa i drugih kognitivnih pristupa raspravljaju i Bechtel (1998), te Wells (1998). Buduća istraživanja trebala bi koristiti i kontinuirane (dinamičke) i diskretne (simboličke) modele da se razumije kognicija, jer nekim kognitivnim kapacitetima bolje odgovaraju jedni, a drugima drugi.

Metakognicija i doživljavanje beskonačnosti

Bez obzira na prijašnju digresiju o alternativnim pogledima na prirodu kognicije i kompjutacije koja nam je poslužila da cjelovitije sagledamo probleme o kojima danas raspravljaju teoretičari kognitivne znanosti, središnje ideje koje želimo istaknuti u ovom tekstu jesu o onoj fascinantnoj složenosti ljudskog uma koji mu omogućuje da se bavi onim problemima koji zadaju toliko briga istraživačima i potiču toliko rasprava. Riječ je o problemima koji u svom temelju imaju postavke halting problema i Gödelova teorema. Dakle, riječ je o ljudskoj intuiciji za beskonačnost. Kada se govori o beskonačnosti, valja napomenuti da imamo dvije vrste: *prebrojivu* i *neprebrojivu*. Do ovog otkrića došao je, krajem 19. st., njemački matematičar Georg Cantor, koji se bavio teorijom skupova. Do tada se matematika nevoljko bavila beskonačnim skupovima. On je dokazao da je skup prirodnih brojeva beskonačan, ali prebrojiv skup: do svakog se elementa može doći operacijom brojenja, odnosno funkcijom sljedbenika. Svaki skup koji se može preslikati na ovaj skup je također beskonačan *prebrojiv* skup. Tako možemo dokazati da skup svih parnih brojeva ima jednak broj elemenata kao i skup svih prirodnih brojeva (a ne upola manje). Isto vrijedi za skup svih racionalnih brojeva. Ali ne vrijedi za skup realnih brojeva. Cantorovom *metodom dijagonalizacije* dokazano je da je skup realnih brojeva *neprebrojiv*. To je druga vrsta beskonačnosti. Ovaj se skup može preslikati na kontinuum. I bilo koji djelić tog kontinuum (npr. interval između 0 i 1) također sadrži neprebrojivo beskonačno točaka, tj. elemenata. S beskonačnošću je uvijek bilo teško baratati u raznim područjima znanosti i razmatranja ovog pojma često su dovodila do paradoksa.

Kakva je veza između beskonačnosti i navedenih matematičkih formulacija? Takva što bilo koja struktura koja se poziva na samu sebe (koja je rekurzivna) ima mogućnost da se "zavrti" u beskonačnoj petlji. Stvar je još složenija kada se ta struktura negira pozivanjem na sebe. Tada, kako je to već opisano, imamo paradoks i nikakvo rješenje se ne može dati. No, tu opet imamo beskonačnu petlju: struktura se negira, onda ta negirana struktura opet se negira čime se afirmira, pa se ta afirmirana i opet negira...itd. Ljudi imaju intuiciju o

beskonačnosti u smislu da je brzo prepoznaju. Nitko se razmišljajući o rekurziji neće naći u beskonačnoj petlji ili mu se neće "srušiti sustav" pri pokušaju rješavanja nekog paradoksa. Razlog tome je što je ljudski kognitivni stroj sposoban ne samo izvoditi kompjutaciju, već i kompjutaciju o toj kompjutaciji, i kompjutaciju o kompjutaciji o kompjutaciji, itd. Taj se fenomen u kognitivnoj psihologiji naziva metakognicija.

Metakognicija se (Kellog, 1996) odnosi na kogniciju o kogniciji, ili mišljenje o mišljenju. Ovaj široki pojam uključuje bilo kakvo znanje ili kognitivni proces koji se odnosi, prati ili kontrolira bilo koji aspekt kognicije (Moses i Baird, 1999). Metakognicija sudjeluje u mnogim aspektima kognicije kao što su pažnja, pamćenje, komunikacija, rješavanje problema, te ima važnu primjenu u područjima poput obrazovanja, razvoja i starenja, neuropsihologije te svjedočenja svjedoka. Tijekom kognitivnog razvoja ljudi razvijaju svoje sposobnosti da prate svoje procese mišljenja i da biraju strategije koje su efikasne za zadatak kojim se bave. Istraživanja metakognicije pri rješavanju problema proučavala su razlike među odraslim ekspertima i početnicima i pokazalo se da se značajno razlikuju u metakognitivnim sposobnostima (Kellog, 1996).

Povezani pojam s metakognicijom i koji je na ovom mjestu interesantniji jest metarezoniranje. Metarezoniranje je u najširem smislu "rezoniranje o rezoniranju" (Russell, 1999). To je bilo koji kompjutacijski proces koji se tiče operacija nekog drugog kompjutacijskog procesa unutar istog entiteta. Istraživanja metarezoniranja bila su nerijetko usmjerena na dizajniranje kognitivnih arhitektura koje mogu podržavati introspekciju i refleksiju. Najpoznatija takva arhitektura je SOAR, koja rješava probleme koristeći se i metarazinama (Laird i sur., 1987, prema Russell i Norvig, 1995).

Ipak, nismo naišli na istraživanja i teorije koje se detaljnije bave ljudskim rezoniranjem s beskonačnošću i strategijama koje se koriste pri rješavanju nerješivih problema. Smatramo da bi detaljnije spoznaje o ovoj vrsti ljudskog rezoniranja mogle ponuditi važne smjernice za shvaćanja o prirodi kompjutacije koju izvodi ljudski kognitivni aparat. Konkretno, radilo bi se o ispitivanju ljudske metakognicije i strategija pri suočavanju s problemima koji se ne mogu riješiti, dakle paradoksa. Pojam metakognicije je važan, jer ljudi su sposobni uvidjeti da odgovarajući problemi nisu rješivi, pa i dokazati zašto nisu rješivi.

ZAKLJUČAK I BUDUĆE SMJERNICE

Konačno, možemo se vratiti na pitanje iz naslova: Kakvi su ljudi kompjutori? Na ovo pitanje još se ne može dati jednostavan odgovor. Kako smo vidjeli, postoje ograničenja kompjutacije i još traju rasprave jesu li ljudski umovi podjednako podložni tim ograničenjima. Ograničenje na koje se usredotočio ovaj rad jest matematički argument o ograničenju kompjutacije. Treba napomenuti da to nije

jedino pitanje. Postoje tzv. velika pitanja vezana uz kogniciju i nagađa se je li uopće moguće dati odgovor na njih. To su problem odnosa uma i tijela, problem svijesti, intencionalnosti, qualie itd. (Harman, 1993). Ovi problemi spadaju u filozofski aspekt kognitivne znanosti.

Ovim radom htjelo se kao prvo ukazati na osnovne postavke i karakteristike jedne velike ideje i cjeline - teorije kompjutacije, na čijim se postavkama može temeljiti opća teorija kognicije. Pri tome smo se trudili objasniti teorijske specifičnosti koje ove postavke čine problematičnim kada se pokušaju primijeniti na ljudsku kogniciju. Pokazali smo da su te specifičnosti povezane s klasom nerješivih problema - tj. problema za koje se ne može stvoriti algoritamski opis. Nadalje, objasnili smo i neke alternativne pristupe u izučavanju kognicije.

Ideja je da se provede istraživanje iz područja metakognicije, ili preciznije metarezoniranja, u kojem bi se u seriji eksperimenata s različitim nerješivim problemima (i sličnim rješivim zbog kontrolne situacije) pokušale pronaći općenitosti u strategijama koje ljudi koriste. Pri tome bi se pokušale utvrditi zakonitosti na temelju kojih ljudi prepoznaju beskonačnost i nerješiv problem. Bile bi korištene uobičajene opisane metode za ispitivanje rješavanja problema. Na ovaj bi se način mogao prikupiti važan materijal koji bi koristio ne samo u ovom specifičnom području ljudske izvedbe suočene s nerješivim problemima i beskonačnosti, već i u znatno širem razumijevanju ljudske kognicije. Dobiveni rezultati mogli bi poslužiti za konstruiranje algoritma sposobnog da se suočava s nerješivim problemima.

LITERATURA

- Ammon, K. (1997). An automatic proof of Gödel's incompleteness theorem. *Artificial Intelligence* 95, 203-207.
- Bechtel, W. (1998). Representations and cognitive explanations: assessing the dynamicist's challenge in cognitive science. *Cognitive Science*, 22, 295-318.
- Clarck, A. (1997). The dynamical challenge. *Cognitive Science*, 21, 461-481.
- Čirović, S. (1996). *Uvod u matematičku logiku i teoriju rekurzivnih funkcija*, Zagreb: Filozofski niz.
- Dietrich, E. (1999). Algorithm. U R. A. Wilson i F. C. Keil (ur.), *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge: MIT Press.
- Harman, G. (1993). Some philosophical issues in cognitive science: qualia, intentionality, and the mind-body problem. U M. I. Posner (ur.), *Foundations of Cognitive Science*. Cambridge: MIT Press.
- Hofstadter, D. R. (1980). Gödel, Echer, Bach: *An Eternal Golden Braid*, London: Penguin Books.
- Hopcroft, J. E., i Ullman, J. D. (1979). *Introduction to Automata Theory, Languages and Computation*. Reading: Addison - Wesley.
- Jibu, M., i Yasue, K. (1997a). What is mind? - Quantum field theory of evanescent photons in brain as quantum theory of consciousness. *Informatica*, 21, 471-490.

- Jibu, M., i Yasue, K. (1997b). Magic without magic: Meaning of quantum brain dynamics. *The Journal of Mind and Behavior*, 18(2,3): 205[103] -228[126].
- Kilian, J., i Siegelmann, H. T. (1996). The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128, 48-56.
- Lucas, J. R. (1961). Minds, Machines and Gödel. *Philosophy*, 36, 112-127.
- Matuszek, D. (1996). Introduction to the Theory of Computation. Villanova University.
- Moses, L. J., i Baird, J. (1999). Metacognition. U R. A. Wilson i F. C. Keil (ur.), *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge: MIT Press.
- Penrose, R. (1991). The Emperor's New Mind: Concerning Computers, *Minds and Laws of Physics*. New York: Penguin Books.
- Podnieks, K. (1992). *Around Goedel's Theorem*. University of Latvia.
- Posner, M. I. (1996). *Foundations of Cognitive Science*. Cambridge: MIT Press.
- Pylyshyn, Z. W. (1996). Computing in cognitive science. U M. I. Posner (ur.), *Foundations of Cognitive Science*. Cambridge: MIT Press.
- Rumelhart, D. E. (1996). The architecture of mind: a connectionist approach. U M. I. Posner (ur.), *Foundations of Cognitive Science*. Cambridge, MIT Press.
- Russell, S. (1999). Metareasoning. U R. A. Wilson i F. C. Keil (ur.), *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge: MIT Press.
- Russell, S., i Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, Prentice Hall.
- Siegelmann, H. T. (1995). Computation beyond Turing limit. *Science*, 268, 545-548.
- Simon, H. A. (1991). Primjena kognitivne znanosti u rješavanju ljudskih problema. U N. N. Šoljan i M. Kovačević (ur.), *Kognitivna znanost*. Zagreb: Školske novine.
- Sipser, M. (1997). *Introduction to the Theory of Computation*. Boston: PWS Publishing Company.
- Smith, B. C. (1999). Computation. U R. A. Wilson i F. C. Keil (ur.), *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge: MIT Press.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, 230-265.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 49, 433-460.
- Van Gelder, T. (1998). Dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21, 615.
- Wells, A. J. (1998). Turing's analysis of computation and theories of cognitive architecture. *Cognitive Science*, 22, 269-294.
- Wilson, R. A. i Keil, F. C. (ur.), *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge: MIT Press.

HOW ARE PEOPLE LIKE COMPUTERS: SOME OPEN-ENDED QUESTIONS ON COMPUTATIONAL ATTITUDE TO MODERN COGNITIVE SCIENCE

Summary

The aim of this paper is to explain the computational idea about the mind. Mathematical basics of the theory of computation are given briefly. Furthermore, arguments are described for and against the thesis that the human mind can accept computational, i.e. algorithmic description. In addition to contemporary speculations and disagreements when trying to base cognition theory, this paper gives mathematical argument against computational hypothesis of the mind that is based on Gödel's incompleteness theorem and the halting problem. The relationship between those mathematical formulations with unsolvable problems and paradoxes is showed. Alternative approaches to the study of cognition: connectionism and dynamism are also described. Finally, there are some suggestions for future investigations about such aspects of human cognition that are concerned with solving such critical tasks in respect to described arguments.

Key words: cognition; computation; Turing machine; algorithm; Gödel's theorem; halting problem; paradoxes; metacognition

Primljeno: 16. 12. 1999.