

LLM chatbot kao pomoćnik za dokazivanje matematičkih teorema

Zlatko Sirotić, univ. spec. inf.

ISTRA TECH d.o.o., Pula

Elen Sirotić, univ. bacc. ing. aedif.

CASE konferencija 2026

09.03. - 10.03.2026., Rijeka

CASE 2023, 2024, 2025

□ 2023.

Prolog i "klasična" umjetna inteligencija

Funkcijsko programiranje, imutabilne kolekcije
i baze podataka

□ 2024.

My name is Programmer, Chat Programmer

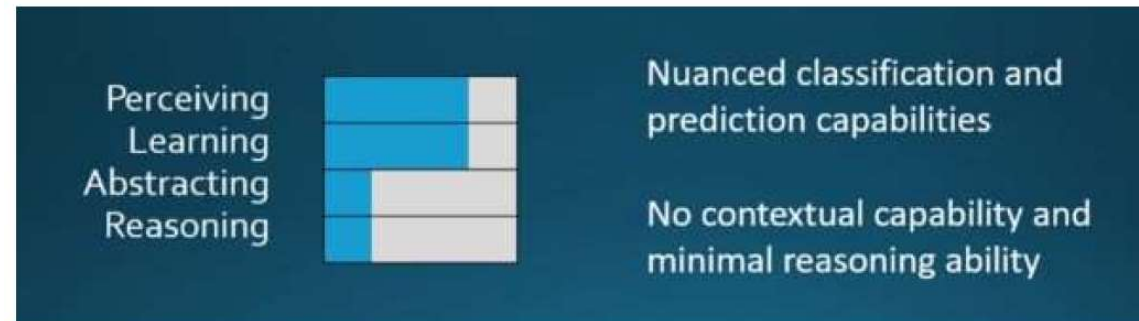
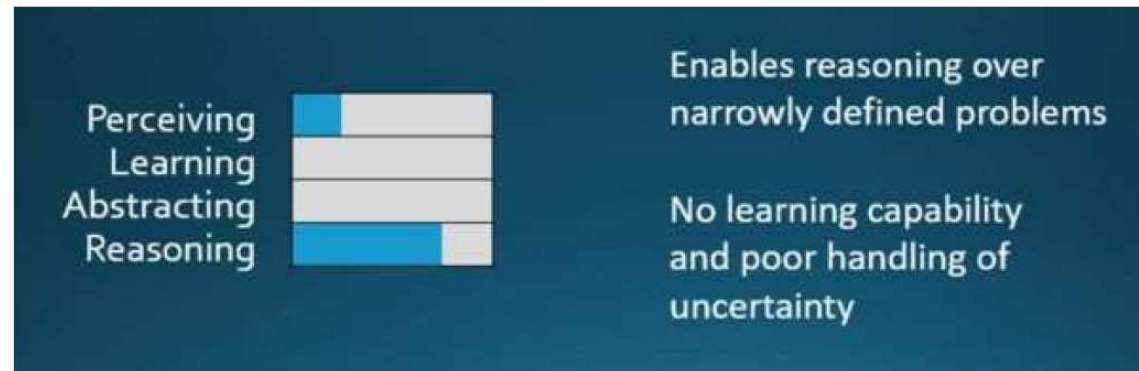
□ 2025.

Kauzalno zaključivanje –
nadogradnja statističkog zaključivanja

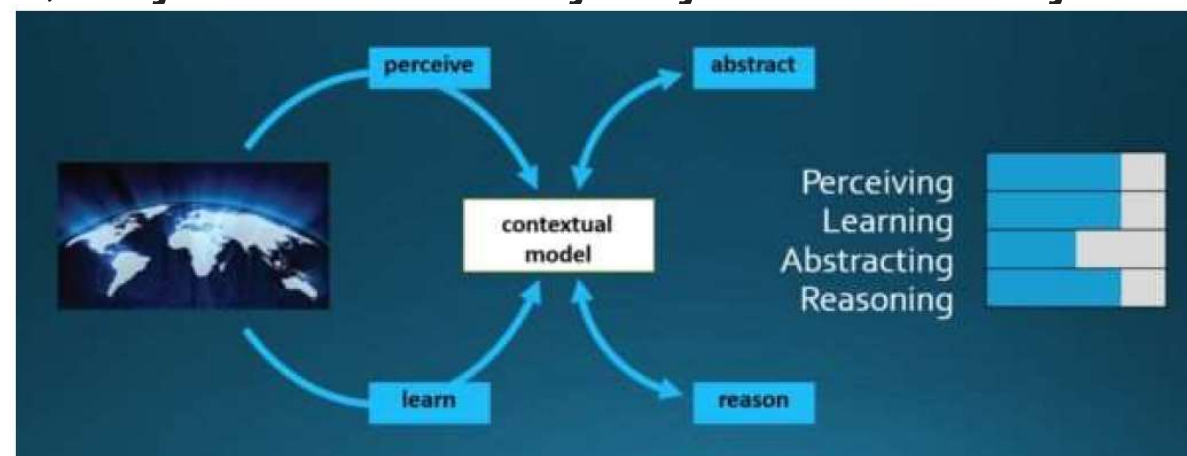
Umjetna inteligencija – kratka povijest

- Pojam 'umjetna inteligencija' prvi se put koristio 1956. na konferenciji u Dartmouthu, ali prvi rad na području UI su još 1943. napisali Walter Pitts i Warren McCulloch "A Logical Calculus of the ideas Imminent in Nervous Activity", prvi matematički model neuronske mreže.
- **Dakle, neuronske mreže su prethodile simboličkoj UI.** Međutim, neuronske mreže, kao i šira subsimbolička (statistička) UI, bile su do prije desetak godina inferiornije.
- Može se reći da postoje tri vala (engl. wave) umjetne inteligencije. **U prvom valu je prevladavala simbolička UI,** koja je vrlo dobra u rasuđivanju (Reasoning), ali ima dosta slabe sposobnosti opažanja (Perceiving), a vrlo je loša kod učenja (Learning) i stvaranja apstrakcija (Abstracting).
- **Drugi val, korištenje subsimboličke UI** (posebno neuronskih mreža), doveo je do velikih poboljšanja kod percepcije i učenja, ali je jako pala sposobnost rasuđivanja.

Umjetna inteligencija



- Radi se na trećem valu, spoju simboličke i subsimboličke UI u **hibridnu UI**, koja bi imala najbolje strane obje.



Uvod

□ Prikazujemo kako su nam LLM chatbotovi (ChatGPT, Gemini, Claude, DeepSeek) pomagali u dokazivanju jedne matematičke tvrdnje vezane za najkraći put između dvije točke položene na suprotnim bazama pravilnog n -terokuta.

□ Ta tvrdnja je ova
(napomena: ovdje još ne dajemo preciznu definiciju što znači p -neovisan put i gdje se točke A i B nalaze):

Svaka neparna pravilna prizma ima barem jedan p -neovisan kraći put, tj. kraći put čija duljina ne ovisi o udaljenosti točaka A i B ("mrava" i "meda") od ruba, dok ga parne prizme nemaju. Duljina najkraćeg puta je:

$$d = b \sqrt{k^2 + 2 \cos(\pi/n) k + 1}$$

Uvod

- Početnu verziju tog matematičkog problema (i računalne igre vezane za to) prikazali smo na CASE 2020, pod nazivom "Mrav i med na prizmi (i valjku) - matematička podloga i koncept igre".
- Sada imamo već 4. verziju te nekadašnje prezentacije, ali ćemo u sadašnjoj prezentaciji i radu staviti naglasak na to koliko nam kod dokazivanja matematičkih teorema može koristiti "čisti" LLM chatbot, bez specijaliziranih dokazivača teorema (theorem prover).

Prikaz prezentacija Mrav i med:

- R1 sa CASE 2020**
- najnovija R4 (Prilog A)**

- Slijedi brzi prikaz dvije prezentacije

Prikaz Python programa za kotrljanje baze po plaštu: - pravih prizmi - valjka

- Slijedi brzi prikaz dva programa

Dokazivanje matematičkih teorema uz pomoć AI

□ Neki načini dokazivanja matematičkih teorema pomoću umjetne inteligencije (u širem smislu) :

1. jednostavno dokazivanje pomoću LLM-a

2. dokazivanje pomoću LLM-a, tako da se koristi LLM proof generator i LLM proof verifier; primjer:

<https://arxiv.org/abs/2510.12829>

3. dokazivanje pomoću dokazivača teorema / asistenta:

- npr. Vampire je Automated theorem prover
- npr. Lean je Proof assistant (Interactive theorem prover)

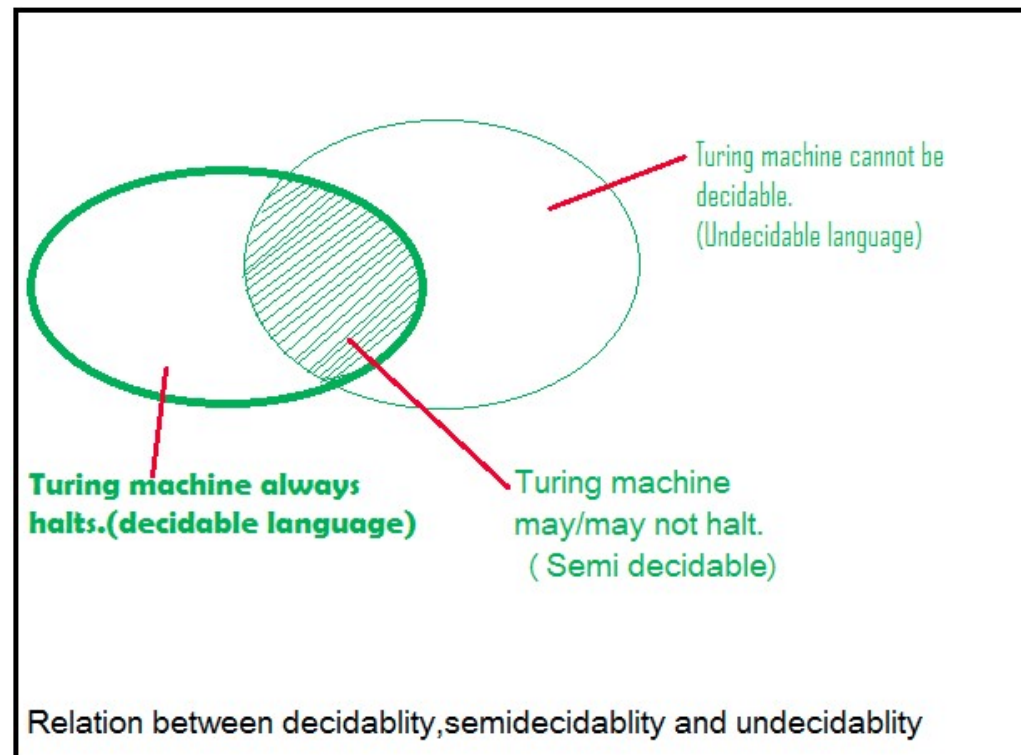
4. dokazivanje pomoću LLM-a i theorem prover-a, tako da LLM prevodi prirodni jezik u npr. Lean i obrnuto

IMO 2025 i zlatna medalja

- ❑ <https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/>
- ❑ <https://garymarcus.substack.com/p/deepmind-and-openai-achieve-imo-gold>

Digresija – odlučivi, poluodlučivi i neodlučivi jezici

- Postoje jezici / problemi koji su **neodlučivi**, tj. koji nemaju gramatiku, i za koje ne postoji automat koji ih prihvaća. Takvih jezika / problema ima čak neprebrojivo beskonačno mnogo (dok su **odlučivi** "samo" prebrojivo beskonačni).
- No ne znači da neki pojedinačni problem iz tog skupa neodlučivih problema nije moguće nikako riješiti.
- Postoje i **poluodlučivi** jezici / problemi (isto su prebrojivo beskonačni).



Vremenska i memorijska složenost algoritama / problema / prihvatanja jezika

- Rekurzivni jezici su uvijek **odlučivi**.
- No to ne znači da je svaki rekurzivni jezik / problem jednostavan za rješavanje. Što ako nam za njegovo rješavanje treba milijardu godina na (danas) najbržem računalu?
- To je tema o (vremenskoj i memorijskoj) **složenosti prihvatanja jezika** (naziva se i **teorija kompleksnosti**).
- U nastavku će samo ukratko biti prikazani neki pojmovi iz teorije kompleksnosti:
P, NP, NP-teški i NP-potpuni problemi.

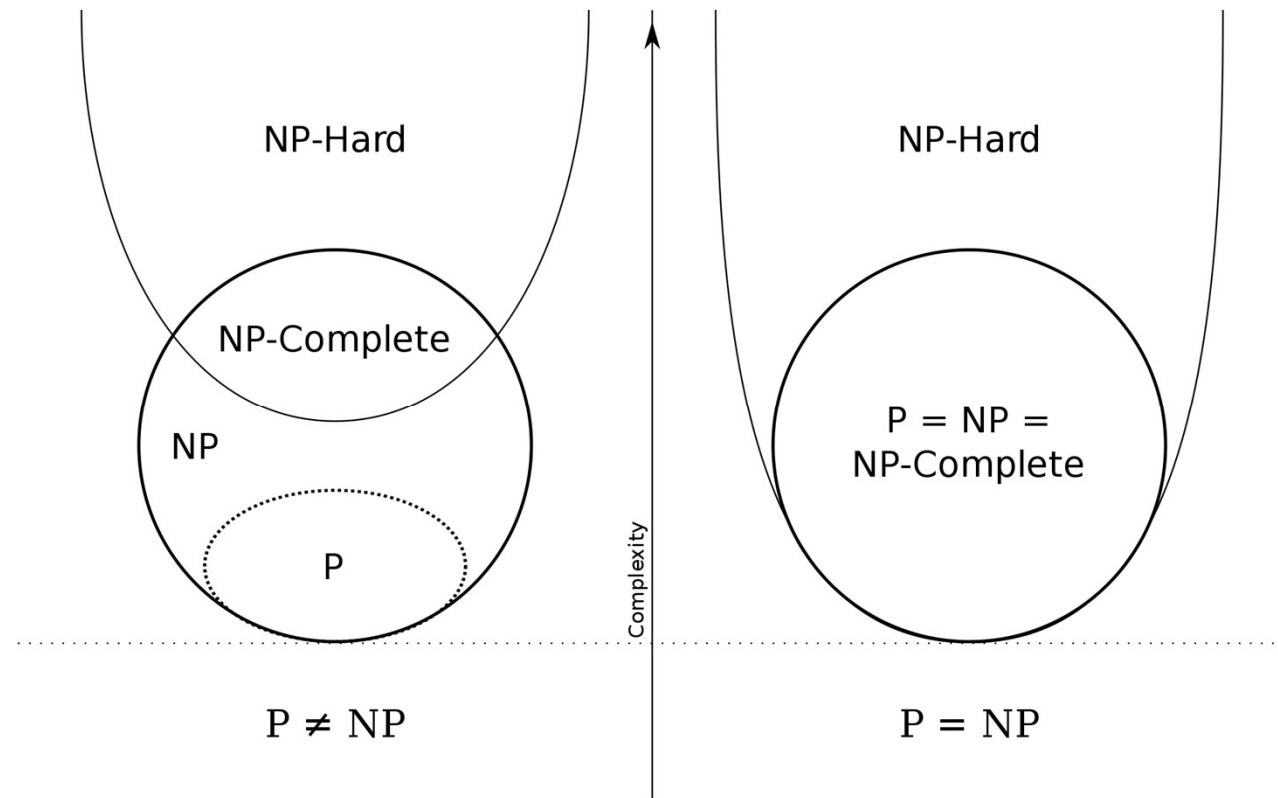
P i NP klase

problema / algoritama / jezika

- Često se smatra da je klasa problema / algoritama / jezika koja ima polinomnu vremensku složenost (relativno) prihvatljiva za praksu (barem u radu s malim potencijama).
- Problemi koji se mogu rješavati u polinomnom vremenu spadaju u **P klasu problema**.
- Postoje problemi koji (vjerojatno) nisu u klasi P, ali koji imaju osobinu da je neko **konkretno potencijalno rješenje** tog problema provjerivo u polinomnom vremenu.
- Klasa koja sadrži te probleme (plus one iz P), naziva se **NP klasa problema** (NP ne znači Ne-Polinomni, nego **Nedeterministički Polinomni**).
- Postoje i tzv. **NP-teški (NP-hard) problemi**, kod kojih niti provjera rješenja ne mora biti moguća u polinomnom vremenu.

P =? NP - pitanje za milijun dolara

- Nitko ne zna da li je klasa P pravi podskup klase NP, ili su možda te dvije klase jednake.
- To je jedan od sedam tzv. **Millennium Prize Problems** u matematici, koje je zadao Clay Mathematics Institute 24.05.2000. Onaj tko korektno riješi neki od tih problema, dobiva milijun dolara od instituta.
- Vjeruje se da je odgovor kao na slici lijevo, a ne desno:



NP-potpuni (NP-complete) problemi

- **NP-potpuni su oni problemi za koje vrijedi da se svaki drugi NP problem može svesti na njih u polinomnom vremenu** (danas ih je poznato više od 100).
- To znači – ako bi se pokazalo da je bilo koji NP-potpuni problem rješiv u polinomnom vremenu, onda bi se dobio odgovor i vrijedilo bi da je $P = NP$.
- To bi puno značilo za rješavanje nekih današnjih problema, za koje još ne poznajemo polinomno rješenje, jer bi se isplatilo tražiti takvo (polinomno) rješenje.
- Međutim, kada bi vrijedilo da je $P = NP$, to bi imalo i negativnih reperkusija, jer bi većina današnjih kriptografskih algoritama postala nesigurna.
- Npr. mnogi današnji kriptografski algoritmi koji koriste proste brojeve, polaze od pretpostavke da je rastav velikog broja na proste faktore nerješiv u polinomnom vremenu.

Primjer NP-potpunog problema

- Boolean satisfiability problem, skraćeno SAT(isfiability)

- **SAT problem** je problem određivanja da li postoji (barem jedan) skup vrijednosti varijabli u (bilo kojoj) Booleovoj formuli (formuli logike sudova), za koji (skup vrijednosti varijabli) formula poprima vrijednost TRUE.
- **SAT je prvi problem za kojeg je dokazano da je NP-potpuni** (Cook–Levin teorem).
- To znači – ako bi netko dokazao da postoji polinomno rješenje SAT problema, riješio bi pitanje $P = ? NP$ (odgovor bi bio potvrđan).
- Ili, ako bi dokazao da ne postoji polinomno rješenje SAT problema, isto bi riješio pitanje $P = ? NP$ (odgovor bi bio negativan).

Analogija s P i NP problemima

- Možemo li reći: "Korisno je koristiti LLM za pomoć u dokazivanju matematičkih teorema, iako je LLM statistička AI, a ne simbolička AI. Naime, općenito je lakše provjeriti matematički dokaz, nego doći do njega. **LLM možemo koristiti za traženje dokaza, a onda čovjek (sa ili bez simboličkih alata) provjerava taj dokaz.**"
- **U NP kontekstu vrijedi:**
 - generiranje rješenja može biti teško,
 - verifikacija danog certifikata može biti znatno lakša.
- **U matematici je često slično:**
 - kreativni čin konstrukcije dokaza je heuristički, nelinearan, često zahtijeva uvid
 - verifikacija dokaza je lokalna, deterministička i mehanička.
- Zato postoji realna metodološka analogija.

Analogija s P i NP problemima

□ Gdje analogija prestaje biti precizna - tri važne razlike:

1. Matematička provjera nije uvijek "polinomna"

U matematici "verifikacija je lakša" je često točno, ali nije teorijski garantirano kao u definiciji NP.

2. LLM ne generira "certifikate"

Izlaz LLM-a nije inherentno formalni dokaz.

3. Statistička vs. simbolička AI

LLM je distribucijski model jezika, optimiziran za vjerojatnost nastavka niza. On nema internu garanciju deduktivne korektnosti.

Teorem o p-neovisnim putevima na neparnim pravilnim prizmama

- Slijedi brzi prikaz par verzija teorema
- Pa prikaz zadnje verzije (Prilog B)

Literatura (dio)

- Ben-Ari M. (2012): Mathematical Logic for Computer Science (3. izdanje), Springer
- Ertel W. (2017): Introduction to Artificial Intelligence, Springer
- Martin, J.C. (2010): Introduction to Languages and The Theory of Computation (4.izdanje), McGraw-Hill
- Skansi, S. (2018): Introduction to Deep Learning, Springer

- Lenat D., Marcus G. (2023): Getting from Generative AI to Trustworthy AI: What LLMs might learn from Cyc, <https://arxiv.org/abs/2308.04445>
- Le Duc H., Liberti L. (2025): Mathematics with large language models as provers and verifiers, <https://arxiv.org/abs/2510.12829>
- Zhihong Shao ... (2025): DeepSeekMath-V2: Towards Self-Verifiable Mathematical Reasoning, <https://arxiv.org/abs/2511.22570>