

## Testul Turing

### Cine a fost Alan Turing?

---

Alan Turing (1912–1954) a fost un matematician și criptograf britanic care a descifrat codul Enigma utilizat de naziști în timpul celui de-al Doilea Război Mondial, contribuind decisiv la scurtarea războiului.

Deși a trăit doar 41 de ani, contribuțiile sale au avut un impact uriaș asupra informaticii moderne. Alături de John von Neumann, care a pus bazele mașinii cu registre, Alan Turing este considerat unul dintre părinții calculatoarelor moderne. Contribuțiile sale sunt studiate în programele universitare de informatică: a creat modelul cunoscut sub numele de mașina Turing și a propus testul Turing în domeniul inteligenței artificiale.

O mașină Turing este un model teoretic de calcul, propus de Alan Turing în 1936, care descrie cum un computer poate procesa informații pas cu pas, prin reguli simple. Este considerată fundamentul teoriei computationale. Separat de acest model, Turing a proiectat mașini electromecanice (Bombe) care au contribuit la spargerea codului Enigma.

## Simulator Mașină Turing

Scop: înlocuiește toate **1** cu **0**, apoi se oprește.

0	0	0	1	0	1	—	—	—
---	---	---	---	---	---	---	---	---

**Stare curentă: start**

Pas următor

### Istoric tranziții:

Stare: start, Poziție: 1, Simbol: 0, Acțiune: scrie 0, deplasează capul dreapta  
Stare: start, Poziție: 2, Simbol: 1, Acțiune: deplasează capul dreapta  
Stare: start, Poziție: 3, Simbol: 1, Acțiune: scrie 0, deplasează capul dreapta

Povestea vieții sale este prezentată în filmul biografic Jocul Imitației (The Imitation Game).



## Ce este testul Turing?

---

În lucrarea „**Computing Machinery and Intelligence**”, Turing nu folosește termenul „Testul Turing” în mod explicit, ci introduce ceea ce el numește „jocul imitației” (the imitation game), un experiment ipotetic pentru a evalua dacă o mașină poate imita comportamentul unui om într-un mod convingător.

Această idee a devenit ulterior cunoscută sub numele de Testul Turing și este considerată unul dintre fundamentele teoretice ale inteligenței artificiale.

Testul Turing este un experiment de gândire propus în 1950 de matematicianul britanic Alan Turing, cu scopul de a răspunde la întrebarea: „Pot mașinile să gândească?”

Prin acest test, o persoană interacționează prin mesaje scrise cu un interlocutor uman și cu un program de computer, fără să știe care este care. Dacă evaluatorul nu poate distinge cu suficientă siguranță răspunsurile omului de cele ale mașinii, se consideră că

programul a trecut testul și demonstrează un nivel de inteligență comparabil cu cel uman.

Testul Turing a devenit un reper important în filozofia minții, inteligența artificială și informatică, inspirând generații de cercetători să dezvolte sisteme capabile să simuleze conversații naturale.

Printre primele programe care au încercat să treacă testul se numără **Eliza**. Astăzi, modelele de limbaj de mari dimensiuni (**LLM**) reprezintă o continuare modernă a acestei idei.

Eliza funcționa prin potrivirea unor șabloane de text și reformularea replicilor utilizatorului sub forma unor întrebări sau afirmații. Cel mai cunoscut script al său a fost DOCTOR, care imita un psihoterapeut rogerian și dădea impresia că înțelege și răspunde empatic.

Deși era un program simplu, Eliza a demonstrat că interacțiunile bazate pe reguli pot crea iluzia unei conversații inteligente, inspirând cercetări ulterioare în domeniul procesării limbajului natural și al chatbot-urilor.

**Testul Turing Total**, propus de Stevan Harnad în 1989, este o extindere a testului Turing clasic care adaugă cerința ca un sistem inteligent artificial să posede nu doar capacități lingvistice, ci și abilități senzoriale și motorii ca un robot. Spre deosebire de varianta originală, care se bazează exclusiv pe imitarea conversației umane, testul total presupune că inteligența reală necesită un „corp” capabil să perceapă și să acționeze în lume. Astfel, pentru a demonstra că înțelege cu adevărat conceptele despre care vorbește, un sistem trebuie să fie ancorat în experiența directă, nu doar să manipuleze simboluri abstracte.

O întrebare fundamentală este dacă calculatoarele sau roboții pot avea conștiință. De la această ipoteză a pornit Alan Turing când a conceput testul său. Testul Turing Total duce această discuție mai departe, punând în centru întrebarea: ce este conștiința? Poate că ea se reduce, în esență, la autonomie.

## Ce este un LLM?

---

Modelele de limbaj de mari dimensiuni (Large Language Models – LLM) sunt rețele neuronale antrenate pe cantități vaste de text pentru a înțelege și a genera limbaj natural.

Aceste modele funcționează prin prezicerea următorului cuvânt în funcție de contextul dat, reușind să producă propoziții coerente, să răspundă la întrebări și să simuleze conversații asemănătoare celor umane.

Arhitectura unui Large Language Model (LLM):

### 1. Input textual

Modelul primește ca intrare un text brut, introdus de utilizator.

### 2. Tokenizare

Textul este segmentat în unități lingvistice fundamentale numite tokeni.

### 3. Embedding

Fiecărui token i se atribuie un vector numeric într-un spațiu de dimensiune fixă, facilitând prelucrarea matematică ulterioară.

### 4. Straturi Transformer

Tokenii vectorizați sunt procesați printr-o arhitectură profundă compusă din mai multe straturi, care includ:

- **Atenție multi-head (self-attention)** – captează relațiile contextuale dintre tokeni.
- **Rețele neuronale feed-forward** – aplică transformări non-liniare.
- **Normalizare și conexiuni reziduale** – îmbunătățesc stabilitatea și fluxul de informație în rețea.

### 5. Head de ieșire

Un strat liniar urmat de funcția **softmax** generează o distribuție de probabilitate asupra vocabularului, indicând continuările posibile ale secvenței.

## 6. Decodare autoregresivă

Modelul generează text în mod secvențial, prezicând fiecare token următor pe baza contextului anterior, într-un proces autoregresiv.

Printre cele mai cunoscute LLM se numără GPT (Generative Pre-trained Transformer), utilizat în ChatGPT, BERT, Gemini și PaLM.

Datorită capacităților lor impresionante, LLM-urile au numeroase aplicații practice, de la chatboti și sisteme de asistență virtuală până la traduceri automate, rezumate de text și analiză semantică.

În cadrul progreselor recente din inteligența artificială, agenții capabili să genereze cod reprezintă, în mod justificat, unele dintre cele mai semnificative inovații. Printre exemplele relevante se numără **GitHub Copilot** și **Gemini Code Assist**, care ilustrează aplicabilitatea acestor sisteme în automatizarea procesului de programare.

**Andrej Karpathy** a popularizat expresia **Vibe Coding** printr-o postare virală pe X (fostul Twitter). El a fost profesor la Stanford și a lucrat în trecut la OpenAI și Tesla. Vibe Coding este o abordare modernă a dezvoltării software cu LLM care pune accent pe creativitate, colaborare și experiență plăcută în procesul de programare. Prin utilizarea unor unelte intuitive numite agenți și a unui mediu de lucru prietenos, Vibe Coding încurajează dezvoltatorii să scrie cod cu entuziasm și să experimenteze liber, reducând stresul și sporind productivitatea.

Vibe Coding poate fi privit ca un altfel de test Turing în care interlocutorul e un programator care scrie cod.

## Bibliografie

---

1. Prezentarea lui Andrej Karpathy la AI Startup School  
– <https://youtu.be/LCEmiRjPEtQ>
2. Turing, A. M. (1950). Computing Machinery and Intelligence. Mind, 59(236), 433–460.

## Resurse

---

1. Prezentare – <https://github.com/adavidoaiei/Prezentare-master>
2. Simulator mașină Turing în HTML și JavaScript  
– <https://github.com/adavidoaiei/masina-turing>