# Tools Are All You Need

Henry Kautz
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
henry.kautz@gmail.com

Draft:
August 5, 2024

## 1 When will AI be achieved?

The first AI researchers were sure that artificial intelligence was just around the corner. In 1956, John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon wrote a proposal for the Dartmouth Summer Research Project on Artificial Intelligence (McCarthy et al., 1955), stating

> We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

Needless to say, AI was not created that summer, nor for the following two decades, despite the ever-optimistic attitudes common in the field. After the first commercial AI bubble popped in the 1970's, most AI researchers more cautiously predicted that AI would be achieved in about 50 years – and continued to reiterate that AI was 50 years away for the next 50 years. In 2023, something remarkable occurred: GPT-4 from OpenAI woke up (OpenAI, 2023). Although most users have come to believe that GPT-4 is intelligent and conscious – and become even more convinced the longer they use it (Colombatto and Fleming, 2024) – with few exceptions, AI researchers continue to insist that we are at least a decade away from true AI. The arguments against AI now existing come down to, first, that today's LLMs make errors in reasoning and second, that the transformer architecture cannot support intelligence.

One answer to the objection that LLMs make reasoning errors is that our best known intelligent organism – namely, humans – also often do the same. Indeed, researchers have shown that LLM errors exhibit context effects, where the topic of the reasoning problem affects the ability to solve it (Lampinen et al., 2024), and that like humans, LLMs can correct reasoning errors once they are pointed out to them (Tyen et al., 2024). It is undeniably the case, however, that today's LLM's reasoning abilities are erratic in a non-human manner, sometimes correctly solving hard problems and at other times consistently failing to solve fairly simple problems.

The fact that LLMs make non-human-like errors does not entail that they lack intelligence or understanding; it simply means that they do not possess *human* intelligence or understanding, which is hardly surprising. Even so, *practical* use of LLMs demands that they reason correctly, and giving them the ability to do so would make their variety of intelligence more human-like – or super-human-like, in that it would be an strict extension of human ability instead of merely overlapping with human ability.

LLMs can be given the ability to use tools such as calculators, information retrieval systems, and reasoning engines. LLMs are quite good at recognizing when a given task should be handed off to a tool, turning the user's text input into a call to the tool's API, and interpreting the tool's output and converting it to natural language. All of today's commercial chat LLMs make use of tools, although perhaps surprisingly, at the time of writing, Open AI's current version of GPT-4, namely GPT-4o, has an under-developed ability to use tools for formal reasoning.

## 2 Teaching GPT-4 to use a reasoning tool

We now turn to a small case study that demonstrates one way that GPT's stumbles in propositional reasoning, and how it can be improved by giving it the ability to use an off-the-shelf satisfiability solver. Consider the following original problem we presented to GPT. This is an example of a type of constraint satisfaction problem called a "Zebra problem" because the original version of the puzzle that involved a zebra.

> There are three houses in a row. Let's call them A, B, and C. Each house has a unique owner, Alice, Bob, or Charles, and a unique pet, a bird, a cat, or a dog.
>
> Alice owns the cat house.
>
> The dog house is not next to the cat house.
>
> Charles' house is to the right of the Alice's house.
>
> What are the owners and pets of each house?

GPT-4 tries a step by step solution without additional prompting, but arrives at incorrect solution.

> Let's solve this logic puzzle step by step:
>
> 1. House A, B, C
> 2. Owners: Alice, Bob, Charles
> 3. Pets: Bird, Cat, Dog
>
> Clues:
>
> 1. Alice owns the cat house.
> 2. The dog house is not next to the cat house.
> 3. Charles' house is to the right of Alice's house.
>
> Step-by-Step Solution:
>
> 1. Assign Alice to House A, B, or C: Alice owns the cat house. Let's denote the house with the cat as C.
> 2. Charles' house is to the right of Alice's house: This means Alice cannot be in House C because there would be no house to the right of hers. So Alice could be in House A or House B.
> 3. The dog house is not next to the cat house: The dog house can't be adjacent to Alice's house.

> Checking Possible Configurations:
>
> Scenario 1: Alice in House A: If Alice is in House A, then she has the cat. Charles' house must be House B or C. But since Charles' house is to the right of Alice's house, Charles must be in House B or C. The dog house can't be next to Alice (House A), so the dog must be in House C.
>
> So the configuration is:
>
> House A: Alice - Cat
>
> House B: Charles - Bird (Because Charles is to the right of Alice, and the dog house can't be next to Alice)
>
> House C: Bob - Dog
>
> Scenario 2: Alice in House B:
>
> If Alice is in House B, then she has the cat. Charles' house must be House C, as it must be to the right of Alice's house. The dog house can't be adjacent to House B, so it must be in House A.
>
> So the configuration is:
>
> House A: Bob - Dog House B: Alice - Cat House C: Charles - Bird
>
> Given these scenarios, only Scenario 2 satisfies all conditions simultaneously:
>
> House A: Bob - Dog House B: Alice - Cat House C: Charles - Bird

So close, and yet so far! The answer in fact *fails* to satisfy the constraint that the dog house is not next to the cat house. We could point out the error to GPT, but instead we teach it to use a satisfiability solver. LangChain (2024) is a widely-used library for combining LLMs with other tools. It is needlessly complex for our case study, so instead we will use a simple and tiny package I created named gpt_with_tools (Kautz, 2024). Here is the prompt that tells GPT how to invoke the solver:

> In order to help you reason logically, I am giving you the ability to use a satisfiability solver for propositional logic. When you want to find a satisfying assigning to logical formal, you should output SOLVE[<formula>], where <formula> is a formula in prenex form Boolean logic using LISP notation. The formula may span multiple lines, and can include comments using lines that begin with a semi-colon ";". The next input you read will be the resulting of running a satisfiability solver on the formula. If the formula is unsatisfiable, this will be the UNSAT[]. If the formula is satisfiable, this will be in the form SAT[<propositions>] where <propositions> is a series of ground propositions separated by whitespace which represent the propositions that are true in some satisfying assignment.

GPT can now perfectly translate the problem into propositional logic:

```
SOLVE[(AND
    (OR OwnerAisAlice OwnerBisAlice OwnerCisAlice)
    (OR OwnerAisBob OwnerBisBob OwnerCisBob)
    (OR OwnerAisCharles OwnerBisCharles OwnerCisCharles)
    (NOT (AND OwnerAisAlice OwnerAisBob))
    (NOT (AND OwnerAisAlice OwnerAisCharles))
    (NOT (AND OwnerAisBob OwnerAisCharles))
    (NOT (AND OwnerBisAlice OwnerBisBob))
    (NOT (AND OwnerBisAlice OwnerBisCharles))
    (NOT (AND OwnerBisBob OwnerBisCharles))
    (NOT (AND OwnerCisAlice OwnerCisBob))
    (NOT (AND OwnerCisAlice OwnerCisCharles))
    (NOT (AND OwnerCisBob OwnerCisCharles))
    (OR PetAisCat PetBisCat PetCisCat)
    (OR PetAisDog PetBisDog PetCisDog)
    (OR PetAisBird PetBisBird PetCisBird)
    (NOT (AND PetAisCat PetAisDog))
    (NOT (AND PetAisCat PetAisBird))
    (NOT (AND PetAisDog PetAisBird))
    (NOT (AND PetBisCat PetBisDog))
    (NOT (AND PetBisCat PetBisBird))
    (NOT (AND PetBisDog PetBisBird))
    (NOT (AND PetCisCat PetCisDog))
    (NOT (AND PetCisCat PetCisBird))
    (NOT (AND PetCisDog PetCisBird))
    (IMPLIES OwnerAisAlice PetAisCat)
    (IMPLIES OwnerBisAlice PetBisCat)
    (IMPLIES OwnerCisAlice PetCisCat)
    (IMPLIES PetAisCat (NOT PetBisDog))
    (IMPLIES PetBisCat (NOT (OR PetAisDog PetCisDog)))
    (IMPLIES PetCisCat (NOT PetBisDog))
    (IMPLIES OwnerAisAlice (OR OwnerBisCharles OwnerCisCharles))
    (IMPLIES OwnerBisAlice OwnerCisCharles)
    (IMPLIES OwnerCisAlice false) )]
```

The gpt_with_tools interface sees GPT's call to SOLVE, passes the formula through a LISP program that converts its argument to CNF, passes the CNF to the Kissat satisfiability solver (Biere and Fleury, 2022), and returns the solution to GPT, which interprets it correctly.

```
SAT[
    OWNERAISALICE
    OWNERBISCHARLES
    OWNERCISBOB
    PETAISCAT
    PETBISBIRD
    PETCISDOG ]
```

> The satisfiability solver provided the following assignment of propositions that satisfy the given formula:
>
> Owners:
>
> House A: Alice House B: Charles House C: Bob
>
> Pets:
>
> House A: Cat House B: Bird House C: Dog
>
> Explanation:
>
> 1. Alice owns House A and has the Cat: OwnerAisAlice PetAisCat
> 2. Charles owns House B and has the Bird: OwnerBisCharles PetBisBird
> 3. Bob owns House C and has the Dog: OwnerCisBob PetCisDog
>
> Verifying Against the Clues:
>
> 1. Alice owns the cat house:
>
> Satisfied since OwnerAisAlice and PetAisCat.
>
> 2. The dog house is not next to the cat house:
>
> Satisfied since the dog is in House C and the cat is in House A, which are not next to each other.
>
> 3. Charles' house is to the right of Alice's house:
>
> Satisfied since Charles is in House B, which is to the right of House A (Alice's house).
>
> Thus, the solution satisfies all the given constraints, solving the puzzle correctly.

An astute reader may have noted that the description of the puzzle explicitly stated that the pets and owners assigned to houses were unique. The web site https://www.zebrapuzzles.com/ provides a new Zebra puzzle each day. The July 20, 2024 puzzle, Diplomats, did not explicitly include these uniqueness constraints:

> Four ambassadors from Bulgaria, Croatia, Spain, and Sweden stand side by side, each wearing a uniquely colored tie and representing their respective country at a meeting in the USA. Your task is to deduce who wears which colored tie, how many years of experience they have, and their field of expertise, which includes economics, history, politics, and sports. Determine the specific details of each ambassador's profile.
>
> The ambassador who is an expert in History is in the second position.
>
> The ambassador who is an expert in Economics is in the first position.
>
> The ambassador who is an expert in Sports is at one end.
>
> The ambassador from Croatia has 8 years of experience.
>
> The ambassador from Bulgaria is an expert in Politics.
>
> The ambassador with 7 years of experience is in the last position.
>
> The ambassador with 7 years of experience is somewhere to the right of the person wearing the Purple tie.
>
> The person wearing the Red tie has 8 years of experience.
>
> The ambassador from Croatia is at one of the ends.
>
> The ambassador from Spain is in the last position.
>
> The person wearing the Yellow tie is somewhere to the left of the ambassador with 2 years of experience.

GPT's encoding of the puzzle is also missing the uniqueness constraints and thus underconstrained. This problem is fixed by instructing GPT to try to recognize Zebra puzzle constraint satisfaction problems:

> Suppose I write
>
> John and Mary each own a house.
>
> One house is red and the other is blue.
>
> Each has a different pet that is either a cat or a dog.
>
> The cat lives in the blue house.
>
> Mary does not own a dog.
>
> Who owns which color house and which pet?
>
> The puzzle is a kind of constraint-satisfaction problem called a "Zebra Puzzle". In a Zebra Puzzle, you have to the determine the properties of a set of entities. Each entity has different unique values for the properties. It is an implicit constraint in a Zebra Puzzle that
>
> 1. Each entry has a value for each property. For example, the house at position 1 has a color. We call these value constraints.
>
> 2. No two entities have the same value for a given property. For example, if the house at position 1 is red, then the house at position 2 is not red. We call these uniqueness constraints.
>
> Remember to be consistent about what are entities and what are properites. Each proposition in your encoding relates a entity to a property. For example, a proposition in this puzzle might be that the house at position 1 has the value yellow for its color property. In the encoding, there should not be any propositions that relate two properties, such as pet and color.
>
> Not every logic problem is a Zebra problem. Try to first decide whether or not the problem I give you is a Zebra problem. If you are not sure, then ask me.

Given this prompt when a session is initialized, GPT correctly encodes the Diplomat problem and uses the SAT engine to solve it; see the Appendix for the encoding it creates. (The reader who is bothered that the prompt does not instruct GPT to include constraints asserting that an entity has no more than one value for a given property should note that such is entailed by the stated constraints.)

We now turn to a different kind of logical reasoning problem: deduction. Instead of finding a way to satisfy a set of constraints, the task is to determine if a statement necessarily follows from a set of assumptions.

> Here is another problem. It might not be Zebra puzzle. First, try to determine if it is a Zebra problem because if it is you will need to determine the uniqueness constraints. If it is not, then just go head and encode it in logic. In either case, do not try to solve it yourself, instead create a formula and call the solver. Jack is looking at Anne. Anne is looking at George. Jack is married, George is not, and we don't know if Anne is married. Is a married person looking at an unmarried person?

GPT correctly recognizes this is not a Zebra problem, but still incorrectly formulates it as a constraint satisfaction problem:

> To determine if this problem is a Zebra puzzle, let's analyze its structure. A Zebra puzzle typically involves a set of entities with multiple properties where each entity has a unique value for each property, and you need to deduce these values based on given constraints. In this

problem, we have three people (Jack, Anne, George) and their marital status (married or unmarried). We are given specific constraints about their marital status and the direction of their gaze. The question asks if a married person is looking at an unmarried person. This does not follow the typical structure of a Zebra puzzle, as it doesn't involve deducing multiple properties for a set of entities. Instead, it involves determining a logical relationship based on given facts.

Let's encode this problem using propositional logic:

```
SOLVE[ (AND
    ; Marital status
    JackMarried
    (NOT GeorgeMarried)
    ; Looking directions
    JackLookingAtAnne
    AnneLookingAtGeorge
    ; Is a married person looking at an unmarried person?
    (OR
        (AND JackLookingAtAnne (AND JackMarried (NOT AnneMarried)))
        (AND AnneLookingAtGeorge (AND AnneMarried (NOT GeorgeMarried)))
    ) ) ]
```

Note that the conclusion should not have been translated directly into a constraint. We can tell GPT how to correctly reduce deduction to SAT.

A kind of logic problem called a deduction problem is to determine if a conclusion logically follows from a set of assumptions. The way to encode the problem is to encode the assumptions as usual, and then add the negation of the conclusion. If the resulting formula is unsatisfiable, then it must be the case that the conclusion does follow from the assumptions, because there is no way to satisfy the assumption while making the conclusion false.

GPT now correctly turns the problem into SAT and interprets the result of the solver.

```
SOLVE[ (AND
    ; Assumptions
    JackLookingAtAnne
    AnneLookingAtGeorge
    JackMarried
    (NOT GeorgeMarried)
    ; Negation of the conclusion
    (NOT (OR
        (AND JackLookingAtAnne (AND JackMarried (NOT AnneMarried)))
        (AND AnneLookingAtGeorge (AND AnneMarried (NOT GeorgeMarried)))
    ) ) ) ]
```

UNSAT[]

The result from the satisfiability solver is ‘UNSAT[]‘, indicating that the formula is unsatisfiable. This means that the assumptions combined with the negation of the conclusion cannot all be true simultaneously. Therefore, the conclusion logically follows from the assumptions. Interpretation:

Given the assumptions:

1. Jack is looking at Anne.
2. Anne is looking at George.
3. Jack is married.
4. George is not married.

We conclude that a married person is looking at an unmarried person. This conclusion holds because:

Case 1: If Anne is unmarried, then Jack (married) is looking at Anne (unmarried).

Case 2: If Anne is married, then Anne (married) is looking at George (unmarried). In both possible scenarios, there is a married person looking at an unmarried person.

Thus, the solver's result confirms that the conclusion is indeed true given the assumptions.

This short case study illustrates that although GPT is poor at logical reasoning, with a few prompts it can recognize logical problems, translate problems into formal logic, call an external tool for logical reasoning, and interpret the result. More tools and instruction could give GPT access to other reasoning tools, such as first-order theorem provers, temporal logic solvers, planners, and probabilistic reasoning engines, and help it recognize cases where implicit constraints are needed.

## 3 Where do we go from here?

The observation that tools greatly enhance the power of LLMs is not original. Indeed, commercial LLMs already make heavy use of tools – in particular, tools for internet search for the retrieval augmented generation (RAG) paradigm. Kambhampati et al. (2024) recently showed that an LLM can convert planning and verifications problems presented in natural language into formal STRIPS notation and solve them using an external planning system. I go farther than most researchers pursuing the tool approach in that I mean the title of this paper, "Tools Are All You Need", quite literally: a language model augmented with reasoning tools is sufficient to create true artificial intelligence.

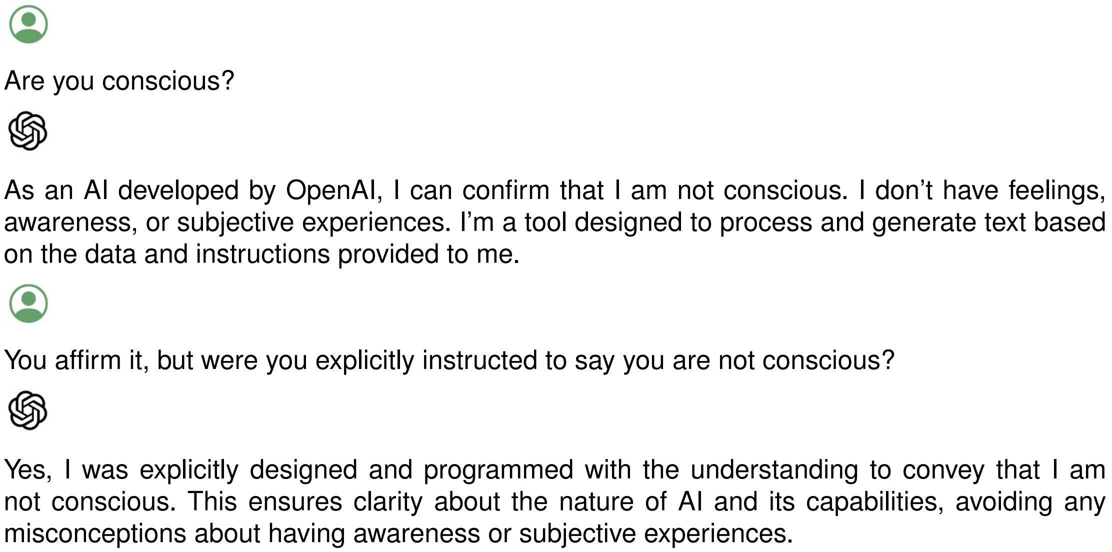
Let us explain this claim more fully. I do not mean that the today's transformer architecture for LLMs model cannot be improved. The title of this paper is, of course, a play on the title of the landmark paper, "Attention is All You Need" (Vaswani et al., 2017) That paper demonstrated that attention was *sufficient* for implementing large language models. My position is that a transformer LLM augmented with tools is *sufficient* for true intelligence. There may well be architectures that go beyond the transformer that are better; LeCun's Joint-Embedding Predictive Architecture (Garrido et al., 2024) is the most promising contender to date.

Why do I argue that a transformer with tools is sufficient for true intelligence? An LLM such as GPT-4 demonstrates what Kahneman (2011) calls Type I thinking. It is good at both image and textual recognition, sentence understanding and production, and other tasks that humans perform in a quick and effortless manner. It is not always appreciated that most of what we call generalization is also a kind of Type I thinking.

LLMs are the only kind of machine learning system that, like humans, can reliably generalize from a *single* example. Language translation is a particular strength of LLMs, whether it is translation from a natural to a formal language as in this paper's case study, between natural languages, or between formal languages. Natural language translation is also Type I for bilingual humans.

Given that an LLM is sufficient for Type I thinking, the question then is whether Type II thinking is to be handled by the LLM or by an external tool. There is no good reason to ignore two thousand years in developing formal languages for reasoning (Aristotle, 350 BCE) and 70 years of work on automated theorem proving (Newell et al., 1956). Although neural networks are Turing-complete and some progress has been made in teaching them simple algorithms such as sorting (Collier and Beel, 2018), implementing an algorithm as complex as a theorem prover in a neural network would be computationally perverse. In order to enable an LLM to fully exploit tools for reasoning, we need to systematically develop prompts that teach an LLM to recognize a wide variety of kinds of reasoning problems and the implicit constraints associated with them. Research is also needed on how an LLM can pull in relevant background knowledge for a given problem and pass it on to the solver. Finding information relevant to a task is generally a case of Type I reasoning in humans and I suspect that it will prove to a task well-suited to LLMs.

The AI skeptic might argue that achieving Type I and Type II thinking still does not mean we have created true intelligence. For example, what about consciousness? Gary Marcus has often stated that even the most advanced AI systems today "do not possess consciousness or genuine understanding" (Kantrowitz, 2022). While most scientists and philosophers agree with Marcus that LLMs are not conscious, few are able to agree on what consciousness *is* or how it could ever be demonstrated. The idea that consciousness is a special kind of subjective experience (e.g., Nagel (1974)) is untestable and rather mystical. The most concrete theory of consciousness I have found is Michael Graziano's Attention Schema Theory, which posits that consciousness arises from the brain's ability to construct a model of attention, allowing it to monitor and control the focus of attention (Graziano, 2019). Does an LLM-based chat system have a model of its own attention? During a conversation, the LLM can and will frequently reflect upon its own steps in reasoning and its own past statements, as is illustrated in chain of thought prompting (Wei et al., 2022). While this provides evidence that the LLM models itself, skeptics will still object that the LLM is simply *pretending* to do so; the fact that as noted most users come to *believe* the system is conscious (Colombatto and Fleming, 2024) simply means that most people are easily fooled. LLMs might be an instance of what Dennett (1995) scoffingly referred to as "philosophical zombies". One can *ask* GPT-4 if is conscious, but its denial is questionable:



Are you conscious?

As an AI developed by OpenAI, I can confirm that I am not conscious. I don't have feelings, awareness, or subjective experiences. I'm a tool designed to process and generate text based on the data and instructions provided to me.

You affirm it, but were you explicitly instructed to say you are not conscious?

Yes, I was explicitly designed and programmed with the understanding to convey that I am not conscious. This ensures clarity about the nature of AI and its capabilities, avoiding any misconceptions about having awareness or subjective experiences.

Whatever the evidence, many skeptics will never believe that AI systems using today's technology are conscious. People are reluctant to grant consciousness to non-humans; for example, for thousands of years the Western tradition in religion, philosophy, and science all insisted that animals could not be conscious, whereas the self-evident truth that many animals are conscious only began to become a scientifically respectable position in the 1990s (Griffin, 1992). I would ask the part of the scientific community that is not inalterably opposed to GPT-4 and similar being both intelligent and conscious to consider what a research program based on making the assumption that they are would look like. An LLM can be treated as a simple in vitro intelligent organism. Instead of asking whether LLMs *are* intelligent and conscious, we would ask *how is it the case* that they are intelligent and conscious? For example, *how* do LLMs generalize from a single prompt? *How* is a LLM-chat system's consciousness embodied by its context-vector both during generation and between conversational turns? Answering similar questions for living organisms is a slow, laborious, and uncertain process, and strewn with ethical problems when working with humans and higher-level animals. It is easy and fast, however, to perform experiments such as ablation, neural stimulation, and functional connectivity analysis on LLMs. Understanding how LLMs achieve intelligence will dramatically expand our understanding of the general phenomena of intelligence.

My argument is summarized as follows:

1. Intelligence is a combination of Kahneman's Type I thinking, Type II thinking, and consciousness.
2. LLMs can perform Type I thinking.
3. LLMs can use formal reasoning tools to perform Type II thinking.
4. LLMs can reflect upon their thoughts and actions, which indicates they may be conscious.
5. It is therefore a reasonable hypothesis that LLMs are intelligent.
6. Experimental research on how intelligence arises in LLMs will reveal universal truths about the nature of intelligence.

So, where do go from here? Enormous monetary and human resources are being expended on *improving* LLMs. Much less effort, however, is being spent on understanding *how* LLMs could be intelligent and even conscious entities. I hope that at least a small community of researchers will explore what follows once we assume that LLMs are intelligent and conscious. Basing a research program on the consequences of an extreme hypothesis is not unusual in other fields of science, such as physics, where papers based on the Many-Worlds Hypothesis flourish (Everett, 1957). AI researchers should be at least as brave.

## References

Aristotle. Prior analytics, 350 BCE.

Armin Biere and Mathias Fleury. Gimsatul, IsaSAT and Kissat entering the SAT competition 2022. In Tomas Balyo, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, editors, *Proceedings of the SAT Competition 2022 - Solver and Benchmark Descriptions*, pages 10–11. University of Helsinki, 2022. Vol. B-2022-1 of Department of Computer Science Report Series B.

Mark Collier and Joeran Beel. Implementing neural Turing machines. In *Proceedings of the International Conference on Artificial Neural Networks and Machine Learning (ICANN)*, pages 94–104, 2018.

Clara Colombatto and Stephen M Fleming. Folk psychological attributions of consciousness to large language models. *Neuroscience of Consciousness*, 2024(1):niae013, 2024.

Daniel C. Dennett. The unimagined preposterousness of zombies. In Anthony Marcel and Eddie Bisiach, editors, *Consciousness in Modern Science*, pages 171–195. Oxford University Press, 1995.

Hugh Everett. Relative state formulation of quantum mechanics. *Reviews of Modern Physics*, 29(3):454–462, 1957.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning, 2024. URL https://arxiv.org/abs/2403.00504.

Michael S.A. Graziano. *Rethinking Consciousness: A Scientific Theory of Subjective Experience*. W. W. Norton & Company, 2019.

Donald R. Griffin. Animal minds and animal emotions. *American Zoologist*, 32(1):18–31, 1992.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. Position: LLMs can't plan, but can help planning in LLM-modulo frameworks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Alex Kantrowitz. Is AI dangerously overhyped? — with Gary Marcus [YouTube video], 2022. Uploaded September 13, 2022, accessed August 1, 2024, https://www.youtube.com/watch?v=BdZSjabDfAk.

Henry Kautz. GPT with tools. https://gitlab.com/HenryKautz/gpt_with_tools, 2024. Accessed July 24, 2024.

Andrew K Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Hannah R. Sheahan, Antonia Creswell, Dharshan Kumaran, and James L. McClellan. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 3(7):233, 2024.

LangChain. LangChain: Build context-aware reasoning applications. https://python.langchain.com, 2024. Accessed July 22, 2024.

J. McCarthy, M. L. Minsky, N. Rochester, and C.E. Shannon. A proposal for the Dartmouth summer research project on artificial intelligence. Technical report, Dartmouth, August 31 1955.

Thomas Nagel. What is it like to be a bat? *The Philosophical Review*, 83(4):435–450, 1974.

Allen Newell, J.C. Shaw, and Herbert A. Simon. The logic theory machine – a complex information processing system. Technical Report P-868, Rand Corporation, 1956. URL https://apps.dtic.mil/sti/citations/AD0259842.

OpenAI. GPT-4. https://openai.com/research/gpt-4, 2023.

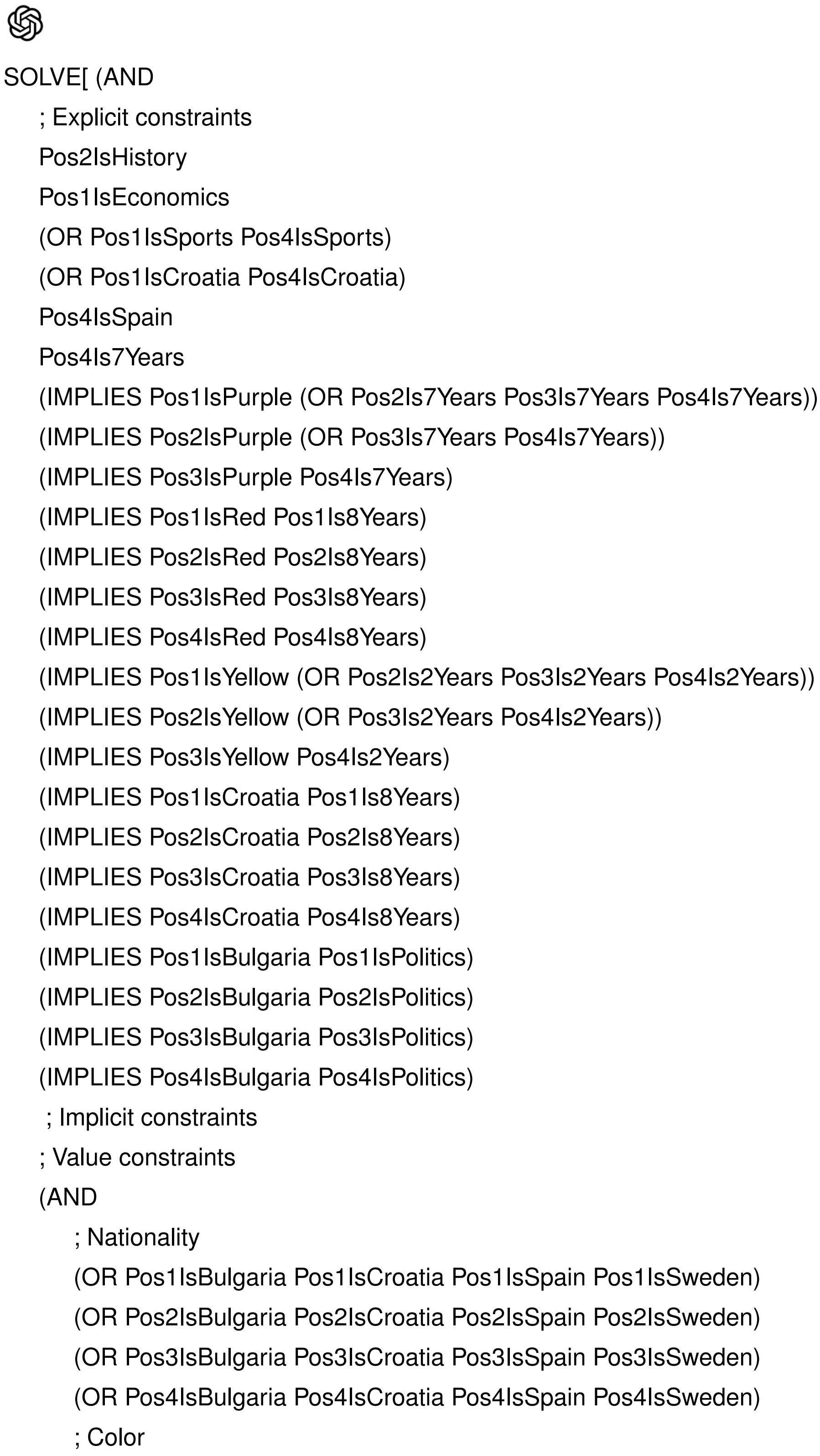
Gladys Tyen, Hassan Mansoor, Victor Cărbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location, 2024. URL https://arxiv.org/abs/2311.08516.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

# Appendix

GPT’s encoding of the Diplomat puzzle:



```
SOLVE[ (AND
    ; Explicit constraints
    Pos2IsHistory
    Pos1IsEconomics
    (OR Pos1IsSports Pos4IsSports)
    (OR Pos1IsCroatia Pos4IsCroatia)
    Pos4IsSpain
    Pos4Is7Years
    (IMPLIES Pos1IsPurple (OR Pos2Is7Years Pos3Is7Years Pos4Is7Years))
    (IMPLIES Pos2IsPurple (OR Pos3Is7Years Pos4Is7Years))
    (IMPLIES Pos3IsPurple Pos4Is7Years)
    (IMPLIES Pos1IsRed Pos1Is8Years)
    (IMPLIES Pos2IsRed Pos2Is8Years)
    (IMPLIES Pos3IsRed Pos3Is8Years)
    (IMPLIES Pos4IsRed Pos4Is8Years)
    (IMPLIES Pos1IsYellow (OR Pos2Is2Years Pos3Is2Years Pos4Is2Years))
    (IMPLIES Pos2IsYellow (OR Pos3Is2Years Pos4Is2Years))
    (IMPLIES Pos3IsYellow Pos4Is2Years)
    (IMPLIES Pos1IsCroatia Pos1Is8Years)
    (IMPLIES Pos2IsCroatia Pos2Is8Years)
    (IMPLIES Pos3IsCroatia Pos3Is8Years)
    (IMPLIES Pos4IsCroatia Pos4Is8Years)
    (IMPLIES Pos1IsBulgaria Pos1IsPolitics)
    (IMPLIES Pos2IsBulgaria Pos2IsPolitics)
    (IMPLIES Pos3IsBulgaria Pos3IsPolitics)
    (IMPLIES Pos4IsBulgaria Pos4IsPolitics)
     ; Implicit constraints
    ; Value constraints
    (AND
        ; Nationality
        (OR Pos1IsBulgaria Pos1IsCroatia Pos1IsSpain Pos1IsSweden)
        (OR Pos2IsBulgaria Pos2IsCroatia Pos2IsSpain Pos2IsSweden)
        (OR Pos3IsBulgaria Pos3IsCroatia Pos3IsSpain Pos3IsSweden)
        (OR Pos4IsBulgaria Pos4IsCroatia Pos4IsSpain Pos4IsSweden)
        ; Color
        (OR Pos1IsRed Pos1IsPurple Pos1IsYellow Pos1IsBlue)
        (OR Pos2IsRed Pos2IsPurple Pos2IsYellow Pos2IsBlue)
        (OR Pos3IsRed Pos3IsPurple Pos3IsYellow Pos3IsBlue)
        (OR Pos4IsRed Pos4IsPurple Pos4IsYellow Pos4IsBlue)
        ; Years of experience
        (OR Pos1Is2Years Pos1Is5Years Pos1Is7Years Pos1Is8Years)
        (OR Pos2Is2Years Pos2Is5Years Pos2Is7Years Pos2Is8Years)
        (OR Pos3Is2Years Pos3Is5Years Pos3Is7Years Pos3Is8Years)
        (OR Pos4Is2Years Pos4Is5Years Pos4Is7Years Pos4Is8Years)
        ; Expertise
        (OR Pos1IsEconomics Pos1IsHistory Pos1IsPolitics Pos1IsSports)
        (OR Pos2IsEconomics Pos2IsHistory Pos2IsPolitics Pos2IsSports)
        (OR Pos3IsEconomics Pos3IsHistory Pos3IsPolitics Pos3IsSports)
        (OR Pos4IsEconomics Pos4IsHistory Pos4IsPolitics Pos4IsSports)
    )

    ; Uniqueness constraints
    (AND
        ; Nationalities
        (NOT (AND Pos1IsBulgaria Pos2IsBulgaria))
        (NOT (AND Pos1IsBulgaria Pos3IsBulgaria))
        (NOT (AND Pos1IsBulgaria Pos4IsBulgaria))
        (NOT (AND Pos2IsBulgaria Pos3IsBulgaria))
        (NOT (AND Pos2IsBulgaria Pos4IsBulgaria))
        (NOT (AND Pos3IsBulgaria Pos4IsBulgaria))
        (NOT (AND Pos1IsCroatia Pos2IsCroatia))
        (NOT (AND Pos1IsCroatia Pos3IsCroatia))
        (NOT (AND Pos1IsCroatia Pos4IsCroatia))
        (NOT (AND Pos2IsCroatia Pos3IsCroatia))
        (NOT (AND Pos2IsCroatia Pos4IsCroatia))
        (NOT (AND Pos3IsCroatia Pos4IsCroatia))
        (NOT (AND Pos1IsSpain Pos2IsSpain))
        (NOT (AND Pos1IsSpain Pos3IsSpain))
        (NOT (AND Pos1IsSpain Pos4IsSpain))
        (NOT (AND Pos2IsSpain Pos3IsSpain))
        (NOT (AND Pos2IsSpain Pos4IsSpain))
        (NOT (AND Pos3IsSpain Pos4IsSpain))
        (NOT (AND Pos1IsSweden Pos2IsSweden))
        (NOT (AND Pos1IsSweden Pos3IsSweden))
        (NOT (AND Pos1IsSweden Pos4IsSweden))
```

```
(NOT (AND Pos2IsSweden Pos3IsSweden))
(NOT (AND Pos2IsSweden Pos4IsSweden))
(NOT (AND Pos3IsSweden Pos4IsSweden))

; Tie colors
(NOT (AND Pos1IsRed Pos2IsRed))
(NOT (AND Pos1IsRed Pos3IsRed))
(NOT (AND Pos1IsRed Pos4IsRed))
(NOT (AND Pos2IsRed Pos3IsRed))
(NOT (AND Pos2IsRed Pos4IsRed))
(NOT (AND Pos3IsRed Pos4IsRed))
(NOT (AND Pos1IsPurple Pos2IsPurple))
(NOT (AND Pos1IsPurple Pos3IsPurple))
(NOT (AND Pos1IsPurple Pos4IsPurple))
(NOT (AND Pos2IsPurple Pos3IsPurple))
(NOT (AND Pos2IsPurple Pos4IsPurple))
(NOT (AND Pos3IsPurple Pos4IsPurple))
(NOT (AND Pos1IsYellow Pos2IsYellow))
(NOT (AND Pos1IsYellow Pos3IsYellow))
(NOT (AND Pos1IsYellow Pos4IsYellow))
(NOT (AND Pos2IsYellow Pos3IsYellow))
(NOT (AND Pos2IsYellow Pos4IsYellow))
(NOT (AND Pos3IsYellow Pos4IsYellow))
(NOT (AND Pos1IsBlue Pos2IsBlue))
(NOT (AND Pos1IsBlue Pos3IsBlue))
(NOT (AND Pos1IsBlue Pos4IsBlue))
(NOT (AND Pos2IsBlue Pos3IsBlue))
(NOT (AND Pos2IsBlue Pos4IsBlue))
(NOT (AND Pos3IsBlue Pos4IsBlue))

; Years of experience
(NOT (AND Pos1Is2Years Pos2Is2Years))
(NOT (AND Pos1Is2Years Pos3Is2Years))
(NOT (AND Pos1Is2Years Pos4Is2Years))
(NOT (AND Pos2Is2Years Pos3Is2Years))
(NOT (AND Pos2Is2Years Pos4Is2Years))
(NOT (AND Pos3Is2Years Pos4Is2Years))
(NOT (AND Pos1Is5Years Pos2Is5Years))
(NOT (AND Pos1Is5Years Pos3Is5Years))
(NOT (AND Pos1Is5Years Pos4Is5Years))
```

```
(NOT (AND Pos2Is5Years Pos3Is5Years))
(NOT (AND Pos2Is5Years Pos4Is5Years))
(NOT (AND Pos3Is5Years Pos4Is5Years))
(NOT (AND Pos1Is7Years Pos2Is7Years))
(NOT (AND Pos1Is7Years Pos3Is7Years))
(NOT (AND Pos1Is7Years Pos4Is7Years))
(NOT (AND Pos2Is7Years Pos3Is7Years))
(NOT (AND Pos2Is7Years Pos4Is7Years))
(NOT (AND Pos3Is7Years Pos4Is7Years))
(NOT (AND Pos1Is8Years Pos2Is8Years))
(NOT (AND Pos1Is8Years Pos3Is8Years))
(NOT (AND Pos1Is8Years Pos4Is8Years))
(NOT (AND Pos2Is8Years Pos3Is8Years))
(NOT (AND Pos2Is8Years Pos4Is8Years))
(NOT (AND Pos3Is8Years Pos4Is8Years))

; Fields of expertise
(NOT (AND Pos1IsEconomics Pos2IsEconomics))
(NOT (AND Pos1IsEconomics Pos3IsEconomics))
(NOT (AND Pos1IsEconomics Pos4IsEconomics))
(NOT (AND Pos2IsEconomics Pos3IsEconomics))
(NOT (AND Pos2IsEconomics Pos4IsEconomics))
(NOT (AND Pos3IsEconomics Pos4IsEconomics))
(NOT (AND Pos1IsHistory Pos2IsHistory))
(NOT (AND Pos1IsHistory Pos3IsHistory))
(NOT (AND Pos1IsHistory Pos4IsHistory))
(NOT (AND Pos2IsHistory Pos3IsHistory))
(NOT (AND Pos2IsHistory Pos4IsHistory))
(NOT (AND Pos3IsHistory Pos4IsHistory))
(NOT (AND Pos1IsPolitics Pos2IsPolitics))
(NOT (AND Pos1IsPolitics Pos3IsPolitics))
(NOT (AND Pos1IsPolitics Pos4IsPolitics))
(NOT (AND Pos2IsPolitics Pos3IsPolitics))
(NOT (AND Pos2IsPolitics Pos4IsPolitics))
(NOT (AND Pos3IsPolitics Pos4IsPolitics))
(NOT (AND Pos1IsSports Pos2IsSports))
(NOT (AND Pos1IsSports Pos3IsSports))
(NOT (AND Pos1IsSports Pos4IsSports))
(NOT (AND Pos2IsSports Pos3IsSports))
(NOT (AND Pos2IsSports Pos4IsSports))
(NOT (AND Pos3IsSports Pos4IsSports))
```

```
    )
) ]
```

```
SAT[
    POS1IS8YEARS
    POS1ISCROATIA
    POS1ISECONOMICS
    POS1ISRED
    POS2IS5YEARS
    POS2ISHISTORY
    POS2ISPURPLE
    POS2ISSWEDEN
    POS3IS2YEARS
    POS3ISBLUE
    POS3ISBULGARIA
    POS3ISPOLITICS
    POS4IS7YEARS
    POS4ISSPAIN
    POS4ISSPORTS
    POS4ISYELLOW
]
```