Chatbots as Turing Machines

1st Sathaporn Hu
Faculty of Computer Science and Technology
Algoma University
Sault Ste. Marie, Canada
hus@algomau.ca

***Abstract*—Under the Church-Turing thesis (CTT), there must exist Turing machines (TM) for non-interactive computers. Therefore, certain chatbots (e.g., ChatGPT, Google Gemini) must have TM counterparts. By considering the limitations of TM, the paper identifies several issues that the chatbots encounter. First, the evaluation of chatbots is difficult due to them possessing undecidable properties associated with TMs. Secondly, since chatbots are reducible to TMs they face the same problems that many computers do. Lastly, I also argue that to achieve a more "intelligent" chatbot, artificial intelligence practitioners must consider breaking CTT and consider concepts like interaction and semiotics.**

***Index Terms*—chatbot, automata theory, Turing machine, cognitive science**

I. Introduction

The Church-Turing thesis (CTT) implies that many computer programs can be described as Turing machines (TM; see Appendix A for a full definition). I argue that modern chatbots (e.g., ChatGPT, DALL·E, and GitHub Co-Pilot) can be turned into TMs. This means they possess all weaknesses of all computers. While TMs are impractical devices, they allow us to make mathematical and formal arguments. For instance, if we demonstrate that a TM cannot solve a problem, neither can a computer. Modern chatbots are reducible to TMs because they are essentially functions that transform a string into another. While chatbots deploy neural networks and advanced mathematical techniques, these are insufficient to break the confines of CTT. To gain freedom, a chatbot must become what Turing described as a choice machine. In this work, I provide TM-based definitions for several aspects of modern chatbots that use generative AI (GAI). Then, I show that determining if a chatbot is good or not is undecidable. Furthermore, I argue that it is incapable of creativity because, unlike natural human interactions, a TM cannot modify its symbols.

It is important to note that this paper does not argue that AI evaluation is impossible. Just because an arbitrary instance of a problem is undecidable, it does not mean that specific instances are not. There are also cases where evaluations must be carried out even if we do not have good knowledge. An example of this is the field of human-computer interaction (HCI) which involves evaluations of user interfaces (UI). While it is difficult to formalize the notion of "good", we still deploy questionnaires (e.g. [1]) to better understand them.

II. Background Information: Turing Machine

A Turing Machine (TM) is an imaginary tape reader/writer. Each tape has an infinite number of cells, and each cell has a single symbol. While any symbol can be on the tape (e.g., English alphabets, Hindu-Arabic numbers, emojis), they must also be defined in the TM's own symbol set. Let Σ be the symbol set. For example, if $\Sigma = \{0, 1\}$, then the tape can only contain 0's and 1's. One of the symbols must be a "blank" symbol which signifies that the cell is empty. The TM also has a series of states and has one initial state that it must begin at. The user provides an input tape, τ. The TM, based on its current state and the symbol that it has read, performs these actions:

- **Write:** Write the same symbol that it has read, or a new symbol within its Σ.
- **Move:** Stay in the same cell, or move left or right by one cell.
- **Transition:** Stay in the same state or change its state.

The machine will repeat these operations until it reaches a halting state. Once the halting state is reached, the content on τ will become its output. Under CTT, we can assume that many algorithms must have TMs that can solve them. The definition of the TM here is based on Goldin et al [2].

A. Variations of Turing Machines

There are variants of TMs. First, a multi-track TM is a TM that can process more than one tape at once [3]. Secondly, a nondeterministic Turing machine (NTM) can produce multiple output based on a single input. This is in contrast to a TM which can only produce one answer for a single input [4]. Neither is more powerful than a TM because a normal TM (i.e. a single-track deterministic TM) can simulate both [3], [4]. Finally, a universal Turing Machine (UTM) is a TM that can simulate other TMs encoded as TM tapes called standard descriptions (SD) [5]. A modern analogue of a UTM is a computer with virtualization software that can simulate virtual computers with have been encoded into files.

III. Related Work

TMs serve as the main inspiration for this work. Utterly impractical as actual computing devices, TMs have nonetheless been used to make predictions about actual computers. TMs are particularly appealing in certain cases, because they capture the most basic notion of computation. In essence, if our argument applies to a TM, then it will apply to any other computer. For example, Cook-Levin's theorem [6] and Karp's reductions [7] argue that some problems are difficult to solve (unless $P = NP$) by using TMs at the foundation of their arguments. Some cognitive scientists like Putnam even argue that Turing machines can be used to represent cognition itself [8]. When we rely on TMs to make argument, we are essentially making an argument based in functionalism. We are making arguments based on what something *does*, and not based on what it *is* [9]. It is important to note that Turing machines may not be able to present all types of modern computers [10]. For instance, a Turing machine cannot represent a computer that involves interactions [10]—e.g., self-driving vehicles [10], and a desktop computer with a mouse.

Generative AI (GAI) is a type of AI that is designed to generate an output based on input information. In the case of Dall·E, an image is created based on the textual input by the user [11]. A simple GAI system like a Naives Bayes classifier does not require a neural network. However, complex systems like large language models (LLM) or Dall·E do. At the simplest level, a neural network contains three main types of nodes: input, output, and hidden layers. The information inside the hidden layers can be difficult to understand. A neural network can contain many hidden layers. Deep learning involves using multiple hidden layers in a single network. There are several types of language learning models that use deep learning. Earlier ones used recurrent neural networks. However, the more successful one is the Transformers [12]. Transformers have been adapted text-to-image AIs such as Dall·E [11]. LLMs, a type of GAI, can have a chatbot interface. Earlier chatbots like ELIZA [13] are primitive and solely rely on pattern matching. Although these chatbots can pass the Turing test [14], they only have a limited number of responses to the user. Current chatbots can incorporate LLMs and GAI to create a more personalized user experience [15].

In order to make computer-science definitions and arguments for chatbots, we must be able to argue that they can behave like a universal TM (UTM). A UTM is a type of TM that can accept the SD of another TM, and simulate it–akin to a computer running a virtual computer. Being to demonstrate that something is a UTM means, that thing is capable of computation. Chung & Siegelmann [16] demonstrate that a recurrent neural network (RNN) can be trained to behave like a UTM. This means many machine learning (ML) models, including chatbot-based AI, can behave like computers.

IV. Formal Definitions of Chatbot with Generative Artificial Intelligence

In this section, I define a chatbot using GAI, and the training software for the chatbot as TMs. While I use NTM and multi-track NTM as part of definitions, these are no more powerful than a normal TM because they can be converted into normal TMs.

Definition 1. For each chatbot using GAI, there exists a non-deterministic TM named M_{AI} which accepts a user prompt as tape τ_p. τ_p encodes p which is a string in a formal language $\mathcal{L}_p$. M_{AI} converts τ_p into τ_a which encodes a, a string in another formal language $\mathcal{L}_a$. $\mathcal{L}_a$ can represent strings, encoded images, encoded audios, encoded videos and more. If the user has previously used M_{AI}, information on the previous τ_a's can be included into τ_p.

Remark. The format of τ_a can be different based on the type of AI. For instance, a chatbot that answers with a text, τ_a might be: $\boxed{C}\boxed{e}\boxed{r}\boxed{t}\boxed{a}\boxed{i}\boxed{n}\boxed{l}\boxed{y}\boxed{!}\ldots$. Meanwhile, an image-generating chatbot like Dall·E may produce τ_a that contains image metadata followed by a set of 3-vectors describing pixel colours. Furthermore, Definition 2 assumes that the chatbot does not have a memory. However, if the memory is to be used, we can encode a memory as symbols on τ_p.

Definition 2. For each training software for M_{AI}, there exists a multi-track NTM named M_T which accepts the SD of M_{AI} as a tape, and a tape named τ_d which contains encoded training data. M_T smithes (i.e. modifies) the SD using τ_d. Let $\mathcal{L}_{a,p}$ be the set of a's that can be produced for the same p. $\mathcal{L}_{a,p} \subset \mathcal{L}_a$. The goal of M_T is to train M_{AI} in order to maximize the size of $\mathcal{L}_{a,p}$ while maintaining the goodness of $a \in \mathcal{L}_{a,p}$.

Definition 3. For each human-in-the-loop software for M_{AI}, there exists a multi-track NTM named M_R which accepts τ_{AI}, τ_A, and τ_h. τ_a contains the SD of M_{AI}. τ_A represents a set named A. A contains answers (a) which have been generated by M_{AI}. All answers are in $\mathcal{L}_a$. Finally, τ_h contains human-generated information on the best $a \in A$. M_R modifies τ_{AI} based on τ_A and τ_h.

V. Undecidability

An important task for evaluating M_{AI} is that we must be able to decide if it will only produce good output. However, this is undecidabl because since M_{AI} is also a TM.

A. Good Output Is Undecidable

The definition of *good* is subjective. Still, I argue that, in general, being "good" means M_{AI} only produces an answer that is appropriate to the prompt. For instance, if we ask M_{AI} to generate an image, it should not contain any hallucination. Furthermore, if M_{AI} must provide an answer to a question encoded in p, the answer should not only be correct, but useful to the user. For instance, most people know that the name of the capital of Thailand is "Bangkok." However, the real and full name of the capital is actually: "Krungthepmahanakhon Amonrattanakosin Mahintharayutthaya Mahadilokphop Noppharatratchathaniburirom Udomratchaniwetmahasathan Amonphimanawatansathit Sakkathattiyawitsanukamprasit."[1] This answer is the most correct one, but should only be presented in rare circumstances. After all, even the local Thai people almost always only use the first few syllables. Therefore, good M_{AI} will answer that "Bangkok", "Krungthep", "Krugthep-mahanakhon" are equally good, but each should be used in

[1]Spelling obtained from: https://en.wikipedia.org/wiki/Bangkok.

a specific context. Local people will colloquially call the capital "Krungthep." "Krugnthepmahanakhon" is more formal and more official. Meanwhile, "Bangkok" can be considered official and formal if used outside of Thailand.

Theorem 1. `Good Output` problem, or determining if M_{AI} will only produce good a is undecidable.

Proof. This is a proof by contradiction. Assume that it is possible to computationally decide if M_{AI} will only produce good $a \in \mathcal{L}_a$. Then, it is possible to construct M_J which accepts τ_{AI}, a tape which encodes the SD of M_{AI}. M_J then determines if M_{AI} will only print specific sets of symbols which lead to good a. However, Turing demonstrates that the `Printing` problem is undecidable; we cannot construct a TM to predict if another arbitrary TM will ever print a specific symbol [17]. Since M_J must decide if M_{AI} will print certain sets of symbols, this is a contradiction and such machine cannot exist. Therefore, M_J may not exist and the problem may be undecidable.

At this point, we cannot show that the problem is truly undecidable for M_{AI}, because the `Printing` problem is still decidable for some TM. For instance, Windows 11 Co-Pilot can easily recognize if this Python code will print "G": *print("G")*, and this Python code can trivially be converted into a TM. As such, some TMs are decidable even if the problem in general is not. Let $\mathcal{M}$ be the set of all TMs and $\mathcal{M}_{AI}$ be the set of all possible M_{AI}'s. It is conceivable that the `Printing` problem is decidable on all $\mathcal{M}_{AI}$ because $\mathcal{M}_{AI}$ is just a subset of $\mathcal{M}$. This means constructing M_J might still be possible for $\mathcal{M}_{AI}$.

To show that M_J cannot be constructed for $\mathcal{M}_{AI}$, we must show that the undecidability of `Printing` problem also applies to $\mathcal{M}_{AI}$ by proving that all members of $\mathcal{M}$ is reducible to a machine in $\mathcal{M}_{AI}$. Let M' be a TM that has memorized a SD of $M \in \mathcal{M}$ in its neural network. M' also contains a RNN which is trained to act like a UTM. When M' runs, it accepts a tape τ. Then, it simulates M which then processes τ. Since M' acts like a chatbot with a neural network that converts one string to another, $M' \in \mathcal{M}_{AI}$. Thus, the `Printing` problem also undecidable for $\mathcal{M}_{AI}$. Therefore, M_J cannot be constructed and `Good Output` is undecidable.

This proof is based on the work by Brennan [18]. Hers is about demonstrating that determining if TM-reducible AI is ethical is impossible. Unlike mine, her proof relies on the theorem which states that the `Halting` problem is undecidable. However, I argue that Turing's theorem is more applicable here because we want to focus more on output. Additionally, Brennan's proof assumes that AI programs are the same as any computer program (i.e. $\mathcal{M}_{AI} = \mathcal{M}$) without any reduction. □

Remark. It is important to note that computational undecidability is a very strict description. While `Printing` is undecidable for an arbitrary TM, there are still TMs that can be decided on. Furthermore, there are multiple degrees of undecidability; some problems have fewer undecidable instances than others [17]. Still, we must be aware that some of these problems can be very difficult. For instance, while `Halting` is undecidable, determining if a NTM will halt within n steps is decidable. However, it is also NP-complete (i.e. hard) [19].

VI. Hardness of Prompts

τ_p may include an instance of a problem that must be solved using an algorithm. This requires M_{AI} matching the tape against various solvers that M_{AI} knows. For instance, an unscrupulous student in a logic class may ask M_{AI} to check if a Boolean expression is satisfiable. M_{AI}, after parsing the student's question, matches p with a `SAT` solver and applies the solver on p.

Definition 4. Let p be a prompt that contains a problem that requires an algorithm to solve, $\mathcal{S}$ be a set of TMs that represent solver algorithms, $\mathcal{R}$ be a set of recognizers. Each $r \in \mathcal{R}$ is a formal grammar. If p is accepted by r, then p solvable by $s \in \mathcal{S}$ and M_{AI} applies s on p.

Remark. It is important to note that this definition only applies to general-purpose chatbots (e.g., ChatGPT). Some chatbots are specialized and are not designed to be universal problem solvers. Furthermore, it is unlikely that $\mathcal{R}$ and $\mathcal{S}$ actually exist or are implemented within the bots' programming. Rather, they have to be inferred.

Since humans generate prompts, r should work with a natural human language. Because human languages can be modelled using context-sensitive grammars (CSG) [20], then r should be a CSG. This is challenging because deciding whether a CSG will accept p is PSPACE-Complete [21]. Since PSPACE-Complete problems are likely to be difficult to solve, M_{AI} may not have enough computational resources to operate the recognizers. Therefore, M_{AI} may simulate r as context-free grammars (CFG) instead because they are more practical. This will cause errors. Still, since human languages are only *mildly* context-sensitive [20], CFG recognizers should be sufficient for most cases.

So far, we assume that p does not contain any ambiguity. However, ambiguity is a feature in all human languages. For instance, a user may ask a chatbot about getting an item from Amazon which can have multiple meanings: (1) an island in Greek mythology, (2) a forest in South America, (3) a river in South America, and (4) an online marketplace. This can pose significant challenges for M_{AI} to process and understand the prompt.

If the problem is matched by a recognizer, the next question is: will M_{AI} solve the problem if it is difficult? Some problems, like solving a Sudoku game, are obviously hard. A Sudoku game is a mathematical puzzle that has been shown to be NP-Complete, a class of hard problems [22]. Some problems are innocuous at first glance. For instance, a user may ask the chatbot how to fit items into a trunk of their car in a way that maximizes the cargo value while respecting the car's weight limit. This is a variation of the knapsack problem, which turns out to be as difficult as Sudoku [7].

I suspect that the enterprise hosting the chatbot is less interested in ensuring the correctness of the recognizers and the solvers. Therefore, they are not willing to solve PSPACE- and NP-complete problems. Instead, they will deploy "good enough" estimates to conserve the resource for M_{AI}.

VII. Discussion: More Intelligent Chatbot

A. Semiotics

Semiotics is a field of study of signs. A sign can be anything—from words written on paper and sounds being uttered, to abstract concepts existing inside someone's mind [23]. Computer scientists, particularly in the field of visualization and analytics, deploy semiotics to better understand how the user perceives information and understands it. An example of this is a paper by Borgo et al. [24] which outlines how semiotics could apply to glyph-based visualization (i.e. using small visual markers to represent information). Semiosis is an important process in semiotics; it involves converting one sign to another. Visualization and analytics experts study semiosis by trying to understand how altering visual content affects how people understand data.

It is important to note that while all TM symbols are signs, not all signs are symbols. Some signs emerge from a set of symbols or combinations of other signs. For example, a sentence is a group of words which are groups of letters, and the sentence, the individual words, and the individual letters are their signs. Furthermore, signs have syntax; they must be put in order. An example is that if we scramble words in a sentence, the sentence will carry a different meaning or cease to make sense. Image-generating chatbots need to deal with signs and semiosis to generate a good output. Fig. 1 shows that AI must first be able to create signs out of its own symbols in Σ. As Fig. 1 was created by an image-generating AI, the AI's Σ contains symbols specifically for encoding colours and other image metadata. However, we can also observe other signs emerge from a combination of the symbols. For instance, we can observe visual signs for

Fig. 1. Image generated from Windows 11 Copilot on March 15, 2024 with the prompt: "draw a therapist named eliza as an anime girl".

women, medical gowns, stethoscopes, and more. The image also demonstrates that the AI does not have a good mastery of semiosis. First, the prompt was to create a therapist, a non-medical professional who counsels people with mental health issues. However, the AI conflates this sign with the sign of all types of therapists—including ones working within hospitals. This results in medical equipment and uniforms being added. Secondly, the AI does not have a full grasp of pictorial syntax (i.e. a flexible set of flexible for placing elements in an image [25]) and places some items haphazardly. Lastly, the AI also hallucinated some items that do not exist in the real world—notably, the strange headphone full of decorations. It is not enough to output pixels but also to critically think about image fragments that are produced afterwards.

I am cognizant that ultimately, similar to computation, human cognition and semiosis must also have the lowest level that cannot be further decomposed. This argument is also advanced by Shpet [26]. It is important to note though that I am not arguing that contemporary mechanical computer devices are incapable of higher levels of semiosis. Modern-day mechanical computers do not merely compute. They can also perform tasks not captured by the formal definition of computation. For instance, they can accept a stream of data from an environmental sensor and act on it on a real-time basis [10]. Nevertheless, human semiosis has a biological root and its ultimate goal is to maximize the chance of survival within the physical world. Although certain types of AI, including LLMs like ChatGPT, try to emulate aspects of the human brain, a biological and survival-oriented realization of semiosis may be necessary for a superior chatbot.

B. Repetition v. Creativity

In this section, we closely examine Definition 2. Within the Defintion, we note that there are two identifiable goals that are conflicting with each other. The first goal is to maintain goodness which involves M_T reducing the size of $\mathcal{L}_a$ to remove some a's. If M_{AI} is allowed to generate as many a's as possible, many of the a's will be irrelevant—i.e. hallucinations. An example of this is the picture of an impossible rat in a retracted paper by Guo, Dong, & Hao [27]. The second goal is to maximize the size of $\mathcal{L}_{a,p}$ so for each same prompt, different users can obtain different answers. If the answers are too similar to each other, the users may invite public scrutiny. An example of this is a retracted paper by Zhang et al. [28] which begins with: "Certainly, here is a possible introduction for your topic:", a phrase indicative of generative AI being used. The two goals can conflict with each other which results in many a's that contain too much hallucination or repetition. Worse, both can also be the case at the same time. An example is the *Shrimp Jesus* images. In an effort to drive up engagement on Facebook, unscrupulous individuals post many AI-generated images of Jesus. Due to a lack of quality control, many images are very surreal in nature. Some include shrimps arranging themselves to form an image of Jesus, hence the name, *Shrimp Jesus* [29].

I also argue that M_{AI}, as a TM, is not truly creative, because M_{AI} is simply generating content within $\mathcal{L}_a$ and its symbol set cannot be modified. The size of $\mathcal{L}_a$ can be extremely large, but M_{AI} is still bound by the rules and operations of a TM. Let ℓ be the maximum length of τ_a that M_{AI} can produce, then M_{AI} can produce at most $|\Sigma|^\ell$ variations. To alter its semiosis, it must be re-smithed using a M_T or M_R. So far, my argument is similar to one advanced by Lovelace [30]. However, just because a TM like M_{AI} is not truly creative, I do not argue that there will be no creative machines. Modern-day computers and software are already more powerful than TMs. For instance, our operating systems (OS) can download new features from the Internet to update themselves. If one of the features happens to include a superior driver for a webcam, a microphone, or another input device, then we can argue that the OS' semiosis has changed. The OS can "see", "hear" or "sense" better with the features (or worse if the features are buggy). These abilities, while not core to computer science and not supported under CTT, may be what unlocks machine creativity.

C. Interaction

Chatbots as TMs do not fully support interaction. Although it can incorporate data from previous interactions and from the Internet, once the user enters τ_p, the chatbots work in isolation from the outside world. I argue, because of this, M_{AI} is not truly intelligent. To be truly intelligent, chatbots should behave more like choice machines. Choice machines, first described by Turing, are machines that are capable of being manipulated externally. For instance, the machine can become stuck in a certain state and require external interaction (e.g., human intervention) to proceed [10]. The benefit of interaction is that the machine is more embodied within the physical world.

The distinction between a choice machine and a TM is a spectrum. At the lowest level, the difference is almost imperceptible. For instance, a choice machine chatbot can ask the user to provide an additional clarification prompt when it is unable to deal with the current one. In this case, the chatbot is stuck in a state and requires user interaction to proceed. However, for such a simple design, we can also devise a TM that simply answers the user that it cannot process the prompt, and asks the user to provide a clearer prompt. For a more complex choice machine, the AI is constantly receiving information from the environment. An example of this is a traffic camera that constantly receives video stream data from the physical world. In this case, we cannot devise any TM at all, because receiving a stream of data is equivalent to having the input tape being constantly edited while the machine is running [10], [31]. Humans, similar to self-driving vehicles, have sensors that constantly monitor the environment. Therefore, it follows that human cognition cannot be modelled using TMs. Accordingly, for a more intelligent chatbot to be created, it must break free from the restrictive framework of CTT.

Applying formal methods to interaction is even more difficult than analyzing a TM. For instance, while a choice machine is not a TM, parts of its operations can still resemble those of a TM. As such, it is subject to all undecidability issues that affect TM and more. This could also explain why in the field of human-computer interaction (HCI), a field that studies how human users interact with computers, never relies on fully formal evaluations. HCI also relies on informal qualitative methods. This is because computers often encounter undecidable problems, and human semioses are too flexible to be captured by formal notions. Therefore, formal methods will fail at certain points. For instance, we may try to create a "safe" chatbot that can never swear, even if the user is tricking it into swearing. The evaluation of such a chatbot will encounter two issues. First, as a chatbot is a TM, we cannot perfectly predict the output. Secondly, while the chatbot may be unfailingly polite, the user–based on their prior experience and background, may see swear words within the output anyway.

D. Undecidability, Evaluation, and Critical Use of AI

While this paper has argued that analyzing chatbots perfectly is mathematically impossible, it does not argue that no evaluation should be carried out. Although we will never live up to Hilbert's mantra: "We must know. We shall know" [32], it does not mean that we should admit defeat. Muller [32] argues that modern computers only exist because, in Turing's attempt to prove that a problem is undecidable, he also inadvertently provided a blueprint to a basic computer. In fact, if we blindly insist on what is mathematically possible, many fields cannot exist. Software verification research will halt. Human-computer interaction will be toppled as humans are even more opaque than computers. This all leads to a saying: "Perfect is the enemy of the good." Nevertheless, being aware of the fundamental limitations of all chatbots is extremely important, as it promotes critical thinking about AI uses. For instance, if a user knows that a chatbot is not a perfect computer coder, they will be more critical of the chatbot's output. Furthermore, they will be less inclined to incorporate the output into the main codebase without any vetting or verification.

VIII. Conclusion

This paper defines chatbots and their training processes as TMs. While TMs are impractical, they allow us to explore certain aspects of computations. The paper also discusses some challenges with chatbots with respect to CTT. Lastly, I argue that semiotics, choice machines, and interactivity are important for more intelligent chatbots. While evaluating chatbots formally will be an insurmountable challenge, this does not mean that we should halt all work. Other advances can provide scientific and technological rewards to those who persist. Being able to tackle some instances of undecidable problems can still be useful. Recognizing the limitations can, however, still allow us to better understand and work productively with AI. Future work should explore hypercomputation which involves tasks not reducible to TM operations [33]. An example of hypercomputation is expanding the Σ set to include real numbers [33].

Declaration of the Use of Generative Artificial Intelligence

Generative AI was used to generate Fig. 1 for this manuscript. The free version of Grammarly was used for editing.

References

[1] J. R. Lewis, B. S. Utesch, and D. E. Maher, "Umux-lite: when there's no time for the sus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 2099–2102. [Online]. Available: https://doi.org/10.1145/2470654.2481287

[2] D. Q. Goldin, S. A. Smolka, P. C. Attie, and E. L. Sonderegger, "Turing machines, transition systems, and interaction," *Information and Computation*, vol. 194, no. 2, pp. 101–128, 2004, special Issue Commemorating the 50th Birthday Anniversary of Paris C. Kanellakis.

[3] Y. D. Sergeyev and A. Garro, "Single-tape and multi-tape turing machines through the lens of the grossone methodology," *The Journal of Supercomputing*, vol. 65, no. 2, pp. 645–663, Aug 2013.

[4] W. J. Savitch, "Deterministic simulation of non-deterministic turing machines (detailed abstract)," in *Proceedings of the First Annual ACM Symposium on Theory of Computing*, ser. STOC '69. New York, NY, USA: Association for Computing Machinery, 1969, p. 247–248.

[5] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937. [Online]. Available: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s2-42.1.230

[6] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, ser. STOC '71. New York, NY, USA: Association for Computing Machinery, 1971, p. 151–158.

[7] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 1972, pp. 85–103.

[8] O. Shagrir, *The Rise and Fall of Computational Functionalism*, ser. Contemporary Philosophy in Focus. Cambridge University Press, 2005, p. 220–250.

[9] L. M. Pereira, "Turing is among us," *Journal of Logic and Computation*, vol. 22, no. 6, pp. 1257–1277, 2012.

[10] D. Goldin and P. Wegner, "The church-turing thesis: Breaking the myth," in *New Computational Paradigms*, S. B. Cooper, B. Löwe, and L. Torenvliet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 152–168.

[11] J. Betkers, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, W. Manassra, P. Dhariwal, C. Chu, Y. Jiao, and A. Ramesh, "Improving image generation with better captions," 2024. [Online]. Available: https://cdn.openai.com/papers/dall-e-3.pdf

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[13] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, p. 36–45, Jan. 1966. [Online]. Available: https://doi.org/10.1145/365153.365168

[14] M. Mitchell, "The turing test and our shifting conceptions of intelligence," *Science*, vol. 385, no. 6710, p. eadq9356, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.adq9356

[15] A. T. Neumann, Y. Yin, S. Sowe, S. Decker, and M. Jarke, "An llm-driven chatbot in higher education for databases and information systems," *IEEE Transactions on Education*, vol. 68, no. 1, pp. 103–116, 2025.

[16] S. Chung and H. Siegelmann, "Turing completeness of bounded-precision recurrent neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 28 431–28 441. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/ef452c63f81d0105dd4486f775adec81-Paper.pdf

[17] S. Lucas, "The origins of the halting problem," *Journal of Logical and Algebraic Methods in Programming*, vol. 121, p. 100687, 2021.

[18] L. Brennan, "Ai ethical compliance is undecidableai ethical compliance is undecidable," *Hastings Science and Technology Law JournalHastings Science and Technology Law Journa*, vol. 14, no. 2, pp. 311–338, 2023. [Online]. Available: https://repository.uchastings.edu/hastings_science_technology_law_journal/vol14/iss2/5

[19] A. Klingler, M. van der Eyden, S. Stengele, T. Reinhart, and G. de las Cuevas, "Many bounded versions of undecidable problems are NP-hard," *SciPost Phys.*, vol. 14, p. 173, 2023. [Online]. Available: https://scipost.org/10.21468/SciPostPhys.14.6.173

[20] E. P. Stabler, T. C. Collier, G. M. Kobele, Y. Lee, Y. Lin, J. Riggle, Y. Yao, and C. E. Taylor, "The learning and emergence of mildly context sensitive languages," in *Advances in Artificial Life*, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 525–534.

[21] B. Slivnik, "Context-sensitive parsing for programming languages," *Journal of Computer Languages*, vol. 73, p. 101172, 2022.

[22] T. Yato and T. Seta, "Complexity and completeness of finding another solution and its application to puzzles," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, 05 2003.

[23] D. Chandler, "Models of the Sign," in *Semiotics: The Basics*, 3rd ed. New York, NY: Routledge, 2018, ch. 1, pp. 11–67.

[24] R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications," in *Eurographics 2013 - State of the Art Reports*, M. Sbert and L. Szirmay-Kalos, Eds. The Eurographics Association, 2013.

[25] K. J. Lande, "Pictorial syntax," *Mind & Language*, vol. 39, no. 4, pp. 518–539, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mila.12497

[26] V. Feshchenko, "Gustav shpet's deep semiotics: A science of understanding signs," *Sign Systems Studies*, vol. 43, no. 2/3, p. 235–248, Nov. 2015.

[27] X. Guo, L. Dong, and D. Hao, "Retracted: Cellular functions of spermatogonial stem cells in relation to jak/stat signaling pathway," *Frontiers in Cell and Developmental Biology*, vol. 11, Feb. 2024.

[28] M. Zhang, L. Wu, T. Yang, B. Zhu, and Y. Liu, "Retracted: The three-dimensional porous mesh structure of cu-based metal-organic-framework - aramid cellulose separator enhances the electrochemical performance of lithium metal anode batteries," *Surfaces and Interfaces*, vol. 46, p. 104081, 2024.

[29] A. Yang, "Facebook users say 'amen' to bizarre ai-generated images of jesus," *NBC News*, 3 2024. [Online]. Available: https://www.nbcnews.com/tech/tech-news/facebook-users-say-amen-bizarre-ai-generated-images-jesus-rcna143965

[30] S. Bringsjord, P. Bello, and D. Ferrucci, "Creativity, the turing test, and the (better) lovelace test," *Minds and Machines*, vol. 11, no. 1, pp. 3–27, Feb 2001. [Online]. Available: https://doi.org/10.1023/A:1011206622741

[31] D. Goldin and P. Wegner, "The interactive nature of computing: Refuting the strong church–turing thesis," *Minds and Machines*, vol. 18, no. 1, pp. 17–38, Mar 2008.

[32] D. Muller, "Math's fundamental flaw," May 2021. [Online]. Available: https://www.youtube.com/watch?v=HeQX2HjkcNo

[33] B. J. Copeland, "Hypercomputation," *Minds and Machines*, vol. 12, no. 4, pp. 461–502, Nov 2002. [Online]. Available: https://doi.org/10.1023/A:1021105915386