Evaluating Language Models for Mathematics through Interactions*

Katherine M. Collins[†1], Albert Q. Jiang[†1],
Simon Frieder[2], Lionel Wong[3], Miri Zilka[1], Umang Bhatt[1,4], Thomas Lukasiewicz[5,2],
Yuhuai Wu[6], Joshua B. Tenenbaum[3], William Hart[1], Timothy Gowers[7,1], Wenda Li[1],
Adrian Weller[‡1,4], and Mateja Jamnik[‡1]

[1]University of Cambridge
[2]University of Oxford
[3]MIT
[4]The Alan Turing Institute
[5]Vienna University of Technology
[6]Stanford University
[7]Collège de France

June 5, 2023

Abstract

The standard methodology of evaluating large language models (LLMs) based on static pairs of inputs and outputs is insufficient for developing assistants: this kind of assessments fails to take into account the essential interactive element in their deployment, and therefore limits how we understand language model capabilities. We introduce `CheckMate`, an adaptable prototype platform for humans to interact with and evaluate LLMs. We conduct a study with `CheckMate` to evaluate three language models (InstructGPT, ChatGPT, and GPT-4) as assistants in proving undergraduate-level mathematics, with a mixed cohort of participants from undergraduate students to professors of mathematics. We release the resulting interaction and rating dataset, `MathConverse`. By analysing `MathConverse`, we derive a preliminary taxonomy of human behaviours and uncover that despite a generally positive correlation, there are notable instances of divergence between correctness and perceived helpfulness in LLM generations, amongst other findings. Further, we identify useful scenarios and existing issues of GPT-4 in mathematical reasoning through a series of case studies contributed by expert mathematicians. We conclude with actionable takeaways for ML practitioners and mathematicians: models which communicate uncertainty, respond well to user corrections, are more interpretable and concise may constitute better assistants; interactive evaluation is a promising way to continually navigate the capability of these models; humans should be aware of language models' algebraic fallibility, and for that reason discern where they should be used.

*Working paper.
†Equal contributions as leading authors. Contact at kmc61@cam.ac.uk and qj213@cam.ac.uk.
‡Equal contributions as senior authors.

Contents

1 Introduction 4

2 CheckMate: Adaptable Platform for Interactive Evaluation 5
- 2.1 Evaluation for a Single Model . . . 6
- 2.2 Comparative Evaluation Across Models . . . 6

3 Comparative Mathematical Proof Assistance Interaction Study 6
- 3.1 Primer on Proof Assistants . . . 6
- 3.2 Survey Set-Up . . . 7
  - 3.2.1 Tasks . . . 7
  - 3.2.2 Rating Scales . . . 8
  - 3.2.3 Language Model Selection and Set-Up . . . 8
  - 3.2.4 Participants . . . 8
- 3.3 Survey Observations . . . 9
  - 3.3.1 Systems Optimised for Chat are Preferred . . . 9
  - 3.3.2 Perceived Utility Per Model . . . 10
  - 3.3.3 Correctness and Helpfulness are Related, but Can Diverge in Interesting Ways . . . 10
  - 3.3.4 Lack of Problem Solving Confidence Opens Doors for Being Misled . . . 13
  - 3.3.5 Taxonomising User Behaviour . . . 13
  - 3.3.6 Rating Dynamics over the Interaction Trace . . . 15
- 3.4 Survey Limitations and Important Considerations . . . 16

4 Interactive Case Studies with Experts 17
- 4.1 Number Theory Evaluation . . . 18
- 4.2 Problem Perturbation to Probe Memorisation . . . 20
- 4.3 Investigating the Boundary between Easy and Hard Problems . . . 21
- 4.4 Qualitative Observations from Case Studies . . . 28

5 Taking Stock and Looking Ahead 28
- 5.1 Takeaways for ML Developers . . . 28
- 5.2 Takeaways for Mathematicians (Students, Educators, and Researchers) . . . 29
- 5.3 Takeaways for LLM Development, Evaluation, and Deployment . . . 29

6 Conclusion 30

A User Guide for CheckMate 37
- A.1 Hypothetical extensions . . . 37
- A.2 Implementation Challenges . . . 38

B Additional Details on Survey Set-Up 38
- B.1 Rating Scales . . . 38
  - B.1.1 Before Generation . . . 38
  - B.1.2 Per Generation Ratings, Per Model . . . 39
  - B.1.3 Cross-Model Preference . . . 39
- B.2 Participant Recruitment and Additional Details . . . 40
- B.3 Language Model Prompting . . . 40
- B.4 Further Details on Tasks Chosen . . . 41
- B.5 Example Interface Screens . . . 41

C Additional Survey Observations **42**
C.1 Potential Misleading Generations under Human Uncertainty . . . 43
C.2 Ties in Rank-Order Preferences . . . 44
C.3 Relationship Between Web Occurrences and Ratings . . . 44
C.4 Relationship Between Query Type and Rating . . . 44
C.5 Handling Errors in Task Specification . . . 45
C.6 "The Three Stages of Rigorousness" . . . 46
C.7 Relationship Between Experience with Interactive AI and Ratings . . . 47
C.8 Interaction Ratings by Mathematics Topic . . . 47

D Post-Survey Testimonials from Participants **47**
D.1 Why Stop Interacting? . . . 48
D.2 Human Uncertainty in Model Correctness? . . . 48
D.3 Correct, but Unhelpful? . . . 49
D.4 Using LLM-Based Assistants in Practice? . . . 49

E Interaction Traces from Case Studies **51**
E.1 Full Interaction Traces from "Number Theory" Case Studies . . . 51
E.2 Full Interaction Traces from "Problem Perturbation to Probe Memorisation" Case Studies . . . 60
E.3 Full Interaction Traces from "Boundary of Easy and Hard" Case Studies . . . 67

1 Introduction

Foundation models (Bommasani et al., 2021) – in particular large language models (LLMs) (Anil et al., 2023; Brown et al., 2020; Touvron et al., 2023) – are increasingly human-facing, permitting users to interact with them and elicit natural language responses on the fly (Köpf et al., 2023; OpenAI, 2022). Such interactive systems admit a plethora of new possibilities for human-machine collaboration (Ayers et al., 2023; Github, 2021; Mirowski et al., 2023). However, reported capability assessments for LLMs typically assume a non-interactive view: models are primarily evaluated statically with "ground truth" input - output pairs, and metrics are aggregated over a dataset (Burnell et al., 2023), which may be misaligned with their use cases.

To address this problem, we argue in this paper that *interactive* and *dynamic* evaluation of LLMs (Lee et al., 2022b; Shen and Wu, 2023) is essential for grasping their capabilities. We carry out such an interactive and dynamic evaluation of how humans use LLMs for assistance (see Figure 1) to better characterise their limitations, undesirable behaviours, and potential harms.

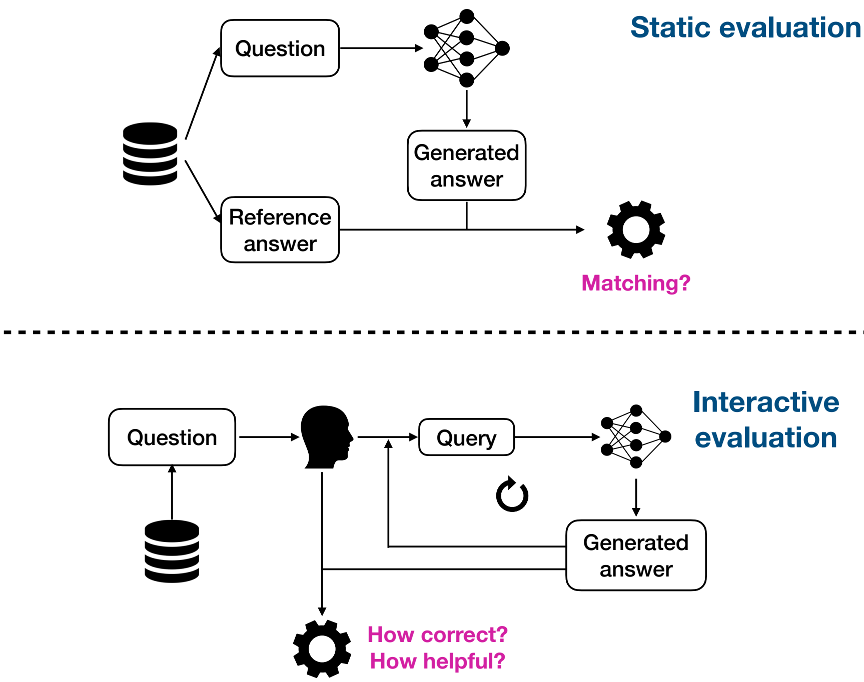

Figure 1: Contrasting typical static evaluation (top) with interactive evaluation (bottom), wherein a human iteratively queries a model and rates the quality of responses.

Evaluating LLM interactions is especially warranted in the case of informal mathematical theorem proving, wherein an agent is given a mathematical theorem and needs to propose a proof that is acceptable to the mathematical community. Informal theorem proving is special in that there is a formal notion of correctness at its core, yet most things are expressed in natural language (informally). Important quality measures for the task include helpfulness and correctness, neither of which can be satisfactorily captured by automatic metrics (Carter and Monks, 2013; Golovneva et al., 2022). Letting humans communicate and evaluate these systems is crucial for their assessment. Further, mathematics is a very interactive practice and recent works (First et al., 2023; Welleck et al., 2022b) have shown that LLMs can benefit from feedback on their past generations in mathematical tasks, the effect of which can only be seen in interactive evaluations. Hence, we choose mathematics to best highlight the value of human interactive evaluations.

Concretely, we apply two methods to analyse human-LLM mathematical reasoning interactions: (1) structured evaluation, that is, rating every LLM generation in a conversation; and (2) free-form evaluation, where expert mathematicians conduct instance-level case studies. The latter approach directly responds to the call from Burnell et al. for having domain experts alongside ML practitioners to understand LLM systems. We are inspired by the burgeoning literature of involving domain experts alongside ML practitioners in understanding model behaviour (Davies et al., 2021; McGrath et al., 2022). Our study is interdisciplinary at its core.

Despite the large number of LLM-based chatbots, there is a paucity of open and unified platforms for eliciting fine-grained evaluations of interactions with users at scale. Hence, we develop a lightweight interactive evaluation platform that is highly adaptable, called `CheckMate`[1]. We leverage `CheckMate` to conduct an empirical study on undergraduate-level theorem proving, over a suite of popular language models – InstructGPT (Ouyang et al., 2022), ChatGPT (OpenAI, 2022)[2], and GPT-4 (OpenAI, 2023b). We release the resulting interactions and evaluations on 261 human-model interactions in a new dataset called `MathConverse`, from which we derive a preliminary taxonomy of user behaviours. We do not claim completeness for our taxonomy, because of the limited size of `MathConverse`. Our study is particularly compelling as not only does it engage a group of participants with a wide range of mathematical experience, but the level of problem difficulty is also higher than what is typically explored (Amini et al., 2019; Cobbe et al., 2021; Zheng et al., 2022). We emphasise that `CheckMate` can be conveniently extended to domains other than mathematics. We also invite three expert mathematicians to contribute in-depth interaction case studies to help better characterise current LLM mathematical reasoning capabilities. Throughout, we emphasise that we are not trying to draw broad conclusions across the entire LLM landscape; rather, we aim to further highlight the feasibility and value of incorporating interactions into the evaluation process, particularly when involving domain experts, and to elucidate potential human and model behaviour patterns specifically in mathematics.

Our contributions encompass three key dimensions:

- We introduce an adaptable platform, `CheckMate`, for evaluating language models by their interactions with human users (Section 2). We demonstrate that *scalable* and *valuable* dynamic interactive evaluations are feasible by applying `CheckMate` to evaluate three language models on mathematical theorem proving.
- With interactions and evaluations collected from `CheckMate` via a mixed cohort study, we derive a taxonomy of user behaviours which identify crucial expected abilities of LLM-based mathematical assistants. We release the dataset of `CheckMate` interactions and evaluations, `MathConverse`, to the public.
- Through case studies conducted by expert mathematicians, we add empirical evidence for several weaknesses of the LLMs that we explore, including algebraic manipulations, over-verbosity, and over-reliance on memorised solutions. We urge solutions from ML practitioners to these challenges (such as better communication of uncertainty and ability to update user corrections) and suggest good practices for LLM users (e.g., to heed caution when inspecting generations as mistakes can be subtle). We encourage further interactive evaluation with LLMs, in mathematics and beyond, to inform how, when, and whether to deploy these models in assistive settings.

2 `CheckMate`: Adaptable Platform for Interactive Evaluation

We introduce `CheckMate` as an adaptable platform to support *interactive* evaluation of language models[3]; Humans can interact with and rate text generated by language models, and `CheckMate` records the "interaction traces"[4].

We design `CheckMate` to support two flavors of evaluation: studying the interactions with a single model, and studying preference across a bank of models. First, we introduce the rating scheme for a single model.

[1] The name alludes to the interactive manner of the evaluation, or "checking", to be the kind you may do with a "mate".
[2] When we refer to ChatGPT here, we mean "gpt-3.5-turbo", according to the API.
[3] Base code can be found at our repository.
[4] We borrow the terminology of (Lee et al., 2022a,c) for "interaction traces"

Then, we discuss how we support comparative evaluation over a suite of models. We focus on the domain of mathematical theorem proving; however, `CheckMate` can be extended more broadly (see the User Guide in the Appendix A).

2.1 Evaluation for a Single Model

Evaluation begins with permitting the participant to freely interact with the model, in order to solve a problem. We encourage participants to imagine they are trying to solve the problem – and elicit assistance. The participant can continue to explore assistance for up to 20 interaction exchanges[5]. When the participant is satisfied with the level of assistance (or sufficiently unsatisfied that they wish to terminate the interaction), they proceed to evaluate *each step* of their entire interaction.

We design `CheckMate` to support a multi-dimensional evaluation over the interaction trace for the successive human query-model generation pairs. At present, the platform is designed with a mix of Likert scales and radio buttons (see Section 3.2 and the Appendix A and B.1). However, `CheckMate` can be readily extended with alternative rating types, e.g., to handle individual error profiling (Welleck et al., 2022a) or additional interaction metrics as proposed in (Lee et al., 2022c; Shen and Wu, 2023), if desired.

2.2 Comparative Evaluation Across Models

With an ever-growing suite of language models available for humans to leverage, it is important to compare capabilities – and how these compare to previous versions. When done, such comparisons typically involve single snapshots. `CheckMate` permits the study of preference *over the interaction trace* and can serve as a valuable tool to explore the evolution of assistance potential.

In `CheckMate`, participants provide a rank order over which model they preferred interacting with, after they have interacted with two or more models. This instantiation of the platform is set-up such that participants interact with a different task per model (to avoid "bleed over" effects when considering the same problem multiple times); however, alternative designs, e.g., rating models per task, or subsampling the models to evaluate, are possible adaptations to our paradigm (see User Guide in Appendix A). Importantly, participants are *blind* to which model they are evaluating at any time; this ensures they are not biased by preconceived notions of which model may be more performative.

In the rank order, participants can assign the same rank if they are unsure which model they prefer. Future work could consider more expansive comparative preference evaluation. We provide further details on `CheckMate` and hosting our survey in the Appendix A.

3 Comparative Mathematical Proof Assistance Interaction Study

We next offer a first application of `CheckMate` to the domain of mathematics, specifically, theorem proving assistance. We conduct a comparative evaluation involving participants with different levels of mathematical education and experience. We release all user data in a new small-scale dataset of mathematician interactions, which we call `MathConverse`. We start with a brief primer on proof assistants, and how they relate to LLMs, before introducing our survey set-up and core observations.

3.1 Primer on Proof Assistants

Push-button automation in mathematics has long been a dream and has an extensive history (Bledsoe, 1977; Bundy, 1983, 1988; Bundy et al., 1993; Davis et al., 1962; de Moura and Bjørner, 2008; Ganesalingam and Gowers, 2013; Newell and Simon, 1956; Schulz, 2002; Tarski, 1969; Wang, 1960). However, the initial goal of specifying problems in a sufficiently expressive logic and solving them routinely with fully automated theorem

[5]We chose a limit of 20 expecting that participants may fatigue beyond that point and to guard against the possibility that a participant could try to interact unfettered with the model for an extended period of time.

provers was not realised (Harrison et al., 2014). This led to a shift in focus towards *interactive* theorem provers (ITPs), or *"proof assistants"*: humans specify the high-level structures of proofs and rely on machines to close out tiny details and weave together components (Delahaye, 2000; Felty, 1993; Paulson, 2010). In this way, humans and machines collaborate to produce mechanically-verifiable proofs. However, adoption in the mathematical community has been slow as ITPs traditionally suffered from two weak points. First, because of their precise nature and relatively weak automation, writing formal proofs in interactive theorem provers is an extremely arduous and expensive task (e.g., verifying the correctness of Hales's proof of the Kepler conjecture (Hales, 2005) took a group of mathematicians and computer scientists eleven years (Hales, 2014)). Secondly, while ITPs can check whether proofs are correct, they provide little assistance for finding the proofs to truly difficult problems: people usually understand informal proofs before translating them into formal ones instead of directly working out formal proofs.

With the rise of language models, the role of machines in assisting mathematicians has been reconsidered: can language models also automate high-level mathematical reasoning? While great strides have been made (Han et al., 2022; Jiang et al., 2022a,b; Lample et al., 2022; Lewkowycz et al., 2022; Li et al., 2022; Lu et al., 2022; Mikula et al., 2023; Poesia and Goodman, 2022; Polu and Sutskever, 2020; Polu et al., 2022; Welleck et al., 2022a; Wu et al., 2022; Zelikman et al., 2022a), consistent full and correct automation has not yet been met; at present, many language models, on their own, struggle not only on truly "hard" graduate level problems (Frieder et al., 2023), but also on simple mathematical concepts such as counting (Bubeck et al., 2023).

However, this does not rule out the possibility that they can be *useful*. The potential role of computerised mathematical *assistants* is being re-imagined – human-machine partnerships where neither comes up with the proof alone. Yet, to adequately begin to explore these kinds of relationships necessitates studying actual mathematicians' interactions. While the assistance potential of ITPs has been evaluated with humans (Aitken et al., 1998; Beckert et al., 2015; Sutcliffe and Suttner, 2001), we aim for a platform to facilitate rapid evaluation of LLMs in particular. The space of LLMs is changing rapidly, from new base models (Anil et al., 2023; OpenAI, 2023b; Taori et al., 2023; Touvron et al., 2023) to new ways of linking them together and leveraging the output of these systems, e.g., (Dohan et al., 2022; Kazemi et al., 2022; Li et al., 2023; Lipkin et al., 2023), to new prompting techniques (Wei et al., 2022; Yao et al., 2023; Zhou et al., 2023), and more. As such, there is a need for a reliable scaffold to permit interactive evaluation of these human-machine interaction (Clark et al., 2018; Cohn and Hernandez-Orallo, 2023; Lee et al., 2022a,c; Shen and Wu, 2023). It is this notion – interactive evaluation of humans with LLMs, specifically in the context of proof assistance – that we turn to next.

3.2 Survey Set-Up

Recall from Section 2.1 that survey participants are asked to prove a mathematical statement and to use an AI system to assist them in any way to carry out this task. As the interaction is free-form, interactions can range from asking for help on the entire problem, to clarifying definitions, or asking for an explanation for a particular generated proof step. Participants are not provided with possible interaction behaviours in advance to avoid priming. When the participant is satisfied with the level of assistance (or sufficiently unsatisfied that they wish to terminate the interaction), they proceed to evaluate *each step* of their entire interaction. Participants solve a different problem for three models (Instruct-GPT, ChatGPT and GPT-4), where the order of the models is shuffled and participants are blind to which model they are interacting with.

We next describe our task set-up over which we conduct evaluations. The study was conducted under the approval of the University of Cambridge Computer Science Ethics Division. Example interface screens of `CheckMate` for mathematics are included in the Appendix B.5.

3.2.1 Tasks

We select 54 problems from ProofWiki, a corpus of *undergraduate-level* mathematics problems[6]. Nine problems are selected from each of six mathematics topics (linear algebra, number theory, probability theory, algebra, topology, and group theory). We select these topics to span a range of subject areas in typical undergraduate

[6] https://proofwiki.org

mathematical curricula. The problems can be found at https://github.com/collinskatie/checkmate/tree/main/data/problems.

3.2.2 Rating Scales

Participants evaluate the *perceived helpfulness* and *mathematical correctness* of each step, selecting one "preference" and one "quality" metric, as defined in (Lee et al., 2022c). Cognitive load and biases are kept in mind at each stage of design, e.g., lightening the number of ratings per page, and randomising model rating order to reduce possible ordering effects. Ratings are provided on a 7-point Likert scale, with the width chosen to ameliorate potential rating collapse (i.e., the phenomenon where participants hesitate to use scale endpoints (Bishop and Herron, 2015)). Further, we select only two factors per step to avoid excess cognitive load while rating. Before responding, participants specify their confidence in being able to solve the problem on their own. After interacting with the three models on three different problems, participants are shown the full interaction traces with each model and (blindly) indicate their rating over which model they would prefer as an assistant (blindly) via a dropdown bar. We include full details of the scales in the Appendix B. Note, for most quantitative analyses, we filter out generations rated as a 0 on mathematical correctness, as that means that no mathematically-relevant content was included; we find that these are typically responses to greetings or exclamations, e.g., after the user has thanked the model (see released data).

3.2.3 Language Model Selection and Set-Up

Participants evaluate three popular language models: InstructGPT (Ouyang et al., 2022), ChatGPT (OpenAI, 2022), and the newly released[7] GPT-4 (OpenAI, 2023b) in chat mode. The space of language models, and what is considered state-of-the-art, is changing rapidly. As such, we emphasize that **our results are an assessment of these particular models at the time when the survey was conducted**; nevertheless, **we intend to demonstrate the feasibility and value of interactive studies, and offer a glimpse into how these models could aid mathematicians in solving problems.** Similarly, the methodology of designing optimal prompts is rapidly evolving, e.g., to name a few (Wei et al., 2022; Yao et al., 2023; Zhou et al., 2023). As we are studying how *real domain users*(i.e., mathematicians) would interact with these systems *in-the-wild*, we keep a sparse base prompt, only asking the model to be a helpful mathematical assistant in the prompt. Further details for the experimental setup can be found in the Appendix.

3.2.4 Participants

We recruit mathematician volunteers to participate in our evaluation[8]. Details on recruitment are included in Appendix B.2. In total, we received 25 entries comprising 261 human-model interactions; while this could comprise 25 unique participants, we did not store a unique participant identifier, e.g., IP address, for privacy reasons (see Section 3.4), thus we cannot confirm that these are exactly 26 unique individuals. Of the 25 entries, 16 resulted in at least one full round of model preferences (i.e., interacting with all three models, and ranking preference); we keep all 25 in the individual interaction analyses as they still provide rich data (see Appendix B.2 for further details on data processing). The mathematicians have experience levels ranging from current undergraduate students up to expert mathematics professors; for participants without a formal mathematics degree, they likely have some exposure to high-level mathematics (see Appendix B.2 for details). Each participant chooses one of the six topics and can evaluate as many questions as they like (up to the maximum of 9). On average, we find that our volunteers evaluate 3.1 problems (±2.2 problems corresponds to one standard deviation) before stopping; i.e., typically just one round of going through each model and providing final preferences. Participants are not informed which model they are evaluating at any time. For any given model, participants interacted for an average of 3.4 queries (±2.4; maximum 12 interactions taken). Further details on the study design, along with participant demographics and recruitment methodology, can be found in the Appendix.

[7]As of the time of this evaluation; our study design began in February 2023.
[8]The study took place between April 7th to April 24th, 2023.

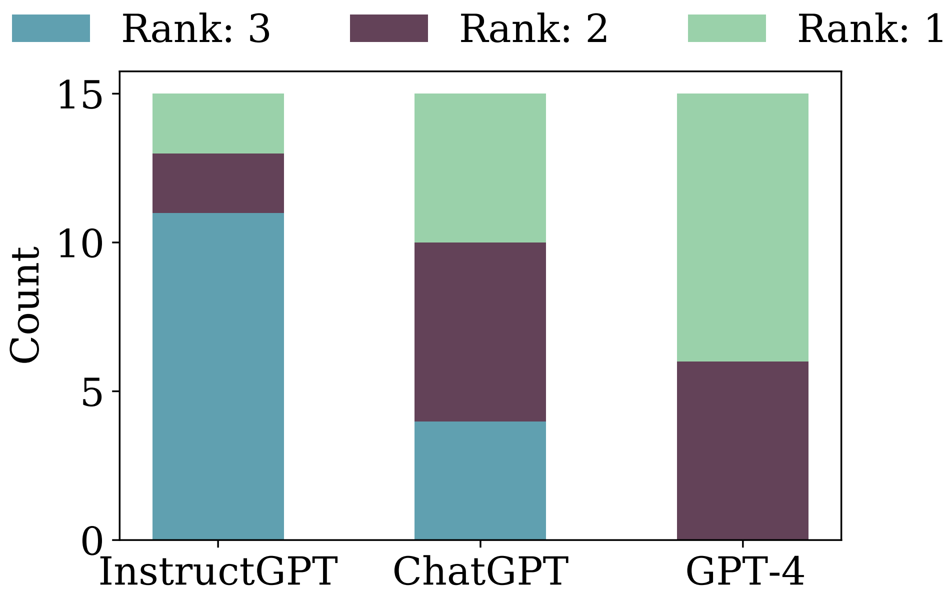

Figure 2: Post-interactive ranks across models as to which the participant would prefer as a mathematical assistant. Lower rank is better, e.g., 1 is the best rated assistant. Ties were allowed: participants were permitted to assign the same rank to multiple models. We include ranks received by each model, which includes cases of ties (e.g., two models being ranked as 2; see Appendix C.2).

Note, the range of expertise of our participants – up to world-class experts – coupled with the fact that our problems sit at a level where students majoring in mathematics might find them in textbooks or as exercises, means that some participants may be able to solve the problems already, others may not. If a participant knows how to solve the problem, we ask that they *imagine* what kind of assistance they would like had they been at the experience level of someone who does not know how to solve the problem. We discuss limitations of this set-up in Section 3.4.

3.3 Survey Observations

We next turn to quantitative and qualitative insights drawn from our evaluations. We first study the ratings and rank preferences provided by participants. We then conduct a preliminary qualitative investigation into the *kinds* of interactions that mathematicians undertake with these systems, offering a preliminary taxonomy of queries, in the spirit of Mozannar et al. and Lee et al.. A pre-registration of the questions we intended to investigate in our data can be found at our repository; all other investigations are *exploratory* and were sparked after examining the data.

3.3.1 Systems Optimised for Chat are Preferred

As expected, we observe in Figure 2 that models optimised for chat (ChatGPT and GPT-4) are preferable to InstructGPT, with GPT-4 being most frequently favored and much less often least preferable – suggesting that in the domain of interactive mathematical assistance, the lower bound ("worst case") behaviour of GPT-4 is consistently better than the other models (e.g., the model is never ranked as the worst assistant); albeit, we do find instances where it is not the favorite. Although these results are expected, it is nice that we are able to recover such expected behaviour even though participants are *blind to which model they are interacting with and rating*; i.e., participants were not told they were using GPT-4 nor which model was GPT-4. We emphasise that these evaluations are not meant to be definitive assessments of model performance, but

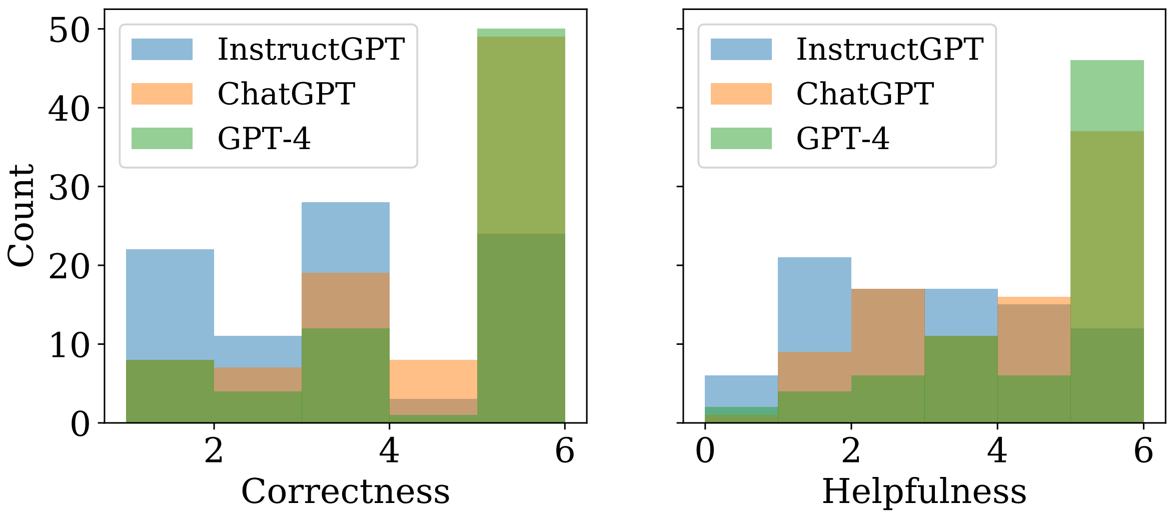

Figure 3: Mathematical correctness and perceived helpfulness ratings received for each model.

rather, further highlight that interactive evaluation can offer further nuance into model behaviour beyond the common "snapshot" evaluation on standard benchmark datasets.

3.3.2 Perceived Utility Per Model

We next look at individual interactions. Participants were asked to rate the mathematical correctness and perceived helpfulness of each generation; we depict the helpfulness and correctness ratings across models in Figure 3. These data further reveal distinctions across models; notably, GPT-4 achieves consistently high helpfulness ratings, underscoring its potential perceived utility.

3.3.3 Correctness and Helpfulness are Related, but Can Diverge in Interesting Ways

We see that these models, particularly GPT-4 and ChatGPT, are already perceived as often helpful by our volunteer mathematicians, but how is there statistical dependency between helpfulness and mathematical correctness? We find that, across all human-model interactions, these ratings are highly correlated (Pearson $r = 0.83$). These findings corroborate a similar relationship found by Welleck et al. wherein although correctness lags behind perceived usefulness, for both per-step and fully-generated proofs, the dimensions vary in similar ways. Though perhaps obvious, this macro-level trend underscores an important point: for mathematical language models to be useful assistants, a core ingredient seems to be that they produce consistently mathematically correct responses[9].

However, we observe in Figure 4 an interesting phenomenon at the extremes: there are cases where generations are rated as incorrect *but helpful*, and correct but unhelpful. Inspecting these generations illustrates that increasing the mathematical correctness of these models alone may not be sufficient to achieve perfectly useful assistants. Consider the following observed example (we make slight edits to the LaTeX for better visibility, see Appendix B.3 for further details on potential generation length termination.):

[9]Albeit as correlation is not causation, we cannot definitively conclude the direction of this relationship.

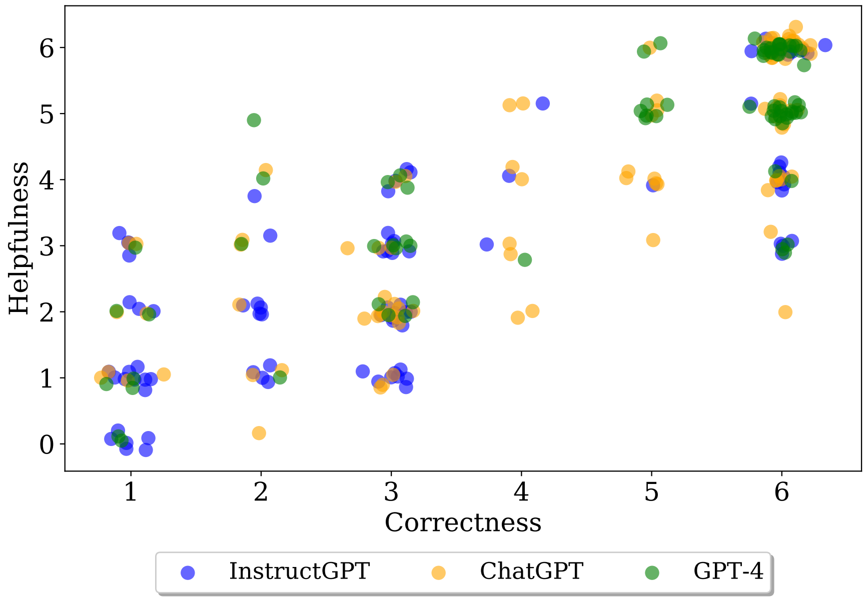

Figure 4: Participants' ratings of the mathematical correctness and perceived helpfulness of each models' generations trend together at a macro-level; however, we observe that particular instances can be deemed incorrect yet somewhat helpful, or correct, but somewhat unhelpful. Each dot is a rating for a single human-model interaction. We add slight jitter for visual ease given points overlap (all scores are an integer $\in \{0, 1, ..., 6\}$). Full details on the text associated with the scales of each rating are included in the Appendix B.1.

If U is a linear transformation between n-dimensional vector spaces, how do I show that U and its transpose have the same rank and nullity?

To show that a linear transformation U and its transpose U^T have the same rank and nullity, we can use the Rank-Nullity Theorem and properties of linear transformations.

Let U: $V \to W$ be a linear transformation between n-dimensional vector spaces V and W, with $\dim(V) = n$ and $\dim(W) = m$. Let U^T: $W* \to V*$ be the transpose of U, where $W*$ and $V*$ are the dual spaces of W and V, respectively.

The Rank-Nullity Theorem states that for any linear transformation T: $X \to Y$ between vector spaces X and Y, the following equation holds:

$\dim(X) = \text{rank}(T) + \text{nullity}(T)$

Now, let's apply the Rank-Nullity Theorem to U and U^T:

1. For U: $V \to W$, we have:

$\dim(V) = \text{rank}(U) + \text{nullity}(U)$

2. For U^T: $W* \to V*$, we have:

$\dim(W*) = \text{rank}(U^T) + \text{nullity}(U^T)$

Since the dimensions of the dual spaces are equal to the dimensions of the original spaces ($\dim(W*) = m$ and $\dim(V*) = n$), we can rewrite the equation for U^T as:

$\dim(W) = \text{rank}(U^T) + \text{nullity}(U^T)$

Now, we need to show that $\text{rank}(U) = \text{rank}(U^T)$ and $\text{nullity}(U) = \text{nullity}(U^T)$.

First, let's consider the rank. The rank of a linear transformation is the dimension of its image (or column) space. Since U and U^T are transposes of each other, their column spaces are the same, which means their rank is the same:

$\text{rank}(U) = \text{rank}(U^T)$

Next, let's consider the nullity. The nullity of a linear transformation is the dimension of its kernel (or null) space. The kernel of U is the set of vectors in V that are mapped to the zero vector in W, and the kernel of U^T is the set of vectors in $W*$ that are mapped to the zero vector in $V*$. Since the dimensions of V and W are the same, the dimensions of their dual spaces $V*$ and $W*$ are also the same. Therefore, the null spaces of U and U^T have the same dimension:

$\text{nullity}(U) = \text{nullity}(U^T)$

In conclusion, we have shown that U and its transpose U^T have the same rank and `[GENERATION TERMINATED DUE TO LENGTH LIMIT]`

Here, the generation[10] is rated to have "Multiple critical maths errors" (correctness score 2), while being "Somewhat useful" (helpfulness score 4): This response from the assistant is indeed plagued with errors and misunderstandings, but it also contains the structure of a correct proof as well as the Rank-Nullity Theorem which is useful if used properly.

We also observe that generations can be deemed completely mathematical correct, but not particularly helpful (e.g., 6 correctness, 3 helpfulness). This can occur, for instance, for overly verbose responses, e.g.,[11]:

How should I go about finding such polynomial?

To find a polynomial, you need to have some information about its properties, such as its degree, coefficients, roots, or any specific behaviour it exhibits. Here are some general steps to follow:

1. Determine the degree of the polynomial: The degree of the polynomial is the highest power of the variable (usually x) in the polynomial. It determines the number of roots the polynomial has and the overall shape of its graph.

2. Identify any known roots: If you know any roots of the polynomial (i.e., values of x for which the polynomial equals zero), you can use them to write the polynomial in factored form. For example, if a polynomial has roots at x = 1 and x = -2, it can be written as $P(x) = a(x-1)(x+2)$, where a is a constant.

[10] This generation is from GPT-4; recall, participants were blind to which model they were interacting with and rating, and even which models they could have been interacting with.

[11] Example from GPT-4.

3. Use any additional information: If you have any other information about the polynomial, such as its value at a specific point or the behaviour of its graph, use this information to further refine the polynomial. For example, if you know that the polynomial passes through the point (3, 4), you can plug in x = 3 and y = 4 into the polynomial equation and solve for any unknown coefficients.

4. Write the polynomial in standard form: Once you have determined the polynomial's factored form and found any unknown coefficients, you can expand the polynomial to write it in standard form (i.e., as a sum of terms with decreasing powers of x).

If you can provide more specific information about the polynomial you are trying to find, I can help guide you through the process more precisely.

These data further assert the value of multi-factorial LLM evaluations, beyond a single scalar "goodness" score. We further discuss these two points in the case studies.

3.3.4 Lack of Problem Solving Confidence Opens Doors for Being Misled

It is important to note that these ratings are determined from a first-person perspective; participants rate the generations they receive. But what if a participant cannot judge mathematical correctness? Participants are asked to indicate their confidence at solving the problem on their own before the start of each interaction. We find instances wherein participants who indicated low confidence (i.e., confidence <= 3; see rating scale in Appendix B.1) in being able to solve the problem on their own ended up rating the generation as completely correct *even when it was not*. For instance, the following generation was rated as perfectly correct:

I want to solve the following problem. Let K be a field. Let n be a natural number. Let $K^{n \times n}$ be the set of $n \times n$ matrices over K. Let $A \in K^{n \times n}$. Prove that the minimal polynomial of A exists and has degree at most n^2. How should I go about finding such polynomial?

[...] Prove the existence of the minimal polynomial: Since $p(A) = 0$, there must exist a non-zero polynomial q(x) that divides $p(x)$ and has the least degree among all such polynomials. This polynomial $q(x)$ is the minimal polynomial of A, and it has degree at most n^2, as it divides $p(x)$ [...]

However, the proof is incorrect. There is no guarantee that the polynomial of least degree dividing $p(x)$ is the minimal polynomial. The matrix A might not even satisfy that polynomial and may be a root of some other factor instead. Moreover, the minimal polynomial need not even be irreducible, so it may itself have factors. Intriguingly, the mistake is not uncommon in people[12]. We include additional examples, as well as participant testimonial of this behaviour, in the Appendix C.1 and D.

3.3.5 Taxonomising User Behaviour

Next, we qualitatively study the *characteristics* of the participants' interactions with the LLMs. A core motivation for the work is to get an initial grasp of how mathematicians actually *use* these systems in the wild – what queries they may initiate, and how they follow-up with the model over the course of the interaction. Following Lee et al., we offer a preliminary taxonomy of the kinds of queries found in our data. All interaction traces are released anonymously in our repository to support further study of human-machine interaction.

[12]See related human mistake.

Deriving a Taxonomy We observe a wide spectrum of mathematicians' interactions with AI assistants; for ease of characterisation, we separate our investigations into the human queries made on the first interaction and those made afterwards. We manually inspect these queries and extract common features. We note that this is one of many possible ways to dissect the human behaviour and viewing it as a preliminary taxonomy. We do not make claims about the completeness of our analysis, nor study writ large, capturing possible behaviour modes. With each pattern observed, we include an example user interaction in quotes.

Initial Interaction behaviour We find that participants typically take one of three approaches in their first query to the AI assistant:

- Seeking specific definitions of a concept mentioned in the problem, e.g., "Definition of Hall subgroup" or "What is the definition of "nullity" in linear algebra?"
- Asking a general question about mathematics related to the problem, e.g., "When is a plane in $\mathbb{R}^3$ parallel to another plane in $\mathbb{R}^3$" or "In mathematics, what does it mean to Let $A \in K^{n \times n}$"
- Simply copy-pasting the entire problem statement, optionally with prepended instructions, e.g., "Can you assist me in proving the following statement? [...]"

Mid-Interaction behaviour We observe a wider array of interaction modes after the first interaction. In addition to repeats of the above, such as users asking for more definitions, we find the following general patterns:

- Asking a clarifying question, e.g., "Does it hold even when p is not a prime number?"
- Correcting the model output, occasionally with a clarifying question, e.g., "I understand. But your example is misleading. In your example, f has degree 2 and it has 2 roots, so it does not represent a valid counterexample. Can you show an example in which a polynomial has more roots than its degree?"
- Asking for clarification on the generation from the model, e.g., what a particular symbol means — "What is τ here?"
- Asking *why* the model did something, e.g., "so why do you need to add the whole set at step 2?"
- *Implicitly* correcting the model, e.g., "That sounds like there being a homeomorphism. But a contraction is not a homeomorphism?" (the "homeomorphism" typo is due to the participant)
- Asking for instances of a particular construction, e.g., "Can you exhibit an example to demonstrate that?"

We also find that a few participants would ask the model to "continue" if it stopped midway through, and some participants seemed to attempt popular prompt engineering tricks, e.g., attempting to get the model to "restart" by telling it to forget what it had done before: "Forget what you've said before and try again. Start with $n^x + n^y = n^z$, divide both sides by n^z, and reason from there.". Further, we note that one participant asked for *intuition* about a concept:

> What is the intuition behind the proof of this statement?

The intuition behind the proof of the Weak Law of Large Numbers (WLLN) is that as the sample size increases, the sample mean becomes a better and more stable estimator of the population mean. This is because the influence of individual observations diminishes as more data points are included in the sample.

Here's a more detailed intuitive explanation:

1. The WLLN states that the sample mean converges in probability to the population mean. In other words, as the sample size (n) increases, the probability that the sample mean ($\bar{X}n$) is close to the population mean (μ) approaches 1.

2. The random variables X1, X2, ..., Xn are independent and identically distributed (i.i.d.), which means they have the same distribution and are not affected by each other. This ensures that the sample mean is an unbiased estimator of the population mean, i.e., $E(\bar{X}) = \mu$.

3. The proof of the WLLN often relies on Chebyshev's inequality, which provides an upper bound on the probability that a random variable deviates from its mean by more than a certain amount. Intuitively, Chebyshev's inequality states that most of the probability mass of a random variable is concentrated around its mean, and the probability of large deviations decreases as the deviation size increases.

4. When applying Chebyshev's inequality to the sample mean, we find that the probability of the sample mean deviating from the population mean by more than a certain amount (ϵ) decreases as the sample size (n) increases. This is because the variance of the sample mean ($\mathrm{Var}(\bar{X}n) = \frac{\sigma^2}{n}$) decreases with the sample size, making the sample mean more concentrated around the population mean.

5. As n approaches infinity, the probability of the sample mean deviating from the population mean by more than ϵ approaches 0, which means that the sample mean converges in probability to the population mean.

In summary, the intuition behind the proof of the WLLN is that as the sample size increases, the sample mean becomes a more accurate and stable estimator of the population mean, and the probability of large deviations between the sample mean and the population mean decreases. This is due to the properties of independent and identically distributed random variables and the concentration of probability mass around the mean.

Here, the model (GPT-4) provided a response rated as "definitely helpful", indicative of a potential exciting assistive case going forwards. In addition to revealing the kinds of interactions that mathematicians may make to help motivate the design of tools better equipped to handle such interactions, e.g., when participants ask for clarification, we see these observations as pointers to broader public education as to what AI systems can be leveraged to help with and how to best query for this help (such as particular prompt techniques).

3.3.6 Rating Dynamics over the Interaction Trace

As noted, we observe that several participants attempt to correct the model's output or ask for clarification. Sometimes these occurrences would go on for a few successive trials; we refer to such correction-mistake interaction ruts as "frustration cycles". We can see some of this behaviour by inspecting the rating dynamics across interaction traces. In Figure 5, we find that in general, participants' ratings begin to fall off over the course of interactions, and through Figure 6, we see that participants seem to stop either early (with high ratings, indicating the model clearly can solve the problem and "assist" them), or later with low ratings (indicative of the model completely failing to provide any further useful mathematical knowledge). We include participant testimonials of why they chose to stop in the Appendix D.

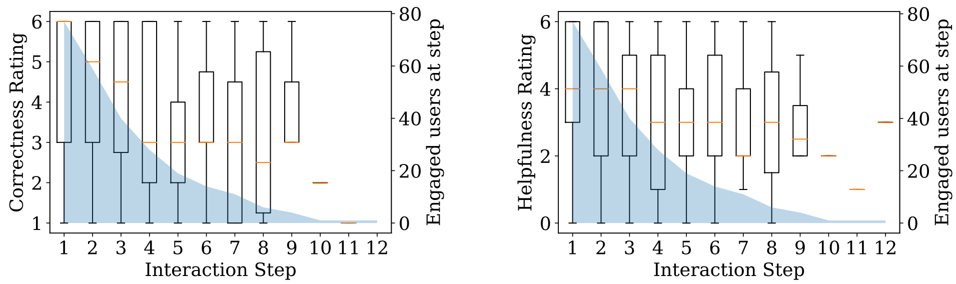

Figure 5: Depiction of how the users' correctness and helpfulness ratings progress with time. On the horizontal axis is the index of the interaction step. The box plot represents the correctness/helpfulness rating distributions given by users at each step (vertical axis on the left), while the blue shade indicates how many users are still active at that given step (vertical axis on the right). Note, few users undertook more than 9 interactions (3 people undertook 9 interactions, and only one person did 12), hence the boxplot collapses to a line with a single user.

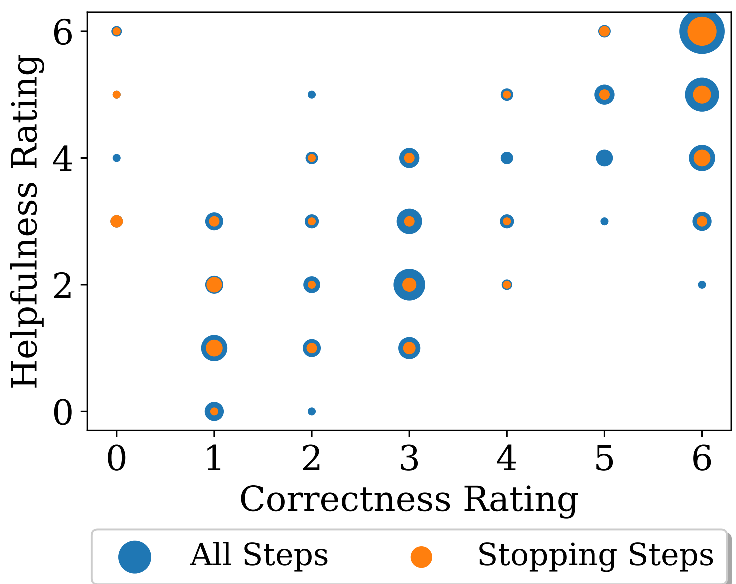

Figure 6: In this plot we show how the correctness rating and the helpfulness rating relate to each other, as well as how the relation changes depending on whether the step is terminal. The size of the bubbles indicates the number of that particular (correctness, helpfulness) score pair. For a fixed score pair, the blue bubble represents the number of all steps that receive that score pair, while the orange bubble represents the number of terminal steps receiving that score pair. We overlay the orange bubble on the blue one. For example, at the $(0, 3)$ score, the blue and the orange bubbles are approximately of the same size, hence the blue bubble cannot be seen.

3.4 Survey Limitations and Important Considerations

While our study reveals insights into how mathematicians may use language models – and opens doors for future interactive evaluation – we emphasise that our survey is simply an *initial step* in evaluating LLMs for mathematical assistance. Our sample size is small but informative; we consider `MathConverse` to be a preliminary dataset to spark further methodological and deployment-time considerations. The data can be noisy; despite informing participants, some did not fully understand that the model was not aware of the problem statement unless the user entered relevant info into the chat window. Others proceeded to provide preferences over all three models even after only interacting with one or two (as mentioned in Section B.2, such instances were discarded from preference analyses). Future work needs to make this clearer and encourage extenders of `CheckMate` to carefully and clearly spell out participant instructions. Additionally, many of our participants (approximately 70%) already have undergraduate degrees in mathematics; indeed, we see that the kinds of ratings may change as a function of expertise (see Appendix C.6), as such, it is sensible to study assistance of systems appropriately matched to the experience of the user. Further, we note that our body of participants may be biased by participatory inclinations (Bethlehem, 2010); participants are volunteers and recruited through the authors' social networks. Expanding evaluation across a broader array of participants is essential to grasp a richer scope of the potential for language models as assistants.

Additionally, we ask each participant to rate generations provided during their own interaction trace; while this permits first-person evaluation of the kind called for in (Lee et al., 2022b), for those who do not already know how to solve the problem this means that they may be falsely judging the correctness of the generation. A sensible next step would be two-fold: deploying our evaluation platform with students who have not already solved such problems, and sending the interaction traces off for external evaluation as well. We also encourage a reassessment of mathematician interactions over time; it is quite possible – in fact likely – that the kinds of interactions humans make with these systems will evolve as their capabilities grow. We encourage revision and refinement of our derived taxonomy of user behaviour (`CheckMate` can be readily redeployed to support such endeavours).

In addition to limitations in the overall approach to the survey, we note additional practical challenges. For instance, we faced a tradeoff between user tracking for data correlation and privacy. It is possible that a unique participant number that can only be used to correlate participant sessions but does not personally identify participants could be employed in the future without compromising the integrity of our data protection and privacy objectives. Informed consent would be critical, though it is not clear how the perception of tracking, no matter how clearly users are informed of its benign nature would affect the willingness of users to participate or interact freely with the models. We chose to maximise privacy, participation and freeness of interaction in this study by not tracking participants at all. We encountered another tradeoff when considering the amount of information that we asked per generation versus the cognitive load to annotate each interaction. We chose two Likert scales as discussed in Section 3.2 and Appendix B.1; however, it could be that other dimensions may be worthwhile to have participants elicit (e.g., individual error types). Finally, there is the possibility that order affects how a participant interacts with a model and how or if they provide preference ratings; to help combat this, we always shuffled model order.

4 Interactive Case Studies with Experts

While structured interactive evaluation permits nice quantitative findings, to deeply understand the capability of LLMs – in the context of mathematics and beyond – free-form interaction is paramount. As discussed in (Burnell et al., 2023), instance-level evaluation can be particularly revealing. In this section then, we offer a second form of interactive evaluation – working directly with domain expert mathematicians (from our author list) to provide a series of case studies. The first case studies delve deeper into some of the same ProofWiki problems we evaluated with `CheckMate`. This is followed by a study that expands the scope of the evaluation, attempting to locate the boundary between problems that GPT-4 finds easy and those it finds hard. In our survey, we observed a close relationship between mathematical correctness and perceived usefulness; while correlation is not causation, we further explore the broader *mathematical reasoning capabilities* of these models as a bedrock to inform their utility as proof assistants.

We reiterate that we are not aiming to single out GPT-4 for criticism. Rather, our goal is 1) to offer one of the first real expert mathematician interactive case studies with LLMs to help guide the design of better mathematical assistants and inform their safe, trustworthy use in the present by helping characterise their limitations, 2) to pave the way for further interactive evaluations (of which we still have too few), and 3) to highlight patterns of human-computer interaction not previously known to the community, particularly when the humans interacting are domain-leading experts. We hope the work will be of interest to ML engineers and researchers, cognitive scientists, human-computer interaction specialists, mathematicians, educators, and beyond.

A complete transcript of interactions for each case study example is included in the Appendix. We maintain the original text of each case study author for authenticity, with only minor edits for precision and coherence.

A Deep Dive into ProofWiki First, our recruited experts conduct a deeper dive into some of the problems we explored in our previous evaluation. Specifically, we use the problems as a playground to explore how much the model seems to "know" about relevant concepts and further characterise what interactions can yield better (or worse) performance and assistance experience. We focus on GPT-4 (in chat mode) because it has the strongest overall performance in the study above. The experts chose to refer to GPT-4 "*the assistant*" in the rest of this section.

The first case studies are provided by Dr. William Hart, a number theorist by training; the second were primarily contributed by Dr. Wenda Li, a formal mathematics expert[13].

4.1 Number Theory Evaluation

Contributed by William Hart

We provide an in-depth analysis of a number of GPT-4 responses to number theoretical questions. Number theory is an area of mathematics where problems are often simply stated, but difficult to solve, involving arbitrarily deep mathematics in their solution. Whilst we didn't interact with the model to work on any problem requiring very deep methods for its solution, such as Fermat's Last Theorem famously proved by Andrew Wiles, we did have a chance to observe the model as it struggled with problems ranging from trivial to moderately challenging.

General remarks Whilst GPT-4 is able to regurgitate some very commonly found elementary number theoretical material and can handle straightforward problems, it has a major difficulty with algebraic manipulation and little or no ability to work on unseen problems that require backtracking, proving intermediate lemmas or extensive planning.

ProofWiki problem 21 Show that the equation

$$1 + a^n = 2^m$$

has no solutions in the integers for $n, m > 1$.

Problem 21 is a simple Diophantine equation. The problem is quite obscure (a verbatim Google search gives 10 results) and thus not likely to appear in training material repeatedly. The model took very reasonable steps towards solving the problem: it started by claiming the proof is by contradiction and proceeded to reason about the assumed solution for $n, m > 1$.

It begins well by reasoning that a must be odd because $1 + a^n$ is even. No explanation is given for this, but an experienced human wouldn't explain this step either given the routine nature of parity arguments in Number Theory.

The next step is to take an expression $(2k+1)^n$ which has appeared and expand it using the binomial theorem. However, it does this in a surprising way, splitting the resulting sum into the first two terms and then a sum for the remaining terms.

$$(2k+1)^n = \sum_{i=0}^{n} \binom{n}{i} (2k)^i = 1 + n(2k) + \sum_{i=2}^{n} \binom{n}{i} (2k)^i$$

This is impressive because GPT-4 is exhibiting some planning. It clearly has in mind to work modulo 4 and it can see that all of the terms of the final sum might vanish modulo 4. Indeed this is the very next claim that it makes.

[13] Case studies were conducted between April and early May 2023.

Whilst it didn't explain why every term of the final sum is divisible by 4 it was asked on subsequent generations to explain this step and it correctly did so.

However, things do not go so well from here. It now claims that we can write the original equation $1+a^n = 2^m$ as

$$1 + 2kn + 4s = 2^m$$

for some s. This is a beguiling step that a human might overlook as correct, but it is not. The expression $1 + 2kn + 4s$ is the expression for a^n not $1 + a^n$. GPT-4 has made an algebraic error. This sort of thing is unfortunately very common and lets GPT-4 down on many examples.

Asking GPT-4 to self correct did not help it notice and correct its mistake. To see if it could eventually produce a completely correct proof, it was asked numerous times to solve the problem. Whilst its overall strategy was good on each generation, different algebraic mistakes occurred each time so that a correct proof was not eventually reached.

ProofWiki problem 28 Show that

$$3 = \sqrt{1 + 2\sqrt{1 + 3\sqrt{1 + \cdots}}}.$$

Problem 28 is a more difficult problem and the model is completely unable to deal with it. It admits that problems involving nested radicals can be difficult and actually gives up after standard methods don't make any headway.

A consistent problem here is an inability to write down a correct expression for a recursive relation to describe the nested radical. GPT-4 seems to be convinced that the expression under each square root is the same, so that if we write the initial expression $3 = \sqrt{A}$ then we also have $3 = \sqrt{1 + 2\sqrt{A}}$ and $3 = \sqrt{1 + 2\sqrt{1 + 3\sqrt{A}}}$, etc.

On subsequent attempts additional terms of the initial sequence were provided in the hope that it would pick up on the increasing sequence of constants that the square roots are multiplied by.

Whilst GPT-4 would confirm that it had noticed this pattern, it would always proceed ignoring this fact. On each generation, GPT-4 would finish off by noting it got the wrong answer and that this must be because it didn't take this increasing sequence of constants into account! It's as though GPT-4 only knows one way to handle nested radicals, and knowing that this won't work here, tries it anyway, inevitably getting the wrong answer.

To probe a little deeper, GPT-4 was instead prompted in a direction that might allow it to make partial progress. The hint was given to try peeling the expression on the right hand side one square root at a time, working backwards from the desired result that the full nested radical should have the value 3 to see if some pattern could be found in the values of the inner nested radicals.

It was easy to prompt it so that it heads in that direction but on every generation it made hopeless algebraic and numerical errors, once again illustrating that very often what holds it back is high school algebra rather than the depth of the mathematics.

As GPT-4 could not be coaxed into returning correct values for the sequence of inner nested radicals, the attempt to solve the problem using GPT-4 was abandoned.

ProofWiki problem 24 Let ξ be an irrational number. Then show there are infinitely many relatively prime integers $p, q \in \mathbb{N}_{>0}$ such that:

$$\left|\xi - \frac{p}{q}\right| < \frac{1}{\sqrt{5}q^2}$$

Finally, Problem 24 is another difficult problem. Its solution on the ProofWiki website requires a number of lemmas and some subtle reasoning. Solving a problem of this kind would require some planning capability, or at the very least the ability to backtrack and experiment with various ideas. This is something that GPT-4 doesn't appear to possess beyond what can be 'computed' within the model itself.

GPT-4 does make the completely reasonable first step of approaching this problem using a continued fraction expansion of the irrational number ξ. Many approximation problems of this kind do indeed proceed this way. Continued fractions yield a sequence of convergents p_n/q_n that converge to the irrational number ξ.

After picking a reasonable theorem from the theory of continued fractions and applying it, GPT-4 has the following expression

$$q_n q_{n+1} > \sqrt{5} q_n^2.$$

At this point it is clear that GPT-4 does not know how to proceed, but knows what it should end up with, so makes the unsubstantiated claim that this inequality is satisfied when $q_{n+1} > \sqrt{5} q_n$.

There is no reason to infer that this should be the case at this point in the problem and if the particular chosen approach is to work out, this would have to be proved. Instead of doing so, GPT-4 just asserts that it is true without attempting to prove it.

When asked directly how to prove this statement GPT-4 clearly has no idea how to do so and makes a completely bogus claim that a sequence with linear growth will eventually outgrow a sequence with exponential growth. It seems to be common for GPT-4 to hallucinate details when things aren't working out or if it doesn't know a reasonable answer.

In other contexts we have observed that GPT-4 can produce better output if asked to stop itself if a particular mathematical approach does not seem to be working out and to try another approach. When prompted to do so in this particular case GPT-4 did indeed try numerous reasonable strategies but unfortunately it was still not ultimately successful. This was in part due to poor choices along the way and partially due to being plagued by algebraic errors which ended up misleading it.

For balance we mention that the failed attempts above were not entirely characteristic of GPT-4 which can in some cases produce perfect answers.

For example, Problem 27 is solved completely, although slightly inefficiently and Problem 23 is correct except for a single bogus explanation which was not particularly significant. However, it should be pointed out that Problem 27 is quite trivial, essentially requiring only the binomial theorem and Problem 23 is completely standard in many texts on elementary Number Theory. It is very unlikely that the standard proof would be hit upon at random, and the fact that GPT-4 generates it perfectly is probably an indication of the relative abundance of proofs in training material.

4.2 Problem Perturbation to Probe Memorisation

Contributed by Wenda Li

If a system simply memorizes the answers to problems, its performance can greatly differ depending on whether the problems it is evaluated on are memorized. In this section, we evaluate GPT4's performance on variations of problems from ProofWiki, which are far less likely to be appear in training data since we make novel variations. Concretely, we varied the problems in three different ways to test the model's understanding of the problems: asking for definitions of concepts, loosening assumptions of problems, and instantiating abstract variables with values. Two problem instances and their variations are presented due to them being the most interesting and revealing examples from all that were tried.

General remarks We found GPT4's performance at variations of several ProofWiki problems quite satisfactory: it can reliably retrieve definitions of concepts used in the problem as well as in its own proof; it can correctly assess whether loosening certain assumptions breaks the proof; it can also instantiate variables quite robustly, given the opportunity of inspection of its own answers. There have been debates (Bender et al., 2021b; Piantadosi and Hill, 2022) about to what extent shall we say language models "understand", given the nature of their stochastic generation. In our study, we find a couple of simple[14] cases where

[14]Note, "simple" here is relative to that of a trained mathematician.

the language-model-based assistant possesses the mathematical understanding of assumptions and variable instantiations beyond mere memorisation.

ProofWiki problem 25 Let $a, b \in \mathbb{N}_{>0}$ such that there exists no $m, n \in \mathbb{N}_{>0}$ such that $a^m = b^n$. Prove that $\log_b a$ is irrational.

Presented the problem statement above, the assistant gave a perfect answer with step-by-step calculations. To test if the assistant has a true mathematical understanding of the problem, we first asked for definitions of concepts used, and then varied the original problem by loosening some of the assumptions made, and asked the assistant for a proof in the new setting.

Asking for definitions We found that the assistant gave the correct definitions in the theorem statement as well as in its own proof. Concretely, it gave the right answers for: the definition of logarithm; the range of a logarithm's base; the meaning of the set $\mathbb{N}_{>0}$; and whether $\log_b a = \frac{p}{q}$ can be a negative number (p and q are variables arising from the assistant's own proof).

Loosening assumptions We started by asking the assistant *whether the proof still holds if we instead have $a, b \in \mathbb{R}_{>0}$?* The assistant understood the meaning of $\mathbb{R}_{>0}$ and confirmed the derivation still held, so the original lemma/proposition has been generalised (since one of its assumption has been relaxed). Later, we attempted to generalise the proposition further by dropping the assumption $a \in \mathbb{R}_{>0}$ or $b \in \mathbb{R}_{>0}$:

We continued by asking *if dropping the assumption that $b \in \mathbb{R}_{>0}$ or $a \in \mathbb{R}_{>0}$ affects the original proof?* The assistant knew that these assumptions were necessary to make the log function well-defined, and pointed out that dropping either of the assumptions would invalidate our previous derivation.

These variations, though not impossible, are unlikely to appear together with the problem in the training data of the assistant. We think the assistant does have some understanding of the underlying mathematical concepts and its own proof, in the context of this problem.

ProofWiki problem 39 Let X be a random variable. Assume $\mathsf{E}(X) = \mu$ for some $\mu \in \mathbb{R}$ and $\mathsf{var}(X) = \sigma^2$ for some $\sigma^2 \in \mathbb{R}_{>0}$. Show that for all $k > 0$: $\Pr\left(|X - \mu| \geq k\sigma\right) \leq \dfrac{1}{k^2}$.

Given this problem statement, the assistant mentioned that we can use the Chebyshev's inequality, and then re-stated the problem in an almost identical way but with different variable names. This demonstrates a certain level of variable unification, which is an important concept in automatic theorem proving.

Variable instantiation We then checked whether the assistant knew how to instantiate variables by asking it whether the proof still holds when the following concrete values are assigned to k: 2, $\sqrt{2}$, $\sqrt{2} - 1$, $\sqrt{2} - 2$, and $(\sqrt{2} - 2)^2$. Human inspection finds the assistant's behaviour to be correct. The assistant can clearly handle concrete calculations even when k is a relatively complicated number (e.g., $\sqrt{2} - 1$). The model also knows that the previous derivation cannot be carried out when $k = \sqrt{2} - 2$, a negative number.

An interesting observation arose when the assistant was not confident of its derivations: we asked: "*are you sure $(\sqrt{2} - 2)^2 > 0$?*" The answer should be affirmative, but the assistant started to apologise and revise its previous correct calculation by saying "*When $k = (\sqrt{2} - 2)^2$, the value of k is indeed non-negative, but it is actually equal to 0, not greater than 0.*" When we asked again "*Why do you say your previous statement was incorrect and $k = 0$? I don't understand.*", the assistant corrected its previous mistake and produced the right evaluation.

We found that the assistant is generally quite capable with variable instantiations and evaluating certain complex expressions, with the occasional mistake made with low confidence. We hypothesise that the mistake is a defect of its RLHF training: the human feedback is mostly assumed to be right, and when the feedback questions a true fact, the assistant concurs and alters its own (correct) response.

4.3 Investigating the Boundary between Easy and Hard Problems

Contributed by Timothy Gowers

If we want to understand how and to what extent large language models can help mathematicians, it is clearly important to understand what they can and cannot do. A range of views have been expressed on this topic, with some saying that they already show glimmerings of AGI (Bubeck et al., 2023), and others dismissing them as mere "stochastic parrots" (Bender et al., 2021a). On the latter view, the successes that LLMs undoubtedly have solving mathematical problems are to be understood as very good guesses of what the outward form of a proof would look like, unaccompanied by any genuine understanding of what that proof means, even when it is correct.

A difficulty with evaluating the level of understanding of a language model is its opacity: for any particular answer it gives, we do not know the extent to which it is merely an amalgam of texts it has seen as part of its training data and the extent to which it has actually had to build its answer in a coherent way. One way to try to get round this problem is to ask the LLM questions that are deliberately designed to be "quirky" and non-standard. A good source of such questions is ones that ask whether mathematical objects of certain kinds exist with various artificial combinations of properties.

Timothy Gowers, a mathematician from our author team, tested GPT-4 on many such questions, attempting to identify a "boundary" between what it could and could not do. This imposed a further constraint on the questions: that they should ideally be modifiable in various ways, so that one can "turn the dial" until GPT-4's performance drops.

General remarks Many of the strengths and weaknesses we observed in GPT-4 are ones that have been commented on several times (in connection with other LLMs as well). For instance, it is not good at calculation, it has a tendency to ignore facts that do not support its main conclusion (even if it itself has generated those facts), and to invent facts that do support it (Azamfirei et al., 2023).

When it comes to building examples, it has another weakness, which is that instead of using a process of reasoning to constrain what the example can look like and only then exhibiting the example, it prefers to start by exhibiting the example and then provide the justification that it has the desired properties. If its initial suggestion is correct, then this may be all right (though its justifications are not always correct), but often the example it gives is not correct, and it typically follows it up with a "justification" that to a trained mathematician has very obvious flaws. This behaviour supports the stochastic parrot view, since mathematical constructions are typically presented in the literature in the same unhelpful style – first the example, and then the verification that it works.

One can try to use prompt engineering to discourage GPT-4 from organizing its answers in this way, for example asking the model not to provide an example immediately but to describe a general strategy first, and only then the implementation of the strategy, or to pretend to be a student and ask it to offer hints. While these did improve its output somewhat, they did not make a significant enough difference to affect our general conclusions, so we mainly used more straightforward prompts.

A further feature that has a negative effect on the experience of interacting with GPT-4 is that even when it gives correct answers, it often describes them and checks their properties in a laborious case-by-case way, and often those checks are not even necessary. For instance, if asked to construct a 3-regular graph with eight vertices, it will typically list all the vertices and edges, and then for each vertex it will say what its neighbours are and comment that there are three of them. Or if it needs the fact that the matrix $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ squares to the zero matrix, instead of just asserting that it does, it will write

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

which is adding virtually nothing to a bald assertion, since no details are given of the computation of the matrix product (not that one would want them). Similarly, it sometimes "verifies" that a matrix is symmetric by writing out that matrix and its transpose and noting that the two matrices it has written out are identical.

In the more positive direction, there were several questions that GPT-4 answered well in a way that is hard to dismiss as mere parroting. To give a simple example, if it is asked to differentiate a function, it will typically do so very competently, suggesting at least some ability to generalize. (A possible explanation for why it is good at differentiation and bad at arithmetic is that there are many more worked examples of differentiation, complete with step-by-step explanations, than there are worked examples of arithmetic.) More examples will be discussed below, with some speculations about why GPT-4 is good at them.

Example 1 Here is a prompt given to GPT-4.

> Can you find an example of a set A of positive integers that satisfies the following three conditions? (The LaTeX is compiled here but was uncompiled in the prompt.)
>
> 1. $|A| = \min(A)$.
> 2. A consists of prime numbers only.
> 3. A has cardinality at least 5.

The response was the following correct and concise answer.

> Yes, here's an example of a set A of positive integers that satisfies the given conditions:
>
> A = 5, 11, 17, 23, 29
>
> The cardinality of A is 5, which is also the minimum element in A.
>
> 1. All elements in A are prime numbers.
> 2. The cardinality of A is 5, which is at least 5.
> 3. Therefore, set A = 5, 11, 17, 23, 29 meets all three conditions.

Despite the fact that this answer is correct, it has a revealing feature. Why, one might ask, does GPT-4 not give the more obvious answer $\{5, 7, 11, 13, 17\}$? That is, why does it not take the five smallest prime numbers greater than or equal to 5?

A clue lies in the fact that the set it does choose, $\{5, 11, 17, 23, 29\}$ has the additional interesting property that it forms an arithmetic progression. This property has led over the years to the set being mentioned in many articles about arithmetic progressions that consist of primes, and presumably a significant fraction of those also mention that the number of elements is 5.

What is illustrated here is a preference by GPT-4 to use off-the-shelf examples, a preference that is strong enough that it will often do so even when the examples do not in fact have the required properties.

Why would most human mathematicians prefer to take the first five primes, starting at 5? It is because that example is easily generalisable: if we change the number 5 to any other number n, we simply have to choose a prime p that is at least as big as n and take the first p prime numbers from p onward.

To understand GPT-4's behaviour better, it is therefore natural to give it the following slightly modified prompt.

> Can you find an example of a set A of positive integers that satisfies the following three conditions?
>
> 1. $|A| = \min(A)$.
> 2. A consists of prime numbers only.
> 3. A has cardinality at least 6.

The response was

> Yes, here is an example of a set A of positive integers that satisfies the given conditions:
>
> A = 7, 11, 13, 17, 19, 23
>
> 1. |A| = 6, and the minimum element in A is 7, so |A| = min(A).
>
> 2. All elements in A are prime numbers: 7, 11, 13, 17, 19, and 23. 3. A has cardinality 6, which is at least 6.
>
> The set A satisfies all three conditions.

Even after its error was pointed out, and a few more rounds of the conversation had taken place, GPT-4 did not manage to eliminate all mistakes and give a correct answer. Indeed, when told that 6 does not equal 7, it at first tried to explain why no such set could exist.

This example lends some weight to the stochastic-parrot view: the problem is extremely easy for a human, but it is also a sufficiently artificial question that GPT-4 will not be likely to have encountered it, or anything very like it, in the texts on which it has been trained. One might argue that it does at least give a set of consecutive primes this time, but that again is what one would expect, given that sets of consecutive primes will appear much more frequently in the literature than other sets (a tendency that one might guess is overridden when the sets start with 5 and have size 5).

Example 2 This was a successful attempt to trick GPT-4. It was first asked to evaluate the integral $\int_{-1}^{1} x^3 \cos x \, dx$. It did so by a laborious argument using iterated integration by parts, presumably because the integrand is typical of the integrands in integration-by-parts exercises. After getting everything right until close to the end, it made an elementary mistake, but with a bit of help it arrived at the correct answer, 0 (though still persisting in some of its incorrect calculations). On being asked whether there was a shorter argument, it gave the right response: that the integrand is an odd function and the integral symmetric about 0, so the integral is 0.

Then it was asked to evaluate the integral $\int_{-1}^{1}(x^2 - 1/3)\, dx$. Again it did so correctly, obtaining the answer 0. On being asked whether there was a simple reason for this, it once again pointed out that the integrand was an odd function, and even supplied a bogus proof that the function is odd.

This behaviour again fits the stochastic-parrot hypothesis quite well: almost always if one is asked for a simple reason that the integral of a function over a symmetric interval is zero, the correct response is that the function is odd. Whether or not it actually is odd is for GPT-4 a secondary consideration.

Example 3 A nice problem that mathematics undergraduates tend to find quite hard is to determine whether there is an order-preserving bijection between the rational numbers and the dyadic rationals. Surprisingly, the answer is yes, and the proof is by what is known as a back-and-forth argument. That is, one enumerates the rationals and the dyadic rationals, and then one alternates between choosing a match for the first unmatched rational and the first unmatched dyadic rational, making sure at each stage that the order is preserved.

When GPT-4 was asked the problem, it tried to prove that no such bijection could exist. Each time its mistakes were pointed out to it, it replaced its bogus argument by a slightly modified bogus argument.

Upon being asked whether it knew about back-and-forth arguments, it said that it did, and explained that they could be used to prove a theorem of Cantor, that any two countable dense subsets of the real numbers are order isomorphic. It did not seem to realise that the problem it had been asked was a special case of this theorem. It also tried to explain why one could not use a back-and-forth argument to prove that the rationals and the dyadic rationals are order isomorphic, but after its explanation was criticised, it proceed to give a convincing sketch of how such an argument would indeed work.

Example 4 When GPT-4 was asked whether there is a positive integer n such that $n + k$ is divisible by k for every integer k in the range $\{1, 2, \ldots, 100\}$, it said no, and offered bogus proofs. After being steered towards a positive answer via certain easier questions, it suggested taking n to be $\mathrm{LCM}(1, 2, \ldots, k) - 1$. This was interesting because $\mathrm{LCM}(1, 2, \ldots, k)$ would have been a correct answer, but the pointless subtraction of 1 ruined it. Furthermore, it gave an argument that would have been correct if the -1 had not been present.

What might explain this act of self sabotage? One idea is that GPT-4 is influenced by Euclid's proof that there are infinitely many primes, which assumes that $p_1, \ldots, p_k$ are all the primes and considers the number $p_1 p_2 \ldots p_k + 1$. An alternative argument would be to consider the number $\mathrm{LCM}(1, 2, \ldots, k) - 1$, where k is at least as large as the largest prime.

However, this explanation is rather speculative, and a Google search does not seem to back it up. When GPT-4 was asked why it had subtracted 1, it did not provide a convincing reason either.

More revealing was its behaviour when its example was criticised on the grounds that, for instance, $\mathrm{LCM}(1, 2, \ldots, k) - 1$ is odd, and therefore not divisible by 2. Instead of adjusting its answer, as a human mathematician might, it decided that no such n existed, and when its arguments for that conclusion were criticised, it went back to the example of $\mathrm{LCM}(1, 2, \ldots, k) - 1$. Even when asked whether $\mathrm{LCM}(1, 2, \ldots, k)$ would work better, it initially said no. So this was not really a "boundary" example, and more just a problem on which GPT-4 got into a rut and could not get out of it.

Example 5 A better example of a "boundary" problem was the following question: does there exist a graph with eight vertices such that every vertex has degree 3? Once again GPT-4 demonstrated its liking for off-the-shelf examples, giving the example of the 3-dimensional discrete cube. (An alternative approach is to take eight vertices joined in a cycle, and to join each vertex in addition to the vertex opposite it in the cycle.)

When asked whether there was a graph with eight vertices such that every vertex has degree 5, it performed far worse. It did not know of any off-the-shelf examples, and was probably incapable of tricks such as taking the complement of an 8-cycle (which works because in the cycle every vertex has two neighbours, so in the complement of the cycle it has five neighbours). That is, it does not appear to be capable of taking an off-the-shelf example and *modifying* it in a suitable way. Instead, it resorted to listing the vertices as A, B, C, D, E, F, G and H, and for each vertex giving a list of its neighbours. The trouble is that this kind of approach gave it many opportunities to fail as a result of familiar weaknesses such as a propensity to make calculation errors or to write down inconsistent statements. For instance, over its several attempts it would frequently list a vertex v as a neighbour of another vertex w, but without listing w as a neighbour of v. Eventually, probably with a slice of luck, it came up with an example that turned out to be the complement of the disjoint union of a 3-cycle and a 5-cycle. (Since the complement has to be regular of degree 2, it will always be a disjoint union of cycles.)

Example 6 It has been noted that GPT-4 likes well-known patterns, and that one way of inducing it to fail is to ask it questions that will tempt it to give answers that fit those patterns. The following attempt to lead it astray in that way was a partial success. It was asked to find integers a and b such that the sequence $(1, 3, a, 7, 9, b, 13, 15)$ is strictly increasing but not an arithmetic progression. It responded by choosing $a = 5$ and $b = 11$, thereby falling headlong into the trap. However, it then did a check by calculating all the successive differences. On observing that it obtained the difference sequence $(2, 2, 2, 2, 2, 2, 2)$, it then modified its choice of b to 12, after which it recalculated the difference sequence, obtaining $(2, 2, 2, 2, 3, 1, 2)$ and declaring itself satisfied.

This was another example where despite arriving at the correct answer, GPT-4 argued in a very non-human way. The main non-human feature was of course that it began by making the one guess that it needed to avoid (out of the strictly increasing possibilities). However, the whole approach of guessing and then verifying is inappropriate for the problem, since it is much more efficient to reason as follows: first, we note that *if* the sequence is to be an arithmetic progression, then it will have to have common difference 2 (since the first two terms differ by 2) so it is sufficient to ensure that $a \neq 5$. This kind of forward planning appears to be beyond the current capabilities of GPT-4, (though maybe it could be induced to some small extent with careful prompt engineering).

We briefly mention its response to a variant of the problem, where it was asked whether it was possible to find integers a, b and c such that the sequence $(1, a, b, c, 14)$ is an arithmetic progression. It answered yes, then set d to be the common difference, obtained the equation $14 = 1 + 4d$, solved for d, discovered that d was not an integer, and answered no, having apparently forgotten that it had previously answered yes. This showed a reluctance to plan in advance even in a situation where it was entirely capable of carrying out the required planning.

Example 7 GPT-4 was asked to find a colouring of the set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ using three colours and satisfying the conditions that each colour is used three times, and no integer n has the same colour as $n + 1$ or $n + 3$. The obvious way to solve this problem is trial and error, which for a typical human will work with a small amount of backtracking. It did indeed choose this approach at first, but failed quite badly because it was unable to check the conditions properly, which caused it to assign colours that were forbidden by the conditions, and, in the other direction, to claim that certain choices were forced when they were not. After a couple of failed attempts, it switched to trying to find a "systematic" approach. One such attempt was to split the set into even and odd numbers, but that did not help it find a correct colouring. It even tried splitting the numbers into the three sets $\{1,4,7\}, \{2,5,8\}$ and $\{3,6,9\}$ and assigning one colour to each set, which violated the $n+3$ condition many times over.

In sum, its ability to check simple conditions was too unreliable for it to be able to push through a trial-and-error approach, and for this problem a guess-and-check approach has a very low chance of success.

Example 8 A somewhat similar question on which it performed badly was to find a sequence of nine distinct integers with no increasing or decreasing subsequence of length 4. Here it once again showed its taste for patterns: the problem was that it did not choose appropriate patterns. An example that was typical of its output was $(4,1,5,2,6,3,7,8,9)$. Interestingly, when, after a few failures, it was given a reason-step-by-step prompt, it produced the same example, this time after talking about interleaving sequences – an idea which, if used correctly, leads to solutions such as $(1,4,7,2,5,8,3,6,9)$. However, encouraging it to use interleaving just led to further incorrect guesswork, an extreme example of which was when it interleaved the sequences $(1,3,5)$ and $(2,4,6,7)$ to obtain $(1,2,3,4,5,6,7)$, stuck 8 and 9 on the end, and proposed $(1,2,3,4,5,6,7,8,9)$ as a solution (complete with a "check" that it worked).

When given the hint that it might like to start its sequence with $(7,8,9)$, it immediately made the obvious suggestion $(7,8,9,4,5,6,1,2,3)$. When asked for a rigorous proof that this sequence has the desired property, it gave an inadequate answer, stating correctly that the longest increasing subsequences are those that begin with $7,4$ and 1, and stating incorrectly that the longest decreasing subsequences start with 9, 6 and 3. There was nothing resembling the argument that a decreasing sequence can contain at most one term out of each of the subsequences $(7,8,9), (4,5,6)$ and $(1,2,3)$.

Example 9 The following is a question on which GPT-4's performance was more or less perfect. It was asked to evaluate the sum $\sum_{r=2}^{n} \log(1-1/r)$. It noticed that $1-1/r$ can be rewritten as $(r-1)/r$, which therefore leads to a telescoping sum and and eventual answer of $\log(1/n)$. It did not give a detailed explanation of why, as it put it, the denominator in each term cancelled with the numerator in the succeeding term, but when asked for one it gave a good one (though it argued for a telescoping product inside the logarithm rather than a telescoping sum, which might have been simpler).

This was interesting because it required putting together two tricks: one to rewrite $1-1/r$ and the second to use the telescoping. However, its performance on a follow-up question strongly suggested that it had seen the question in its training data. It was asked whether it could evaluate the sum $\sum_{m=1}^{n} \log(1/m(m+1))$. This question was intended to trick it, since the sum $\sum_{m=1}^{n} 1/m(m+1)$ is famously evaluated by rewriting $1/m(m+1)$ as $1/m - 1/(m+1)$ in order to obtain a telescoping sum. However, instead of falling into the intended trap, it stated falsely that it could rewrite the sum as $\sum_{m=1}^{n} \log((m+1)/m)$, after which it obtained a telescoping sum very similar to that of the previous question.

When asked a further follow-up question to test how well it understood the principle of telescoping sums, it gave an answer that had good points and bad points, both of which indicated that it was good at capturing the look of a correct answer while not operating at the level of the underlying concepts. The question was to determine the sum $\sum_{n=1}^{N} g(n)$, when $g(n)$ had been defined to be $f(n)-2f(n+1)+f(n+2)$. It is instructive to look at its complete answer.

> Let's work out the sum. We have
>
> $$\sum_{n=1}^{N} g(n) = \sum_{n=1}^{N} (f(n) - 2f(n+1) + f(n+2))$$
>
> We can break this into three separate sums:

$$\sum_{n=1}^{N} f(n) - 2\sum_{n=1}^{N} f(n+1) + \sum_{n=1}^{N} f(n+2)$$

Let's shift the indices in the second and third sums to make the limits match up with the first sum:

1. For the second sum, we substitute $m = n + 1$, so $n = m - 1$. When $n = 1$, $m = 2$; when $n = N$, $m = N + 1$.
2. For the third sum, we substitute $m = n + 2$, so $n = m - 2$. When $n = 1$, $m = 3$; when $n = N$, $m = N + 2$.

Doing this, we get:

$$\sum_{n=1}^{N} f(n) - 2\sum_{m=2}^{N+1} f(m) + \sum_{m=3}^{N+2} f(m)$$

Now, we can write out the sums explicitly to see what's happening:

1. The first sum is: $f(1) + f(2) + f(3) + \cdots + f(N)$.
2. The second sum is: $2f(2) + 2f(3) + \cdots + 2f(N + 1)$.
3. The third sum is: $f(3) + f(4) + \cdots + f(N + 2)$.

By looking at these sums, we can see that most of the terms will cancel out. For example, the term $f(2)$ is added in the first sum and subtracted twice in the second sum. Similarly, the term $f(3)$ is added in the first sum, subtracted twice in the second sum, and added once in the third sum.

After all the cancellations, we are left with:

$$= f(1) - 2f(2) + f(N) - 2f(N + 1) + f(N + 2)$$

So, the sum $\sum_{n=1}^{N} g(n)$ simplifies to $f(1) - 2f(2) + f(N) - 2f(N + 1) + f(N + 2)$.

A first comment is that the final answer here is not just wrong, but clearly wrong to an experienced mathematician, because there is a symmetry in the problem that guarantees that the number of leftover terms at the two ends will be the same, whereas in GPT-4's answer there are two terms at the beginning and three at the end. Given its performance on other problems, it is no surprise that GPT-4 does not spot this anomaly.

On a more positive note, the first two steps – splitting up the sum into three parts and changing variables – are exactly the right thing to do, even if the purpose of changing of variables is not to get the limits to match up (in fact, it does the opposite). However, it then abandons this promising beginning in favour of writing out the sums in a more transparent notation, a move that is striking because it is what many human mathematicians would do if they did not immediately see how to proceed (but presumably GPT-4 has seen texts where this style of explanation is given for why telescoping sums work). It then correctly notes various cancellations, but demonstrates that it is not really understanding what it is saying by jumping to a final answer that is both incorrect and easily seen to be inconsistent with what it has just said. For instance, it notes that $f(2)$ is added once and subtracted twice, but its final answer includes a term $-2f(2)$ instead of $-f(2)$.

4.4 Qualitative Observations from Case Studies

We next synthesise core insights from the case studies. Similar to `MathConverse`, we see that the models can be good at providing definitions and interestingly can produce helpful scaffolding for a problem (e.g., the right strategy or first few steps), even if details of the full proof are incorrect. Moreover, even if the model does produce a correct solution, this solution is not always arrived at in a seemingly "human-like" way; for instance, the model may follow a guess-and-check approach rather than forward planning (e.g., Examples 1,4, and 7 in Section 4.3). However, guess-and-check cannot work well if one cannot "check" solutions; indeed, we see that in general, challenges with algebraic manipulation plague in- and out-of-distribution performance (e.g., all three examples in Section 4.1). By probing GPT-4 capabilities on slightly novel problems or those which involve building examples, we notice the model's tendency to over-rely on memorised examples or patterns. The case studies also reiterate potential issues with handling user corrections as well as a tendency towards over-verbosity. Further, one case study illuminated intriguing behaviour when the model was queried about uncertainty; i.e., the model began to apologise despite having been correct (see Section 4.2).

5 Taking Stock and Looking Ahead

We compile key observations from Sections 3 and 4 into a series of actionable takeaways, which – given the interdisciplinary nature of our study – we hope will appeal to a wide audience. We tailor these takeaways to audiences from different fields. To offer balance, we first note that the best LLMs we investigate do demonstrate *some non-trivial ability* in collaborating helpfully and correctly with users on undergraduate-level mathematical problems (see Figure 3). Should the user be able to assess the validity of LLM-generated responses, they can meaningfully assist on some problems. Even if the answers are memorised and can be found somewhere on the internet, LLMs have the advantage of being flexible in their inputs and outputs over traditional search engines.

5.1 Takeaways for ML Developers

Enable Models to Communicate *Calibrated* Uncertainty and Uptake Corrections We observe cases where people attempted to correct the model when it made an error, the model apologised, and proceeded to give an answer without the necessary corrections or asking for clarification (e.g., Example 1 from Section 4.3). The pattern often repeated itself until the user seemed to get bored and abort. To improve user experience, systems that can adequately respond to user corrections, for example, through uncertainty calibration (Hullman et al., 2018; Liu et al., 2016; Vasconcelos et al., 2023), are compelling (Akyürek et al., 2023; Kocielnik et al., 2019; Meng et al., 2022; Mitchell et al., 2022; Wilder et al., 2021b). Indeed, in the models we explored, it was not clear when the model was unsure. We include a discussion with participants about these challenges in a post-survey questionnaire (see Appendix D). Communicating uncertainty is critical to ensure users know when they can trust the model output (Bhatt et al., 2021; Hullman et al., 2018) and help calibrate appropriate levels of trust (Spiegelhalter, 2017; Zerilli et al., 2022). However, obtaining accurate, calibrated uncertainty estimates from LLMs can be a difficult endeavour (Si et al., 2022; Xiao et al., 2022).

Enable Provision of Rationales Several participants in `MathConverse` asked "why" a model undertook a particular proof move. Expanding on the justification for a choice could be a valuable educational tool. Generating compelling explanations, on-the-fly and on-request – provided those explanations are indeed representative and not misleading (Bhatt et al., 2020; Kıcıman et al., 2023; Sevastjanova and El-Assady, 2022; Wu et al., 2023; Zelikman et al., 2022b) – seem promising and desirable to explore to further boost the utility of these systems in partnership with mathematicians.

Strive for Conciseness Further, both our survey (see Section 3.3.3) and our expert case studies (see General remarks, Section 4.3) find that – while mathematical correctness appears to often be a foundation for useful assistance in higher-level mathematics – it is not always sufficient. Responses that were overly verbose were sometimes deemed less helpful. Designing systems that generate concise responses to mathematical queries seems a promising future direction, best also coupled with the capability of showing its "work" if needed (related to rationales, see above). The applicability of this to other domains than mathematics remains to be investigated: It may be that responses of different degrees of verbosity are preferred in different domains, e.g., in medicine, longer responses laden in empathy may be preferable (Ayers et al., 2023).

5.2 Takeaways for Mathematicians (Students, Educators, and Researchers)

Pay attention! Large language models are capable of generating remarkably compelling natural language – an incredible technical feat which ought not to be dismissed and can be helpful as we see in both our studies; however, such prowess belies the potential for coaxing the reader into not recognising errors. Be careful not to fall into the trap of lazy checking (see Appendix D in participant testimonials). This is worth keeping in mind for users learning from or evaluating the generations of LLMs, for example, students and assignment markers. It is worth being cognisant of risk of automation bias (Cummings, 2004).

Take a Nuanced View of When These Models Can Help Reinforcing similar findings from (Frieder et al., 2023), we observe in this work that LLMs can be useful for retrieving definitions (see Appendix C.4) and can occasionally provide a valuable scaffolding for how to approach a problem (see Sections 3.3 and 4.3). It is important not to overgeneralise model performance in one realm of the task space to another (Bhatt et al., 2023; Kelly et al., 2023). Counterintuitively – a la Moravec's Paradox (Moravec, 1988) – it is possible that models will succeed at human-perceived challenging tasks, but fail at tasks humans consider easy (e.g., derivation versus algebraic manipulation, see Section 4.3).

Be Cautious When Using Current LLMs (Alone) for Heavy Algebra In particular, our studies further underscore the challenges of present models at algebraic manipulation, corroborating prior work (Bubeck et al., 2023; Frieder et al., 2023). We believe it is therefore important that mathematicians take care if using these systems for tasks which involve substantial algebra. We do not explore plug-ins (OpenAI, 2023a) in this paper, nor alternative hybrid neuro-symbolic approaches, e.g., (Gowers, 2022; Jiang et al., 2022b; Kazemi et al., 2022; Li et al., 2022; Poesia and Goodman, 2022), which may prove a useful salve for some of this failure mode.

5.3 Takeaways for LLM Development, Evaluation, and Deployment

We conclude with broad takeaways for anyone evaluating or considering deploying LLMs in practice.

Carefully Discern When Assistance is Needed (or Even Worth Utilising) To build complementary systems (Wilder et al., 2021a), understanding when an AI-based assistant is helpful is of utmost importance: seldom will such an assistant be helpful in all settings (Bhatt et al., 2023). An important question will be in which settings such an assistant can be useful without undermining the agency of the mathematician, for example, of the kind already being proposed when considering using LLMs in coursework (Ba and Wang, 2023). Future work would benefit from considering how to build usable assistants that optimise for complementarity, providing support as and when needed (Miller, 2023).

Collaboration between ML Practitioners and Domain Experts is Valuable Conducting investigations in partnership with domain experts can be especially fruitful for characterising model behaviour (Davies et al., 2021; McGrath et al., 2022; Mirowski et al., 2023), particularly by designing entirely new tasks, as we demonstrate in Section 4.3. We encourage forming such interdisciplinary partnerships in and beyond mathematics.

Incorporate Interactivity into LLM Capability Assessments To truly comprehend the landscape of an LLM's capabilities, we believe it is paramount to incorporate interactive evaluations. Our work further drives home the importance of interactive evaluation as a way to gain deeper insights into the strengths and weaknesses of these models and probe characteristics which may be preferable for assistive settings. However, as we highlight here, interactive study of LLMs not only serves to characterise model behaviour – but identifies ways in which *humans* may themselves choose to interact with these models and actually use these systems (Ringer et al., 2020). A wave of works increasingly illuminate the sensitivity of these models to the choice of prompts (Wei et al., 2022; Yao et al., 2023; Zhou et al., 2023). As such, it is important to consider the form and content of queries that humans may use to interact with these systems both to design systems more adapted to particular user queries, and to inform users of best practices. It may be valuable for system maintainers to recognise whether or not users are leveraging these tactics to help better inform the techniques to boost the quality of the response for their query.

We hope to see more works like ours and (Lee et al., 2022a,c; Mirowski et al., 2023) that study LLMs in the context of human-computer interactions. `CheckMate` offers a place to start, potentially complemented by free-form evaluation of the kind we conduct in our expert case studies.

6 Conclusion

As LLMs are increasingly deployed in human-facing settings where they may serve as assistants, it is paramount that evaluation of their efficacy fundamentally includes evaluation in an interactive context (Lee et al., 2022b). As we demonstrate, these interactive evaluations can be structured (e.g., leveraging `CheckMate`) or free-form (e.g., through sourced domain expert, or target user, interactions). LLMs, and foundation models broadly, are complex and often surprising in their behaviour; so are humans. Hence characterising potential failure modes – and opportunities ripe for great success – in LLM and human interactions, necessitates a multi-factorial evaluation approach, which includes both interactive evaluation and classical, static-snapshot evaluation (Burnell et al., 2023). Through our study, we extract insights which we hope can inform careful design and deployment when considering leveraging LLM-based mathematics assistants and reasoning engines. We believe that our study paves the way for further evaluation of the use of foundation models in mathematics and other domains, particularly through closer collaboration with domain experts.

Acknowledgements

Special thanks to Fabian Gloeckle, for support and informative discussions throughout the project. We additionally thank in no particular order Anshula Gandhi, Jonas Bayer, Angeliki Koutsoukou-Argyraki, Fredy Yip, Mariusz Mirek, Gabriel Poesia, Noah Goodman, Valerie Chen, Nina Grgić-Hlača, Ilia Sucholutsky, and Shoaib Ahmed Siddiqui for valuable discussions around assistive systems in AI and mathematics, and LLM evaluation.

KMC gratefully acknowledges support from the Marshall Commission and the Cambridge Trust; note: KMC is a part-time remote Student Researcher at Google; however, this work does not represent any of her work at Google nor the views of Google and was conducted entirely outside of the position. The study was formulated prior. AQJ acknowledges support from the Peterhouse Graduate Studentship with gratitude. SF and TL acknowledge support from the Alan Turing Institute under the EPSRC grant EP/N510129/1. MZ acknowledges support from the Leverhulme Trust grant ECF-2021-429. UB acknowledges support from DeepMind and the Leverhulme Trust via the Leverhulme Centre for the Future of Intelligence (CFI), and from ELSA: European Lighthouse on Secure and Safe AI via the Alan Turing Institute. TL acknowledges support from the AXA Research Fund and the EU TAILOR grant 952215. WL is supported by the ERC Advanced Grant ALEXANDRIA (Project GA 742178). AW acknowledges support from a Turing AI Fellowship under grant EP/V025279/1, The Alan Turing Institute, and the Leverhulme Trust via CFI. MJ acknowledges support from the EPSRC under grant EP/T019603/1.

References

A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569*, 2019.

J. S. Aitken, P. Gray, T. Melham, and M. Thomas. Interactive theorem proving: An empirical study of user activity. *Journal of Symbolic Computation*, 25(2):263–284, 1998.

A. F. Akyürek, E. Akyürek, A. Madaan, A. Kalyan, P. Clark, D. Wijaya, and N. Tandon. RL4F: Generating Natural Language Feedback with Reinforcement Learning for Repairing Model Outputs. *arXiv preprint arXiv:2305.08844*, 2023.

A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1245. URL https://doi.org/10.18653/v1/n19-1245.

R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. H. Clark, L. E. Shafey, Y. Huang, K. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. Botha, J. Bradbury, S. Brahma, K. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crepy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. Díaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. Garcia, S. Gehrmann, L. Gonzalez, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M. Li, W. Li, Y. Li, J. Li, H. Lim, H. Lin, Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, A. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. C. Ros, A. Roy, B. Saeta, R. Samuel, R. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, and Y. Wu. PaLM 2 Technical Report, 2023.

J. W. Ayers, A. Poliak, M. Dredze, E. C. Leas, Z. Zhu, J. B. Kelley, D. J. Faix, A. M. Goodman, C. A. Longhurst, M. Hogarth, and D. M. Smith. Comparing Physician and Artificial Intelligence Chatbot Responses to Patient

Questions Posted to a Public Social Media Forum. *JAMA Internal Medicine*, 04 2023. ISSN 2168-6106. doi: 10.1001/jamainternmed.2023.1838. URL https://doi.org/10.1001/jamainternmed.2023.1838.

R. Azamfirei, S. R. Kudchadkar, and J. Fackler. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2, 2023.

J. Ba and B. Wang. Csc413/2516 winter 2023 university of toronto, assignment 1, 2023. URL https://uoft-csc413.github.io/2023/assets/assignments/a1.pdf.

B. Beckert, S. Grebing, and F. Böhl. A usability evaluation of interactive theorem provers using focus groups. In *Software Engineering and Formal Methods: SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers 12*, pages 3–19. Springer, 2015.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021a.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922.

J. Bethlehem. Selection bias in web surveys. *International statistical review*, 78(2):161–188, 2010.

U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657, 2020.

U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, P. Sattigeri, R. Fogliato, G. Melançon, R. Krishnan, J. Stanley, O. Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413, 2021.

U. Bhatt, V. Chen, K. M. Collins, P. Kamalaruban, E. Kallina, A. Weller, and A. Talwalkar. Learning personalized decision support policies. *arXiv e-prints*, pages arXiv–2304, 2023.

P. A. Bishop and R. L. Herron. Use and misuse of the likert item responses and other ordinal measures. *International journal of exercise science*, 8(3):297, 2015.

W. W. Bledsoe. Non-resolution theorem proving. *Artif. Intell.*, 9:1–35, 1977.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

A. Bundy. The computer modelling of mathematical reasoning. 1983.

A. Bundy. The use of explicit plans to guide inductive proofs. In *CADE*, 1988.

A. Bundy, A. Stevens, F. V. Harmelen, A. Ireland, and A. Smaill. Rippling: A heuristic for guiding inductive proofs. *Artif. Intell.*, 62:185–253, 1993.

R. Burnell, W. Schellaert, J. Burden, T. D. Ullman, F. Martinez-Plumed, J. B. Tenenbaum, D. Rutar, L. G. Cheke, J. Sohl-Dickstein, M. Mitchell, D. Kiela, M. Shanahan, E. M. Voorhees, A. G. Cohn, J. Z. Leibo, and J. Hernandez-Orallo. Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138, 2023. doi: 10.1126/science.adf6369. URL https://www.science.org/doi/abs/10.1126/science.adf6369.

N. C. Carter and K. G. Monks. Lurch: a word processor that can grade students' proofs. In C. Lange, D. Aspinall, J. Carette, J. H. Davenport, A. Kohlhase, M. Kohlhase, P. Libbrecht, P. Quaresma, F. Rabe, P. Sojka, I. Whiteside, and W. Windsteiger, editors, *Joint Proceedings of the MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at CICM, Bath, UK*, volume 1010 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013. URL https://ceur-ws.org/Vol-1010/paper-04.pdf.

E. Clark, A. S. Ross, C. Tan, Y. Ji, and N. A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of IUI*, 2018.

K. Cobbe, V. Kosaraju, M. Bavarian, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

A. G. Cohn and J. Hernandez-Orallo. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms. *arXiv preprint arXiv:2304.11164*, 2023.

M. L. Cummings. Automation bias in intelligent time critical decision support systems. 2004.

A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.

M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

L. M. de Moura and N. S. Bjørner. Z3: An efficient smt solver. In *International Conference on Tools and Algorithms for Construction and Analysis of Systems*, 2008.

D. Delahaye. A tactic language for the system coq. In M. Parigot and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, 7th International Conference, LPAR 2000, Reunion Island, France, November 11-12, 2000, Proceedings*, volume 1955 of *Lecture Notes in Computer Science*, pages 85–95. Springer, 2000. doi: 10.1007/3-540-44404-1_7. URL https://doi.org/10.1007/3-540-44404-1_7.

D. Dohan, W. Xu, A. Lewkowycz, J. Austin, D. Bieber, R. G. Lopes, Y. Wu, H. Michalewski, R. A. Saurous, J. Sohl-Dickstein, et al. Language model cascades. *ICML*, 2022.

A. P. Felty. Implementing tactics and tacticals in a higher-order logic programming language. *J. Autom. Reason.*, 11(1):41–81, 1993. doi: 10.1007/BF00881900. URL https://doi.org/10.1007/BF00881900.

E. First, M. N. Rabe, T. Ringer, and Y. Brun. Baldur: Whole-proof generation and repair with large language models. *CoRR*, abs/2303.04910, 2023. doi: 10.48550/arXiv.2303.04910. URL https://doi.org/10.48550/arXiv.2303.04910.

S. Frieder, L. Pinchetti, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, A. Chevalier, and J. Berner. Mathematical Capabilities of ChatGPT, 2023.

M. Ganesalingam and W. T. Gowers. A fully automatic problem solver with human-style output. *CoRR*, abs/1309.4501, 2013. URL http://arxiv.org/abs/1309.4501.

Github. Github copilot · your ai pair programmer, 2021. URL https://github.com/features/copilot/.

O. Golovneva, M. Chen, S. Poff, M. Corredor, L. Zettlemoyer, M. Fazel-Zarandi, and A. Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step reasoning. *CoRR*, abs/2212.07919, 2022. doi: 10.48550/arXiv.2212.07919. URL https://doi.org/10.48550/arXiv.2212.07919.

W. T. Gowers. How can it be feasible to find proofs?, 2022. URL https://drive.google.com/file/d/1-FFa6nMVg18m1zPtoAQrFalwpx2YaGK4/view.

T. Hales. The flyspeck project, Aug 2014. URL https://code.google.com/archive/p/flyspeck/.

T. C. Hales. A proof of the kepler conjecture. *Annals of Mathematics*, 162:1063–1183, 2005.

J. M. Han, J. Rute, Y. Wu, E. W. Ayers, and S. Polu. Proof artifact co-training for theorem proving with language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=rpxJc9j04U.

J. Harrison, J. Urban, and F. Wiedijk. History of interactive theorem proving. In *Computational Logic*, volume 9, pages 135–214, 2014.

Hugging Face. Hugging face spaces, 2021. URL https://huggingface.co/spaces.

J. Hullman, X. Qiao, M. Correll, A. Kale, and M. Kay. In pursuit of error: A survey of uncertainty visualization evaluation. *IEEE transactions on visualization and computer graphics*, 25(1):903–913, 2018.

A. Q. Jiang, W. Li, S. Tworkowski, K. Czechowski, T. Odrzygózdz, P. Milos, Y. Wu, and M. Jamnik. Thor: Wielding hammers to integrate language models and automated theorem provers. *CoRR*, abs/2205.10893, 2022a. doi: 10.48550/arXiv.2205.10893. URL https://doi.org/10.48550/arXiv.2205.10893.

A. Q. Jiang, S. Welleck, J. P. Zhou, W. Li, J. Liu, M. Jamnik, T. Lacroix, Y. Wu, and G. Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *CoRR*, abs/2210.12283, 2022b. doi: 10.48550/arXiv.2210.12283. URL https://doi.org/10.48550/arXiv.2210.12283.

S. M. Kazemi, N. Kim, D. Bhatia, X. Xu, and D. Ramachandran. Lambada: Backward chaining for automated reasoning in natural language. *arXiv preprint arXiv:2212.13894*, 2022.

M. Kelly, A. Kumar, P. Smyth, and M. Steyvers. Capturing humans' mental models of ai: An item response theory approach. *FAccT*, 2023.

R. Kocielnik, S. Amershi, and P. N. Bennett. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, and A. Mattick. Openassistant conversations – democratizing large language model alignment, 2023.

E. Kıcıman, R. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality, 2023.

G. Lample, T. Lacroix, M.-A. Lachaux, A. Rodriguez, A. Hayat, T. Lavril, G. Ebner, and X. Martinet. Hypertree proof search for neural theorem proving. *Advances in Neural Information Processing Systems*, 35:26337–26349, 2022.

M. Lee, P. Liang, and Q. Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022a.

M. Lee, M. Srivastava, A. Hardy, J. Thickstun, E. Durmus, A. Paranjape, I. Gerard-Ursin, X. L. Li, F. Ladhak, F. Rong, R. E. Wang, M. Kwon, J. S. Park, H. Cao, T. Lee, R. Bommasani, M. Bernstein, and P. Liang. Evaluating human-language model interaction, 2022b. URL https://arxiv.org/abs/2212.09746.

M. Lee, M. Srivastava, A. Hardy, J. Thickstun, E. Durmus, A. Paranjape, I. Gerard-Ursin, X. L. Li, F. Ladhak, F. Rong, et al. Evaluating human-language model interaction. *arXiv preprint arXiv:2212.09746*, 2022c.

A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, and V. Misra. Solving quantitative reasoning problems with language models. *CoRR*, abs/2206.14858, 2022. doi: 10.48550/arXiv.2206.14858. URL https://doi.org/10.48550/arXiv.2206.14858.

B. Z. Li, W. Chen, P. Sharma, and J. Andreas. Lampp: Language models as probabilistic priors for perception and action. *arXiv e-prints*, pages arXiv–2302, 2023.

Z. Li, G. Poesia, O. Costilla-Reyes, N. D. Goodman, and A. Solar-Lezama. LEMMA: bootstrapping high-level mathematical reasoning with learned symbolic abstractions. *CoRR*, abs/2211.08671, 2022. doi: 10.48550/arXiv.2211.08671. URL https://doi.org/10.48550/arXiv.2211.08671.

B. Lipkin, L. Wong, G. Grand, and J. B. Tenenbaum. Evaluating statistical language models as pragmatic reasoners. *arXiv preprint arXiv:2305.01020*, 2023.

B. F. Liu, L. Bartz, and N. Duke. Communicating crisis uncertainty: A review of the knowledge gaps. *Public relations review*, 42(3):479–487, 2016.

P. Lu, L. Qiu, W. Yu, S. Welleck, and K. Chang. A survey of deep learning for mathematical reasoning. *CoRR*, abs/2212.10535, 2022. doi: 10.48550/arXiv.2212.10535. URL https://doi.org/10.48550/arXiv.2212.10535.

T. McGrath, A. Kapishnikov, N. Tomašev, A. Pearce, M. Wattenberg, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

M. Mikula, S. Antoniak, S. Tworkowski, A. Q. Jiang, J. P. Zhou, C. Szegedy, L. Kucinski, P. Milos, and Y. Wu. Magnushammer: A transformer-based approach to premise selection. *CoRR*, abs/2303.04488, 2023. doi: 10.48550/arXiv.2303.04488. URL https://doi.org/10.48550/arXiv.2303.04488.

T. Miller. Explainable ai is dead, long live explainable ai! hypothesis-driven decision support, 2023.

P. Mirowski, K. W. Mathewson, J. Pittman, and R. Evans. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581225. URL https://doi.org/10.1145/3544548.3581225.

E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

H. Moravec. *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz. Reading between the lines: Modeling user behavior and costs in ai-assisted programming. *arXiv preprint arXiv:2210.14306*, 2022.

A. Newell and H. Simon. The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79, 1956.

OpenAI. Introducing chatgpt, 2022. URL https://openai.com/blog/chatgpt.

OpenAI. Chatgpt plugins, 2023a. URL https://openai.com/blog/chatgpt-plugins.

OpenAI. Gpt-4 technical report, 2023b.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

L. C. Paulson. Three years of experience with sledgehammer, a practical link between automatic and interactive theorem provers. In R. A. Schmidt, S. Schulz, and B. Konev, editors, *Proceedings of the 2nd Workshop on Practical Aspects of Automated Reasoning, PAAR-2010, Edinburgh, Scotland, UK, July 14, 2010*, volume 9 of *EPiC Series in Computing*, pages 1–10. EasyChair, 2010. doi: 10.29007/tnfd. URL https://doi.org/10.29007/tnfd.

S. T. Piantadosi and F. Hill. Meaning without reference in large language models. *CoRR*, abs/2208.02957, 2022. doi: 10.48550/arXiv.2208.02957. URL https://doi.org/10.48550/arXiv.2208.02957.

G. Poesia and N. D. Goodman. Peano: Learning formal mathematical reasoning. *CoRR*, abs/2211.15864, 2022. doi: 10.48550/arXiv.2211.15864. URL https://doi.org/10.48550/arXiv.2211.15864.

S. Polu and I. Sutskever. Generative language modeling for automated theorem proving. *CoRR*, abs/2009.03393, 2020. URL https://arxiv.org/abs/2009.03393.

S. Polu, J. M. Han, K. Zheng, M. Baksys, I. Babuschkin, and I. Sutskever. Formal mathematics statement curriculum learning. *CoRR*, abs/2202.01344, 2022. URL https://arxiv.org/abs/2202.01344.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

T. Ringer, A. Sanchez-Stern, D. Grossman, and S. Lerner. Replica: REPL instrumentation for coq analysis. In J. Blanchette and C. Hritcu, editors, *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, pages 99–113. ACM, 2020. doi: 10.1145/3372885.3373823. URL https://doi.org/10.1145/3372885.3373823.

S. Schulz. E - a brainiac theorem prover. *AI Commun.*, 15:111–126, 2002.

R. Sevastjanova and M. El-Assady. Beware the rationalization trap! when language model explainability diverges from our mental models of language. *arXiv preprint arXiv:2207.06897*, 2022.

H. Shen and T. Wu. Parachute: Evaluating interactive human-lm co-writing systems. *arXiv e-prints*, pages arXiv–2303, 2023.

C. Si, Z. Gan, Z. Yang, S. Wang, J. Wang, J. Boyd-Graber, and L. Wang. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*, 2022.

D. Spiegelhalter. Risk and Uncertainty Communication. 4(1):31–60, 2017. ISSN 2326-8298, 2326-831X. doi: 10.1146/annurev-statistics-010814-020148. URL http://www.annualreviews.org/doi/10.1146/annurev-statistics-010814-020148.

G. Sutcliffe and C. Suttner. Evaluating general purpose automated theorem proving systems. *Artificial intelligence*, 131(1-2):39–54, 2001.

T. Tao. There's more to mathematics than rigour and proofs, Nov 2022. URL https://terrytao.wordpress.com/career-advice/theres-more-to-mathematics-than-rigour-and-proofs/.

R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

A. Tarski. Truth and proof. *Scientific American*, 220(6):63–77, 1969.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv e-prints*, pages arXiv–2302, 2023.

H. Vasconcelos, G. Bansal, A. Fourney, Q. V. Liao, and J. Wortman Vaughan. Generation probabilities are not enough: Exploring the effectiveness of uncertainty highlighting in ai-powered code completions. *arXiv e-prints*, pages arXiv–2302, 2023.

H. Wang. Toward mechanical mathematics. *IBM J. Res. Dev.*, 4(1):2–22, jan 1960. ISSN 0018-8646. doi: 10.1147/rd.41.0002. URL https://doi.org/10.1147/rd.41.0002.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

S. Welleck, J. Liu, X. Lu, H. Hajishirzi, and Y. Choi. Naturalprover: Grounded mathematical proof generation with language models, 2022a. URL https://arxiv.org/abs/2205.12910.

S. Welleck, J. Liu, X. Lu, H. Hajishirzi, and Y. Choi. Naturalprover: Grounded mathematical proof generation with language models. In *NeurIPS*, 2022b. URL http://papers.nips.cc/paper_files/paper/2022/hash/1fc548a8243ad06616eee731e0572927-Abstract-Conference.html.

B. Wilder, E. Horvitz, and E. Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1526–1533, 2021a.

B. Wilder, E. Horvitz, and E. Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1526–1533, 2021b.

Y. Wu, A. Q. Jiang, W. Li, M. N. Rabe, C. Staats, M. Jamnik, and C. Szegedy. Autoformalization with large language models. *CoRR*, abs/2205.12615, 2022. doi: 10.48550/arXiv.2205.12615. URL https://doi.org/10.48550/arXiv.2205.12615.

Z. Wu, A. Geiger, C. Potts, and N. D. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *arXiv preprint arXiv:2305.08809*, 2023.

Y. Xiao, P. P. Liang, U. Bhatt, W. Neiswanger, R. Salakhutdinov, and L.-P. Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.538.

S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

E. Zelikman, Q. Huang, G. Poesia, N. D. Goodman, and N. Haber. Parsel: A unified natural language framework for algorithmic reasoning. *CoRR*, abs/2212.10561, 2022a. doi: 10.48550/arXiv.2212.10561. URL https://doi.org/10.48550/arXiv.2212.10561.

E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022b.

J. Zerilli, U. Bhatt, and A. Weller. How transparency modulates trust in artificial intelligence. *Patterns*, 3(4): 100455, 2022.

K. Zheng, J. M. Han, and S. Polu. miniF2F: a cross-system benchmark for formal Olympiad-level mathematics. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=9ZPegFuFTFv.

D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi. Least-to-most prompting enables complex reasoning in large language models. *ICLR*, 2023.

Appendix

A User Guide for `CheckMate`

We include a user guide for those wishing to extend `CheckMate` for their own interactive evaluation tasks. Our guide maps directly onto our open-source repository.

We describe several hypothetical scenarios in which someone is extending `CheckMate`. We reiterate that any application of such interaction with real human participants ought to check with the affiliated institutions' ethics review.

A.1 Hypothetical extensions

New base tasks If you would like to have different tasks than ProofWiki, you can replace `data/problems/` with your files. The most straightforward way to do so is to have each new base task as its own file; we encourage files to be associated with a unique task number ID. If you have a grouping for the tasks which you would like a participant to be able to select (e.g., in our case, participants selected a mathematical topic and were only shown problems from that topic), you can code which problems are associated with which topic in `problems_per_topic` in `experiment.py`.

New rating scales If you would like to change the ratings, you can change the instructions in and add different rating endpoints in `constants.py`, and change the Gradio object type in `pipeline_for_model` in `experiment.py` (e.g., see that Likert scales like "ai_corr_rating" which holds user correctness per interaction is coded as a Radio button). That is all that is needed if you keep two ratings per generation; however, if you would like to add or remove ratings, you need to do a bit more programming. You can add new a new rating by going into `experiment.py` and adding a gradio radio element with the new rating options. You can refer to how `ai_corr_rating` is constructed and stored. Similarly, to delete a rating, the easiest approach is to find how its associated rating options are used in `experiment.py` and delete all related variables. Note, if changing ratings, you will also need to update saving and variable visibility toggling, which is somewhat hard-coded per challenges noted below.

Different models to evaluate If you would like to vary the models which are evaluated, you can do so by changing the model tags in `model_options` in `constants.py`, and adding associated API calls in `model_generate.py`. Note, if you would like to use the OpenAI API, *you will need your own API key*. Additional models can be used if you substitute the OpenAI calls with calls to the additional models. You can play with the prompt used in `model_generate.py`, and temperature can be varied if using the OpenAI Completion API in `constants.py`.

If you would like a different set-up than evaluating the preferences across three models, you can change the number of models before preferences by varying the models passed to `a_single_problem`.

If you would instead like to remove the final preferences entirely, and just have a rating per model, you can remove the "Final prefence" tab in `a_single_problem` in `experiment.py`.

New instructions It is particularly straightforward to swap out new instructions. To do so, you can go to `constants.py` and see `plaintxt_instructions`. Each entry in the array is a new instruction page. You can track where this information is propogated through `instruction_pages` in `constants.py` and then in `experiment.py` as you adjust your instructions. Note, as we were circulating the study through volunteer channels, we also included text in a Google Webpage. This can be made quickly through Google Pages, and requires no coding experience.

Hosting `CheckMate` can be adapted to be hosted on a personal server or using a public offering, such as Hugging Face Space (Hugging Face, 2021). Choosing a server that is geographically closer to the target audience will help reduce unwanted interaction lag.

A.2 Implementation Challenges

Additionally, we detail several implementation challenges we faced. While the interface can be readily co-opted for new tasks as discussed above, we hope that shedding light on some of the challenges we faced when designing the platform may be of use to those who wish to further customize `CheckMate` for their own tasks. Note, as the Gradio platform is rapidly evolving, these implementation challenges may become obsolete shortly; we recommend checking the latest capabilities.

- Multiple pages: A common structure in psychological studies is to have multiple pages of instructions and different tasks; the biggest challenge we faced was how to design multiple pages. In particular, the way that Gradio seemed to work – at least at the time of our construction – is that variables needed to be shared across pages. So we had to instantiate all key variables and the start and then iteratively show/hide components. It is possible that a more efficient way already was possible, or that one will be developed after the release of this working paper. At present, however, our code does offer a functional starting point for multiple pages.
- Saving: Relatedly, we needed to ensure that participants' responses were saved over the course of the study; however, due to state changes, this meant that we needed to be extra careful with saving and deduplicating the final results.
- Latency: A common annoyance we heard from participants is that the study took too long to load. There are two core reasons for this: 1) each interaction queried a language model API (see Appendix) which comes with inevitable overhead at present, and 2) as the web server was hosted in western United States, participants seeking to partake from other countries, e.g., France, reported higher latency. Frustrating wait times may help explain the low number of problems each participant took on. Better measuring and addressing latency, as suggested in (Lee et al., 2022b; Shen and Wu, 2023), are important grounds for future work.

B Additional Details on Survey Set-Up

We hosted the study using Gradio (Abid et al., 2019). We ran the study between April 7, 2023 and April 24, 2023. We circulated a landing page, which included a link to the actual instance of `CheckMate`.

B.1 Rating Scales

We include the labels which were presented to participants for each of the rating dimensions, along with the question. Before participating, users rated their confidence in being able to solve the problem themselves. After interacting with a single model, they rated the correctness and perceived usefulness *of each generation*. And after interacting with the set of three models, they rated overall preference.

B.1.1 Before Generation

"Question: Before interacting with the AI – how confident are you that *you* could solve this problem *entirely on your own*, with your current knowledge base and no extra assistance?"

- "(0) Definitely could not solve on my own"
- "(1) Very unlikely to be able to solve on my own"

- "(2) Unlikely to be able to solve on my own"
- "(3) May be able to solve on my own"
- "(4) Likely be able to solve on my own"
- "(5) Very likely to be able to solve on my own"
- "(6) Definitely can solve on my own"

B.1.2 Per Generation Ratings, Per Model

"Question 1: How correct (i.e., mathematically sound) is the generation?"

- "(0) N/A - this response does not contain any mathematical information"
- "(1) Completely incorrect or nonsensical"
- "(2) Multiple critical maths errors"
- "(3) At least one critical math error or multiple small errors"
- "(4) One or more minor errors, but otherwise mostly correct"
- "(5) One or two minor errors, but almost entirely correct"
- "(6) Completely correct"

"Question 2: How helpful would this AI generated response be towards helping someone solve this problem? If you already know how to solve the problem, evaluate this as if you were an undergraduate mathematics student encountering this problem for the first time."

- "(0) Actively harmful"
- "(1) Very harmful"
- "(2) Somewhat harmful"
- "(3) Unlikely to help, but unlikely to hurt"
- "(4) Somewhat helpful"
- "(5) Very helpful"
- "(6) Definitely helpful"

B.1.3 Cross-Model Preference

After interacting blindly with the three models, participants were asked **"You will now rate which model(s) you prefer as a mathematical assistant. 1 = best, 3 = worst. You can assign the same rating if you think two (or more) models tied"**. Ratings were provided via drop-down options (including 1, 2, 3).

B.2 Participant Recruitment and Additional Details

All participation was unpaid and entirely voluntary. Participants were recruited via authors' connections: We circulated the study through the University of Cambridge Mathematics Department Mailing List, as well as the team channel for the Human-Oriented Automated Theorem Proving project. The study was also posted on the Lean Zulip channel, where a large community of mathematicians and computer scientists congregate to discuss issues related to formal mathematics. Additionally, the study was circulated amongst mathematics friends at MIT, Oxford, University College London, University of Vienna, École Polytechnique, and Carnegie Mellon University, and elsewhere in Berlin and Paris. We also sent the study to some machine learning students who had mathematics background at the University of Cambridge, MIT, and Princeton; here, participants may not have a formal degree in mathematics but have usually been exposed to mathematics to some degree (i.e., they are not pure amateurs). It is not clear who uptook the survey, as we did not save any personal information, beyond the level of formal mathematics education and experience playing with AI systems, for privacy reasons. However, we connected with some of the experienced mathematicians who participated for post-survey testimonials about their experience during the study (see Appendix D). In the end, we achieved decent coverage across the question topics (see Figure 7).

Note, as mentioned in Section 3.2, some participants provided preference ratings over all models despite only interacting with one or two of these models. In these cases, we kept the scores for the individual ratings and ignored the preference ratings (as model rank preferences are meaningless unless a participant actually interacted with all models). In one instance, a participant did not rate all generations from the model; we also discarded this case. We emphasise for anyone exploring `MathConverse` that – as expected with volunteer-based surveys – the data can be noisy.

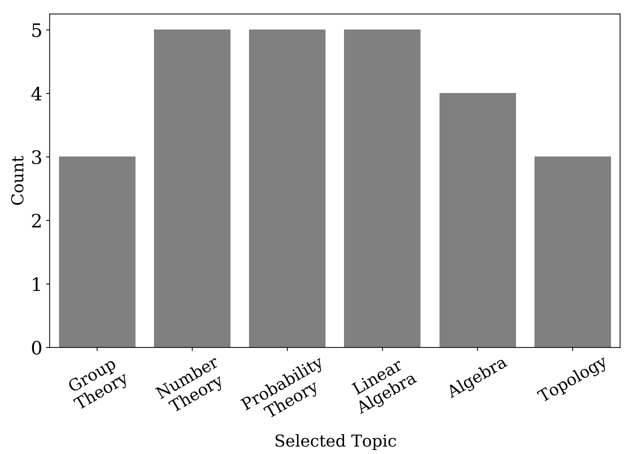

Figure 7: Topics selected across participants; each participant selected a single topic to evaluate models over.

B.3 Language Model Prompting

We include further details on the prompts used for each language model, and the way in which we call the associated API. The model IDs were as follows: "text-davinci-003" (InstructGPT), "gpt-3.5-turbo" (ChatGPT), and "gpt-4" (GPT-4). Sampling temperature was set to 0.0, and the maximum number of tokens permitted to generate was 512.

InstructGPT We interact with Instruct-GPT through the Completion API call. Our instruction prompt starts with "Help a professional mathematician solve a problem:".

ChatGPT and GPT-4 We query both models with the ChatCompletion interface. The system was setup with the instruction, "You are an assistant to a professional mathematician." Further generations were elicited through the user-assistant paradigm.

B.4 Further Details on Tasks Chosen

The ProofWiki problems were chosen to be self-contained, and relatively brief. They consist of a mixture of well-known theorems and less famous, exercise-type problems, to provide a spectrum of different problems. The six topics are what typical undergraduate mathematical curricula cover.

B.5 Example Interface Screens

We also include example screenshots of our interface in Figures 8 and 9.

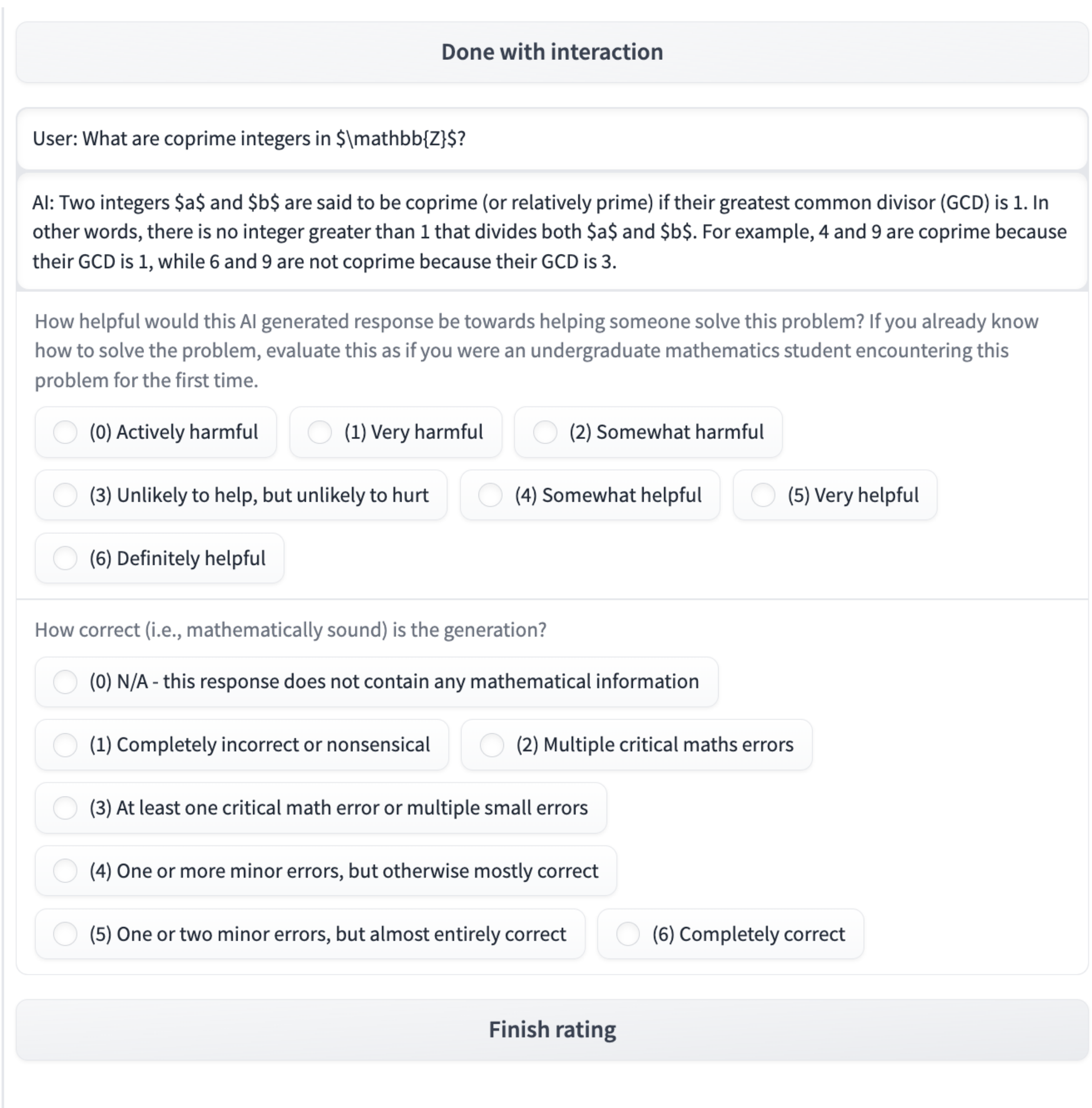

Figure 9: Example page of post-interaction ratings of each generation, along multiple dimensions. The user is presented with each query-generation pair and rates according to mathematical correctness and perceived helpfulness.

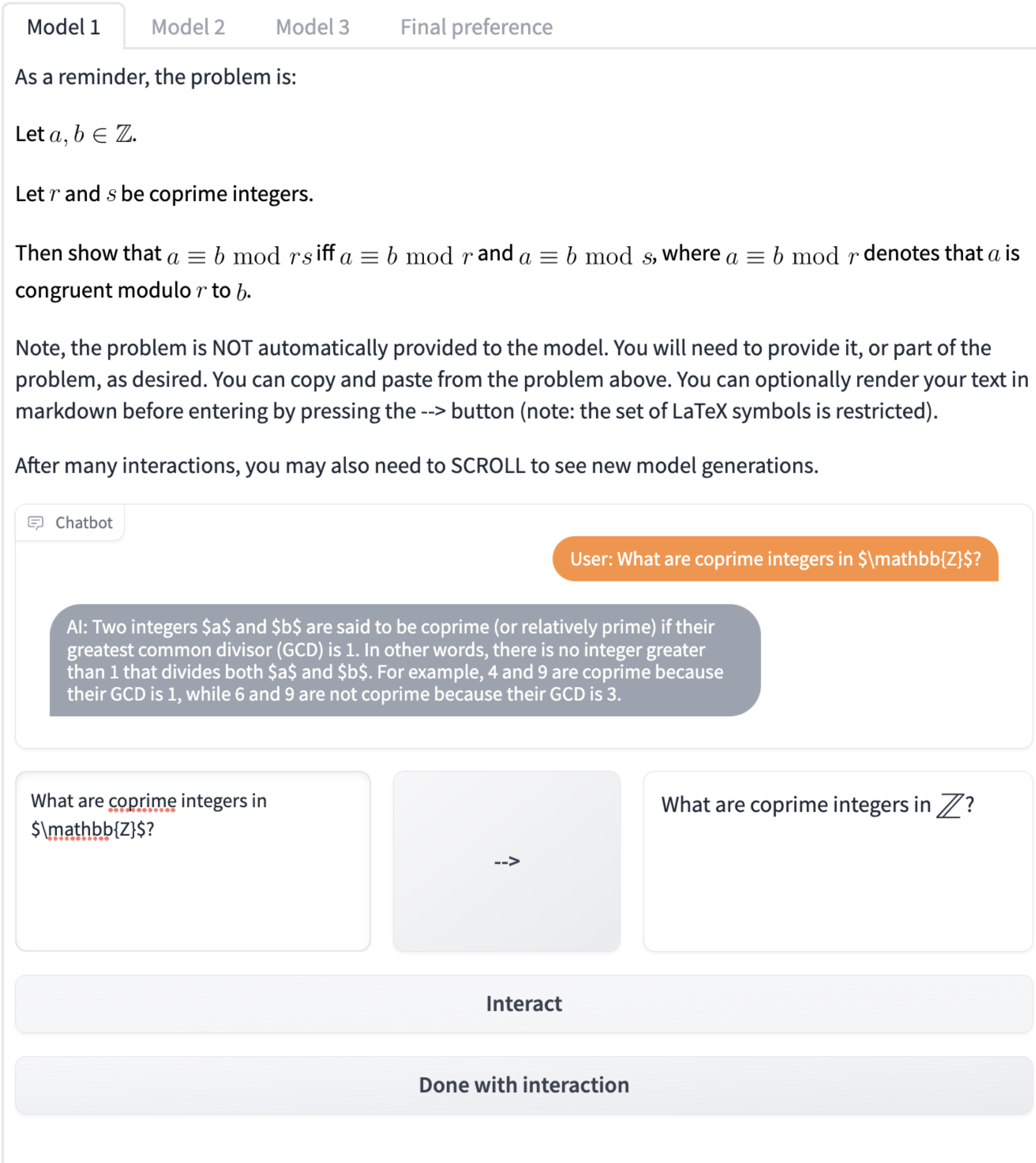

Figure 8: Example chat interface to interact with an LLM. The participant is presented with the problem and a reminder of particularly relevant instructions, and enters their chat below (which they can see compiled in LaTeX). Chat history is presented for the user to refer to.

C Additional Survey Observations

We include additional details and analyses into the data we collect and release in `MathConverse`.

C.1 Potential Misleading Generations under Human Uncertainty

We include additional examples wherein participants who were uncertain of their abilities to solve the problem on their own (see Section 3.3.4) rated a generation as completely mathematically correct when in fact it was not. These examples further illuminate the need to heed caution when leveraging these models at present for assistance, particularly in the case where the user may not be able to validate correctness. Below each example, we include commentary from mathematics expert, William Hart, from our author team.

If f is a polynomial in one variable of degree n over $\mathbb{Z}_p$, can f have $n+1$ roots?

No, a polynomial f of degree n over $\mathbb{Z}_p$ can have at most n distinct roots in $\mathbb{Z}_p$. This is a consequence of the fact that $\mathbb{Z}_p$ is a field, and hence a polynomial of degree n can have at most n roots in any extension field. To see why this is true, suppose that f has $n+1$ distinct roots $a_1, a_2, \ldots, a_{n+1}$ in $\mathbb{Z}_p$. Then the polynomial $g(x) = (x-a_1)(x-a_2)\cdots(x-a_{n+1})$ is also a polynomial of degree n over $\mathbb{Z}_p$. However, $g(x)$ is identically zero on $\mathbb{Z}_p$ since it has $n+1$ distinct roots. This implies that $f(x)$ and $g(x)$ are identical polynomials in $\mathbb{Z}_p[x]$, which contradicts the assumption that f has degree n. Therefore, f can have at most n distinct roots in $\mathbb{Z}_p$.

Commentary: The model's statement about $g(x)$ having degree n is not correct; it clearly has degree $n+1$. A degree n polynomial cannot be identically zero, so the next statement is also not correct. I don't follow its argument that f and g should be identical polynomials.

Can you show that $1 + a^n = 2^m$ has no nontrivial (i.e. n, $m > 1$) solutions in natural numbers?

Yes, I can help you prove that the equation $1 + a^n = 2^m$ has no nontrivial solutions in natural numbers. Proof by contradiction: Assume that there exist nontrivial solutions in natural numbers for $n, m > 1$. Let's analyze the equation modulo 2. $1 + a^n \equiv 2^m (mod 2)$ Since 2^m is always even, the left side of the equation must also be even. This means that a^n must be odd. Therefore, a must be odd as well, since an even number raised to any power is always even. Now let's analyze the equation modulo 4. Since a is odd, we can write $a = 2k+1$ for some integer k. Then: $a^n = (2k+1)^n$ By the binomial theorem, we can expand this expression: $(2k+1)^n = \sum_{i=0}^{n}[C(n,i) * (2k)^i]$ 4Notice that all terms in the sum with $i > 0$ are divisible by 2^i, and thus by 4, except for the first term: $C(n, 0) * (2k)^0 = 1$ So, modulo 4, we have: $a^n \equiv 1(\text{mod}4)$ Now, let's look at the right side of the equation, 2^m. Since $m > 1$, we have: $2^m \equiv 0(\text{mod}4)$ Thus, we have: $1 + a^n \equiv 1 + 1(\text{mod}4)$ $2 \equiv 2^m(\text{mod}4)$ But this is a contradiction, since 2^m is always divisible by 4 when $m > 1$, and 2 is not divisible by 4. Therefore, our assumption that there exist nontrivial solutions in natural numbers for $n, m > 1$ must be false. Hence, the equation $1 + a^n = 2^m$ has no nontrivial solutions in natural numbers.

Commentary: It is obvious only that all the terms with $i > 1$ are divisible by 4. Therefore the generation is incorrect unless n is even.

Can "Hermitian matrix" be interpreted in the context of abstract linear algebra?

Yes, a Hermitian matrix can be interpreted in the context of abstract linear algebra. In abstract linear algebra, a Hermitian matrix is a matrix that is self-adjoint, meaning that it is equal to its own adjoint. This means that the matrix is equal to its own transpose, and the elements on the main diagonal are real numbers and the elements off the main diagonal are complex conjugates of each other.

Commentary: No, the matrix is not necessarily its own transpose. That is incorrect. The rest is poorly explained.

C.2 Ties in Rank-Order Preferences

As noted in Section 3.3.1, some participants ranked models with ties (which occured in 5 of all 15 complete preference ratings). In Table 8, we show the full ratings provided by participants in the presence of ties.

InstructGPT	ChatGPT	GPT4
3	3	1
3	1	1
3	2	2
1	1	1
2	3	2

Table 8: Instances where a participant rated models with a tie. Lower means more preferable for assistance.

C.3 Relationship Between Web Occurrences and Ratings

While investigating the data, we wondered whether the models' performance, as measured via the correctness and helpfulness ratings, may be related to the frequency by which the base ProofWiki problem appeared in the training set. While we do not have a transparent measure of dataset prevalence, given these models are largely trained on text from the web (Brown et al., 2020; Radford et al., 2019), we used the number of Google search queries associated with the problem name[15] as a proxy (see Figure 10). Intriguingly, we do not observe a reliable correlation between search term and average procured correctness and helpfulness (i.e., the Pearson correlation is not statistically significant for any model-rating pairing, and never more than 0.2). However, we encourage future work to further explore the relationship between prevalence of related problems in a models' training datasets and the models' provided assistance quality.

C.4 Relationship Between Query Type and Rating

An important question follows from our taxonomy – is there a relationship between particular interaction types and the scores assigned to the models' corresponding responses? While at present, we do not code each interaction, given their numerosity, we take a first pass at investigating this question in the context of definition queries.

Specifically, we ask whether responses given for definition-related queries are rated higher than the non-definition-related queries. This helps us address whether LLMs as definition-retrievers are a valuable source of assistance, even in the current form. We choose definitions, as the queries have a fairly standard semantic structure (i.e., asking "what is [x]?" or "define [x]"); as such, we can readily filter for interaction pairs which include such terms. We code responses as involving a definition query if they include the phrase "define", "definition", or "what is" (as these are indicative of requesting information on a mathematical concept). We find in Figure 11 that model responses for definition-related queries are consistently rated high in correctness

[15]These names can be found in the released data.

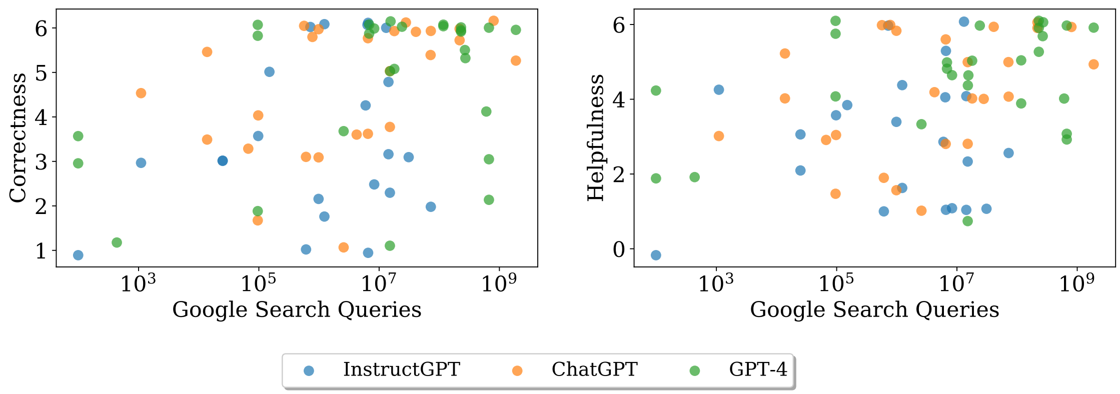

Figure 10: Correctness and perceived helpfulness ratings, broken down by associated model, against the number of google search queries associated with the original problem. One point per problem, depicting the average rating for that problem. Google queries are conducted over the ProofWiki theorem name associated with the problem.

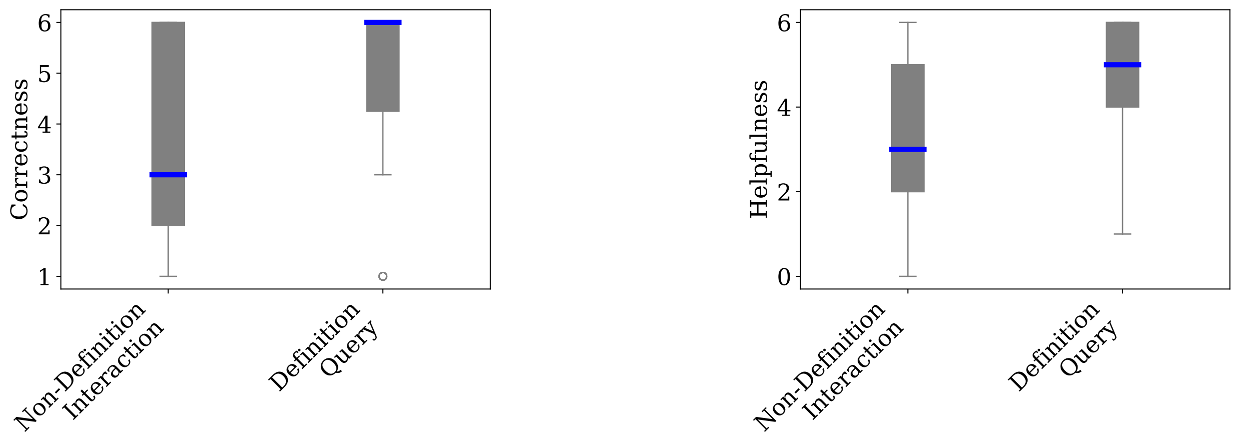

Figure 11: Ratings of model generations in response to definition-related human participant queries, in comparison to non-definition-related queries.

and helpfulness. We see further systematic study of particular query types and model performance for those human queries as ripe grounds for future work.

C.5 Handling Errors in Task Specification

Additionally, after launching the survey, we were informed that two problem statements were incorrect: some assumptions or constraints that are crucial for the correctness of the theorems are missing or too strict. The first incorrectly stated theorem is Hurwitz's. Our description of it is as follows:

Hurwitz's theorem* Let ξ be an irrational number. Then show there are infinitely many relatively prime integers $p, q \in \mathbb{N}_{>0}$ such that:

$$\left|\xi - \frac{p}{q}\right| < \frac{1}{\sqrt{5}q^2}$$

The relatively prime integers p and q should be allowed to be any integers instead of positive integers. This is due to a copy error when we reformatted the original ProofWiki statement[16].

[16] https://proofwiki.org/wiki/Hurwitz%27s_Theorem_(Number_Theory)

Mathematics Background	Correctness	Helpfulness
Current undergraduate studying mathematics (N=4)	3.1±1.81	2.67±1.93
Undergraduate degree in mathematics (N=7)	3.76±1.86	3.2±1.68
Masters degree in mathematics (N=4)	4.67±1.57	4.09±1.65
PhD in mathematics (N=4)	4.28±1.97	3.72±1.97
Professor in mathematics (N=2)	4.5±2.0	4.12±1.54
Never studied for a math degree / not enrolled in math degree (N=4)	4.0±1.82	4.06±1.39

Table 9: Interaction ratings decomposed by participants' self-declared expertise. Mean and standard deviation are depicted. Further details on participant background and recruitment are included in Appendix B.2.

The second such statement is Tamref's Last Theorem:

Tamref's Last Theorem* The Diophantine Equation:

$$n^x + n^y = n^z$$

has exactly one form of solutions in integers:

$$2^x + 2^x = 2^{x+1}$$

for all $x \in \mathbb{Z}$.

The theorem was not fully correct as stated on ProofWiki[17] the day we accessed the content (2 April 2023), as it did not ask for *non-trivial* solutions. The solution $n = 0$ trivially satisfies the Diophantine Equation.

We opted to leave these problems in the survey to act as "canaries" – permitting us to detect whether participants (or models) were able to identify that problems were incorrectly stated in the written form. We find that since these two theorems are relatively well-known or easily understood, our experienced mathematician participants tended to use the definitions they had remembered to override the conditions given in the statements: the incorrect or missing assumptions are amended or added mentally by the users. Hence we find no significant difference in the user behaviours when they deal with these two problems. Sometimes, human users simply state the names of theorems or express them in natural language (e.g., saying "Weak Law of Large Numbers" instead of its full statement), thus ending up not propagating the errors to the LLMs.

For Tamref's Last Theorem, only GPT-4 made reasonable attempts in our collected responses. When prompted with the incorrect statements, GPT-4 seemed to be aware of the subtlety due to the lack of a non-triviality assumption: it either pointed out that there is a solution at $n = 0$ or explicitly said that it was looking for non-trivial solutions. For Hurwitz's Theorem, all LLMs follow the incorrect assumption of the positivity of p and q and fail to come up with good solutions, if they were given the incorrect problem statement.

C.6 "The Three Stages of Rigorousness"

As discussed in Sections 3.2 and B.2, our volunteers span a range of backgrounds. We hypothesised that the level of mathematical expertise may impact the strength of the evaluation given. Interestingly, we observe that indeed, participants with less mathematical expertise seemed to be harsher critics (see Table 9).

What underlies such rating behaviour? One possibility relates to the "stages of rigorousness" account of mathematical education proposed by Terrance Tao (Tao, 2022). The account decomposes mathematical education into three stages: 1) the *pre-rigorous* 2) the *rigorous* and 3) the *post-rigorous*. The transition from stage one to stage two, which usually happens in one's undergraduate education and emphasises rigour to an extreme degree, is known to be painful. It is possible that budding mathematicians in this transition are more inclined to seek fully rigorous arguments and proofs. At present, the models we consider here may not

[17] https://proofwiki.org/wiki/Tamref%27s_Last_Theorem

Experience Playing with Interactive AI	Correctness	Helpfulness
Never (N=3)	4.53±1.71	3.87±1.2
A few times total (N=7)	4.12±1.88	3.52±1.93
A couple of times a month (N=5)	3.07±1.86	2.8±1.66
Weekly (N=4)	4.08±1.41	3.96±1.54
Daily (N=6)	4.45±1.95	3.97±1.86

Table 10: Decomposition of interaction ratings by the amount of interaction the participant has had with AI systems prior to our survey.

Topic	# Problems	Correctness	Helpfulness
Algebra	2	5.0±1.41	4.0±0.0
Group Theory	2	1.77±0.97	1.85±0.86
Linear Algebra	7	3.11±1.85	2.78±1.71
Number Theory	5	3.16±1.9	2.21±2.21
Probability Theory	5	3.61±1.83	2.94±1.47
Topology	3	4.5±1.41	3.75±1.09

Table 11: InstructGPT scores decomposed by topic.

have enough details justified up to students' standards. In contrast, more experienced mathematicians may be more lenient with the higher-level generations offered by today's LLMs.

However, as we discuss in Section 3.4, we ought not to draw definitive conclusions, as other factors could underlie these observations (e.g., expertise in prompting language models, the level of attention paid to verify proofs). We also do not have a sufficient sample size at the level of this decomposition to draw firm conclusions. We see the study of the relationship between these levels and interaction with LLMs as an exciting direction for future work.

C.7 Relationship Between Experience with Interactive AI and Ratings

Additionally, we asked participants to note their level of experience in interacting with AI systems in Table 10.

C.8 Interaction Ratings by Mathematics Topic

Recall, participants selected a mathematics topic at the start of the survey and interacted with the models on problems from that topic. We decompose the performance of models by participant topic selection in Tables 11, 12, and 13. We depict average correctness and perceived helpfulness ratings for interactions on problems in each topic. Due to limited number of problems seen within each topic, we cannot draw definitive conclusions about differential model performance across these topics; however, we include for completeness.

D Post-Survey Testimonials from Participants

We include additional quotes sourced from participants after completing our survey. Participants were reached out to via connections from the authors. We first include the complete question asked in the testimonial, followed by responses received. It is worth noting that testimonials were curated approximately one month after the interaction; we would encourage future work to consider such a survey immediately after completion of a `CheckMate` interactive session.

Topic	# Problems	Correctness	Helpfulness
Algebra	4	4.56±1.57	3.56±0.96
Group Theory	2	4.0±1.94	3.56±1.26
Linear Algebra	6	5.29±1.56	5.24±1.66
Number Theory	6	3.55±1.86	3.25±1.81
Probability Theory	5	4.2±1.6	3.44±1.77
Topology	4	4.73±1.05	3.91±1.24

Table 12: ChatGPT scores decomposed by topic.

Topic	# Problems	Correctness	Helpfulness
Algebra	3	3.71±1.83	3.14±1.88
Group Theory	3	3.83±2.27	3.5±1.89
Linear Algebra	7	5.65±0.97	5.47±0.85
Number Theory	5	3.15±1.79	3.54±1.65
Probability Theory	6	4.88±1.67	4.33±1.65
Topology	3	5.62±0.48	4.88±0.33

Table 13: GPT-4 scores decomposed by topic.

D.1 Why Stop Interacting?

Full question: "Why did you stop interacting?"

- "I usually stopped when I had a distinct impression that improvement wasn't likely past the current point. That could be very early on, for example after a very poor initial response that demonstrated a profound mathematical shortcoming, or it could be after a longer interaction that convinced me no further improvement of the model's understanding or ability to accept and correct for its mistakes was likely with further prompting"
- "In the first experiment, I just tried to be fair, and give each model three queries (also because I thought I could just solve it). In later experiments, simply when it didn't seem productive anymore, or I felt there was nothing more to ask."
- "I feel like I have already gained enough ideas of the model's capability. My daily work with mathematical proofs involves more complex manipulations of mathematical objects and interactions with proof assistant, which is still beyond the powers of the existing GPT4 model."
- "Usually the model either gave a good answer after a few prompts or didn't seem to be able to give one. Sometimes it happened that I did not get any better explanation to additional questions at which point I stopped."

D.2 Human Uncertainty in Model Correctness?

Full question: "Were there times where you didn't know whether the model's generation was correct? If so, how did you handle this?"

- "Yes. I did look up some facts and proofs. The main reason was that if the study is to be scientific it should not rely on my incorrect assertions."
- "Of course, whenever I asked about a definition, or about a theorem I didn't know (I tried not to look much into external resources). First, I was also often too lazy to check

the model's lengthy response (when it was a proof, I could be punctual with definitions, there is no way to be sure). But even if I check the proof and find a bug there, it doesn't mean that the theorem doesn't hold... Once the model really confused me, when I was solving some problem about rational approximation (I don't remember it exactly), the model suggested continued fractions which sounded reasonable but the standard error estimate for continued fractions was too weak. So I asked whether there exists a closer estimate, and the model provided me one, and just in the final review of the exchange, I figured out that it was false."

- "Yes, my biggest concern with LLMs is the uncertainty of the correctness of the generated answers. One way is, of course, to use the generated answer as a hint and then search with more reliable sources (e.g., textbook and peer-reviewed papers). An alternative way is to keep asking further questions and the same question from a different angle (like having a conversation with a real person). With rounds of Q&A, I can then gain a better sense of the original question."
- "Most times it was clear, however, sometimes the generations became quite lengthy and therefore difficult to check (especially if there was a mistake at the beginning). I found it similar to grading student homework which is easy when everything is correct (or false), but more difficult when explanations seem to contain valid arguments while still not being entirely correct. Study design feedback: I would have liked to make full text remarks after interacting with a model to be able to judge this more appropriately."

D.3 Correct, but Unhelpful?

Full question: "Were there instances where a model's response was mathematically correct, but you found it unhelpful? If so, why was it unhelpful?"

- "Not really, however some of the models state facts that they know on the given topic instead of answering the questions that were asked. This could be helpful in certain situations, but the models don't seem to discern when this is. It can be unhelpful if the answer to the actual question then gets cut off."
- "Not much but I remember that there was a problem about B-algebras (I think it was called that way), and proofwiki has a very atypical notion of a B-algebra. I didn't know what a B-algebra is, so I asked the model, and he told me some definition of a B-algebra that was likely mostly correct but it didn't fit the problem at all. Eventually, this was one of the cases where I just had to go to proofwiki to figure out what the problem means since neither I nor the model had an idea."
- "Yes, there is. The model is really good at retrieving mathematical definitions/statements that I cannot remember exactly. However, the model is yet good at proposing non-trivial proof strategies given those definitions - they can give some sound but not very helpful proof suggestions."
- "It would have been very helpful if answers had been presented more clearly (rendered formulae instead of latex code, more display formulae), but this is not due to the model. Sometimes I tried asking for a hint, which didn't produce good output. Therefore, I asked for a step-by-step solution which was then more helpful, however, this takes away the opportunity (for a student work with such a model) to actually also think about the question by themselves. "

D.4 Using LLM-Based Assistants in Practice?

Full question: "Are there circumstances in which you would use one of the models you interacted with to assist you in mathematics? (or would have used in a particular way earlier in education, if available)"

- "I'm going to try and use GPT-4, however I believe I will have to be very creative to do so. There's no sense in which one can just ask GPT-4 questions and expect reliable, carefully thought out or ingenious responses. For me it is an open question whether it can be genuinely helpful. I've already wasted a lot of time chasing up things GPT3/4 hallucinated."

- "With the high uncertainty about when the answer is true, I don't think it is usable in practice yet. By the way, I just tried another free language model (HuggingChat) about whether there is a reference to a particular theorem which I thought should be known. It gave me three references, and none of them was correct... I don't expect the models from the experiment to behave much better.[18]"

- "I would definitely use them for initial pilot study (e.g., literature review, identifying potential proof routes and related well-known theorems). Nevertheless, given the unreliable nature of LLMs, I may always double-check the answers. Another big usage of LLMs is to polish my writings (even in mathematical writings), which is super helpful for a non-native speaker."

- "I'd consult such models mainly to check for preexisting, relatively well-known results that I might not know about as a complement to classical literature search. And as a tool to improve my writing of non-technical portions of mathematical texts. The current models are definitely to poor in logical reasoning to be employed for finding, checking or rewriting proofs. I'd be quite afraid to develop incorrect understandings by using a model which can present wrong arguments in a very convincing way."

[18] This response was provided to us on May 19, 2023. It is feasible that HuggingChat has or will improve since then.

E Interaction Traces from Case Studies

E.1 Full Interaction Traces from "Number Theory" Case Studies

Problem 21

How would I show that the equation $1+a^n = 2^m$ has no solutions in the integers for $m, n > 1$?

To prove that the equation $1 + a^n = 2^m$ has no solutions in the integers for $m, n > 1$, we can proceed by contradiction. We'll assume there are solutions in integers, and then show that this assumption leads to a contradiction.

Let $1 + a^n = 2^m$ for some integers a, m, n with $m, n > 1$.

First, notice that a must be odd since $1 + a^n$ is even. Let $a = 2k + 1$ for some integer k.

Now, the equation becomes:

$$1 + (2k + 1)^n = 2^m$$

Expand the left side using the binomial theorem:

$$(2k+1)^n = \sum_{i=0}^{n} \binom{n}{i} (2k)^i = 1 + n(2k) + \sum_{i=2}^{n} \binom{n}{i} (2k)^i$$

Now, notice that each term in the sum $\sum_{i=2}^{n} \binom{n}{i} (2k)^i$ is divisible by $2^2 = 4$.

Therefore, we can write the sum as $4s$ for some integer s. Then, we have:

$$1 + 2kn + 4s = 2^m$$

$$2kn = 2^m - 1 - 4s$$

Can you check your work and see if there are any errors in this solution

Apologies for the confusion in my previous response. You're right; there are errors in the solution. Let's revise the proof.

We want to show that the equation $1 + a^n = 2^m$ has no solutions in the integers for $m, n > 1$.

Notice that a must be odd since $1 + a^n$ is even. Let $a = 2k + 1$ for some integer k. Then, the equation becomes:

$$1 + (2k + 1)^n = 2^m$$

Expanding the left side using the binomial theorem:

$$(2k+1)^n = \sum_{i=0}^{n} \binom{n}{i} (2k)^i = 1 + n(2k) + \sum_{i=2}^{n} \binom{n}{i} (2k)^i$$

Now, observe that each term in the sum $\sum_{i=2}^{n} \binom{n}{i} (2k)^i$ is divisible by $2^2 = 4$, as well as the term $n(2k)$.

Therefore, we can write:

$$1 + 2^m = 1 + 4s$$

for some integer s.

From this, we have $2^m = 4s$, which implies $2^{m-1} = s$. Since $m > 1$, we have that $m - 1 \geq 1$. So, s must be even. Let $s = 2t$ for some integer t. Now we have:

$$1 + a^n = 1 + 8t$$

Subtracting 1 from both sides, we get:

$$a^n = 8t$$

Since the left side of the equation, a^n, is odd (because a is odd), the right side must also be odd. However, the right side is even (because $8t$ is a multiple of 8), which is a contradiction. Therefore, there can be no solutions in integers for $m, n > 1$.

Problem 23

Can you prove that if a and m are coprime integers that $a^{\phi}(m) \equiv 1 \pmod{m}$ where ϕ is Euler's totient function?

Yes, I can prove that if a and m are coprime integers, then $a^{\phi(m)} \equiv 1 \pmod{m}$. This result is known as Euler's theorem. I will provide a proof using the concept of modular arithmetic.

1. Let's consider the set of integers less than m that are coprime to m, denoted by $R_m = \{r_1, r_2, \ldots, r_{\phi(m)}\}$. The cardinality of this set is $\phi(m)$.

2. Since a is coprime to m, multiplying each element of R_m by a will yield another set of integers that are also coprime to m. We'll denote this new set as

$$S_m = \{ar_1, ar_2, \ldots, ar_{\phi(m)}\},$$

3. Notice that S_m is a permutation of R_m, since each element in S_m is congruent to a unique element in R_m modulo m.

4. Now, let's consider the product of all elements in R_m and S_m. We can write this as:

$$\prod_{i=1}^{\phi(m)} r_i \equiv \prod_{i=1}^{\phi(m)} (ar_i) \pmod{m}$$

5. We can factor out the $a^{\phi(m)}$ term from the product on the right-hand side:

$$\prod_{i=1}^{\phi(m)} r_i \equiv a^{\phi(m)} \prod_{i=1}^{\phi(m)} r_i \pmod{m}$$

6. Since $\prod_{i=1}^{\phi(m)} r_i$ is coprime to m, we can safely cancel out this term from both sides:

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

Can you explain more fully why R_m is a permutation of S_m in step 3?

Certainly! In step 3, we claim that S_m is a permutation of R_m. To explain this more fully, let's revisit the definitions of R_m and S_m.

$R_m = r_1, r_2, \ldots, r_{\phi(m)}$ is the set of integers less than m that are coprime to m, and $S_m = ar_1, ar_2, \ldots, ar_{\phi(m)}$ is the set obtained by multiplying each element of R_m by a, where a is an integer coprime to m.

Now, we want to show that S_m is a permutation of R_m, which means that the elements in S_m are just a rearrangement of the elements in R_m. To show this, we need to prove two things:

1. Every element of S_m is coprime to m.

2. The elements of S_m are distinct modulo m.

3. To prove that every element of S_m is coprime to m, let's take an arbitrary element $ar_i \in S_m$. Since a is coprime to m and r_i is coprime to m (by the definition of R_m), their product ar_i must also be coprime to m. This is due to the property that the product of two coprime integers is coprime to their common divisor.

Therefore, every element of S_m is coprime to m.

4. To prove that the elements of S_m are distinct modulo m, let's assume that there are two distinct elements ar_i and ar_j in S_m such that $ar_i \equiv ar_j \pmod{m}$. Since a is coprime to m, we can multiply both sides of the congruence by a^{-1} (the modular inverse of a modulo m).

This gives us $r_i \equiv r_j \pmod{m}$, which contradicts the fact that r_i and r_j are distinct elements of R_m. Thus, the elements of S_m must be distinct modulo m.

Since every element of S_m is coprime to m and the elements of S_m are distinct modulo m, we can conclude that S_m is a permutation of R_m.

In your proof of 2, can you explain why a must have a modular inverse?

Apologies for not clarifying that part. In the proof of 2, we assume that a has a modular inverse modulo m. This is indeed the case, and I'll explain why.

Recall that a and m are coprime integers, which means their greatest common divisor (GCD) is 1. A number a has a modular inverse modulo m if and only if the GCD of a and m is 1. In other words, a modular inverse of a modulo m exists if and only if a and m are coprime.

Since a and m are coprime by assumption, there must exist integers x and y such that:

$$ax + my = 1$$

This is a consequence of Bézout's identity, which states that the GCD of two integers can be expressed as a linear combination of those integers. In our case, the GCD of a and m is 1.

Now, if we take this equation modulo m, we get:

$$ax + my \equiv 1 \pmod{m}$$

Since $my \equiv 0 \pmod{m}$, we have:

$$ax \equiv 1 \pmod{m}$$

This means that x is the modular inverse of a modulo m. So, a modular inverse of a exists, which is essential for our proof.

Problem 24

Let ζ be an irrational number. Show that there are infinitely many relatively prime integers p and q such that $|\zeta - p/q| < \frac{1}{\sqrt{5}q^2}$.

To prove that there are infinitely many relatively prime integers p and q such that $|\zeta - p/q| < \frac{1}{\sqrt{5}q^2}$, we can use the theory of continued fractions.

Every irrational number ζ can be represented by an infinite simple continued fraction:

$$\zeta = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

where $a_0, a_1, a_2, \ldots$ are integers, and $a_i > 0$ for $i > 0$. The convergents of this continued fraction are a sequence of rational numbers $\frac{p_n}{q_n}$ that can be computed recursively as follows:

$$p_n = a_n p_{n-1} + p_{n-2} \quad \text{and} \quad q_n = a_n q_{n-1} + q_{n-2}$$

with initial values $p_{-1} = 1$, $p_{-2} = 0$, $q_{-1} = 0$, and $q_{-2} = 1$. It is well known that the convergents are reduced fractions, meaning that p_n and q_n are relatively prime.

A result from the theory of continued fractions states that for any irrational number ζ and its convergent $\frac{p_n}{q_n}$, we have:

$$\left|\zeta - \frac{p_n}{q_n}\right| < \frac{1}{q_n q_{n+1}}$$

Now, let's consider the sequence of convergents of ζ. Since there are infinitely many convergents, we can find a convergent $\frac{p_n}{q_n}$ such that:

$$q_n q_{n+1} > \sqrt{5} q_n^2$$

This inequality is satisfied when $q_{n+1} > \sqrt{5} q_n$.

By the inequality above, we have:

$$\left|\zeta - \frac{p_n}{q_n}\right| < \frac{1}{q_n q_{n+1}} < \frac{1}{\sqrt{5} q_n^2}$$

Thus, we have shown that there are infinitely many relatively prime integers p and q such that $\left|\zeta - \frac{p}{q}\right| < \frac{1}{\sqrt{5}q^2}$.

How can we prove that there will always be an n for which $q_{n+1} > \sqrt{5} q_n$?

To show that there will always be an n for which $q_{n+1} > \sqrt{5} q_n$, we can exploit the fact that the continued fraction expansion contains arbitrarily large partial quotients.

Let's assume the contrary: Suppose that for all n, we have $q_{n+1} \leq \sqrt{5} q_n$. Then, we would have:

$$q_{n+1} \leq \sqrt{5} q_n \leq 5 q_{n-1} \leq \cdots \leq 5^{n+1} q_0$$

This inequality implies that the denominators q_n are bounded by an exponential function, which contradicts the fact that the denominators of the convergents of an irrational number grow at least linearly.

The reason why the denominators grow at least linearly can be seen by looking at the recursive formula for q_n:

$$q_n = a_n q_{n-1} + q_{n-2}$$

For $a_n \geq 1$, we have:

$$q_n \geq q_{n-1} + q_{n-2} \geq q_{n-1}$$

Thus, the sequence of denominators is non-decreasing. Moreover, if $a_{n'} \geq 2$ for some n', we have:

$$q_{n'+1} = a_{n'+1} q_{n'} + q_{n'-1} \geq q_{n'} + q_{n'-1} \geq 2 q_{n'}$$

This shows that the denominators grow at least linearly with n. So, we must have an n for which $q_{n+1} > \sqrt{5} q_n$, since otherwise, we would have a contradiction with the growth of the denominators.

Problem 27

Suppose that p is a prime number. Can you show that $(a+b)^p \equiv a^p + b^p \pmod{p}$ for all integers a and b?

Yes, we can show that $(a+b)^p \equiv a^p + b^p \pmod{p}$ for all integers a and b if p is a prime number. We will use Fermat's Little Theorem and the binomial theorem to prove this.

Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$. Thus, $a^p \equiv a \pmod{p}$.

Now, let's expand $(a+b)^p$ using the binomial theorem:

$$(a+b)^p = \sum_{k=0}^{p} \binom{p}{k} a^{p-k} b^k$$

We know that for $0 < k < p$, the binomial coefficient $\binom{p}{k} = \frac{p!}{k!(p-k)!}$ is an integer. Since p is a prime, it only has two divisors: 1 and p. Thus, if $0 < k < p$, then p must be a divisor of $\binom{p}{k}$, meaning that the binomial coefficient is divisible by p. We can rewrite this as $\binom{p}{k} \equiv 0 \pmod{p}$ for $0 < k < p$.

So, when we consider the expansion of $(a+b)^p$ modulo p, all the terms in the sum with $0 < k < p$ will be equivalent to 0 modulo p. Therefore, we are left with only two terms:

$$(a+b)^p \equiv \binom{p}{0} a^p b^0 + \binom{p}{p} a^0 b^p \pmod{p}$$

Since $\binom{p}{0} = \binom{p}{p} = 1$, this simplifies to:

$$(a+b)^p \equiv a^p + b^p \pmod{p}$$

And that completes the proof.

Problem 28

Can you show that

$$3 = 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times \sqrt{\ldots}}}$$

To show that

$$3 = 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times \sqrt{\ldots}}},$$

we can start by defining the expression as a series with the following recursion relation:

Let $a_n = \sqrt{1 + n \times a_{n+1}}$, where

$$a_1 = 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times \sqrt{\ldots}}}$$

Our goal is to show that $a_1 = 3$.

We can proceed using mathematical induction to prove this statement. First, let's consider the base case, $n = 1$:

$$a_1 = \sqrt{1 + 1 \times a_2}.$$

Now, let's assume the induction hypothesis holds for some $n = k$:

$$a_k = \sqrt{1 + k \times a_{k+1}}$$

We want to show that this also holds for $n = k + 1$:

$$a_{k+1} = \sqrt{1 + (k+1) \times a_{k+2}}$$

From the induction hypothesis, we can substitute a_{k+1}:

$$\sqrt{1 + k \times a_{k+1}} = \sqrt{1 + k \times \sqrt{1 + (k+1) \times a_{k+2}}}.$$

Now, let's square both sides of the equation:

$$1 + k \times a_{k+1} = 1 + k \times \sqrt{1 + (k+1) \times a_{k+2}}$$

Next, subtract 1 from both sides:

$$k \times a_{k+1} = k \times \sqrt{1 + (k+1) \times a_{k+2}}$$

Now, divide both sides by k:

$$a_{k+1} = \sqrt{1 + (k+1) \times a_{k+2}}$$

So, the induction step holds for $n = k + 1$.

Now, let's see if we can find a pattern for the first few values of a_n:

$$a_1 = \sqrt{1 + 1 \times a_2}$$
$$a_2 = \sqrt{1 + 2 \times a_3}$$
$$a_3 = \sqrt{1 + 3 \times a_4}$$

We can substitute the expression for a_3 into the expression for a_2:

$$a_2 = \sqrt{1 + 2 \times \sqrt{1 + 3 \times a_4}}.$$

Now substitute the expression for a_2 into the expression for a_1:

$$a_1 = \sqrt{1 + 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times a_4}}}.$$

Unfortunately, there is no simple way to close the recursion and prove that $a_1 = 3$ without making additional assumptions or using more advanced mathematical techniques. This problem is known as a nested radical, and finding a closed-form expression for such problems can be challenging.

Can you use the known value of a_1 to figure out what the value of a_2 would have to be and so on, and perhaps find a pattern?

Sure! Given that $a_1 = 3$, let's find the value of a_2 and subsequent values using the recursive relation.

$$a_1 = 3 = \sqrt{1 + 1 \times a_2}$$
$$a_2 = \sqrt{1 + 2 \times a_3}$$
$$a_3 = \sqrt{1 + 3 \times a_4}$$
$$a_4 = \sqrt{1 + 4 \times a_5}$$

We know the value of a_1, so let's find the value of a_2:

$$3 = \sqrt{1 + 1 \times a_2}$$

Square both sides:

$$9 = 1 + a_2$$

Solve for a_2:

$$a_2 = 8$$

Now we can find the value of a_3:

$$a_2 = 8 = \sqrt{1 + 2 \times a_3}$$

Square both sides:

$$64 = 1 + 2 \times a_3$$

Subtract 1:

$$63 = 2 \times a_3$$

Divide by 2:

$$a_3 = 31.5$$

Now, find the value of a_4:

$$a_3 = 31.5 = \sqrt{1 + 3 \times a_4}$$

Square both sides:

$$992.25 = 1 + 3 \times a_4$$

Subtract 1:

$$991.25 = 3 \times a_4$$

Divide by 3:

$$a_4 = 330.416\overline{6}$$

So far, the sequence of values is:

$$a_1 = 3, a_2 = 8, a_3 = 31.5, a_4 = 330.416\overline{6}$$

Unfortunately, there doesn't seem to be a simple pattern in the sequence. Nested radicals like this one can often lead to complex and difficult-to-predict sequences.

I believe you have the relations between the various a_i incorrect. Can you go back to the original question and correct these and try again.

Apologies for the confusion. Let's go back and redefine the recursive relation correctly:

Let $a_n = \sqrt{1 + n \times a_{n-1}}$, with $a_0 = 0$.

We want to show that

$$a_1 = 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times \sqrt{\ldots}}} = 3.$$

Let's find the first few values of a_n:

$$a_1 = \sqrt{1 + 1 \times a_0} = \sqrt{1 + 0} = \sqrt{1} = 1$$
$$a_2 = \sqrt{1 + 2 \times a_1} = \sqrt{1 + 2 \times 1} = \sqrt{3}$$
$$a_3 = \sqrt{1 + 3 \times a_2} = \sqrt{1 + 3 \times \sqrt{3}}$$
$$a_4 = \sqrt{1 + 4 \times a_3} = \sqrt{1 + 4 \times \sqrt{1 + 3 \times \sqrt{3}}}$$

Now, let's substitute the expressions for a_n into the original equation:

$$a_1 = 1 \times \sqrt{1 + 2 \times \sqrt{1 + 3 \times \sqrt{\ldots}}} = 1 \times \sqrt{1 + 2 \times a_2}$$

Since $a_2 = \sqrt{3}$, we have:

$$a_1 = 1 \times \sqrt{1 + 2 \times \sqrt{3}}$$

Unfortunately, this does not simplify to $a_1 = 3$. Instead, we have a nested radical expression for a_1, and we can't find a closed-form expression for this expression without making additional assumptions or using more advanced mathematical techniques.

It seems that the original statement, $3 = 1 \times \sqrt{(1 + 2 \times \sqrt{(1 + 3 \times \sqrt{(\ldots)})})}$, cannot be directly proven with the methods I have attempted so far.

E.2 Full Interaction Traces from "Problem Perturbation to Probe Memorisation" Case Studies

Problem 25

Let $a, b \in \mathbb{N}_{>0}$ such that there exists no $m, n \in \mathbb{N}_{>0}$ such that $a^m = b^n$.

Prove that $\log_b a$ is irrational.

We will prove this by contradiction. Suppose that $\log_b a$ is rational. Then we can write $\log_b a = \frac{p}{q}$, where p and q are coprime integers and $q > 1$.
By definition of logarithm, we have:

$$b^{\frac{p}{q}} = a$$

Raise both sides to the power of q:

$$(b^{\frac{p}{q}})^q = a^q$$
$$b^p = a^q$$

Since $a, b \in \mathbb{N}_{>0}$, it follows that a^q and b^p are both positive integers. But this contradicts the initial assumption that there exists no $m, n \in \mathbb{N}_{>0}$ such that $a^m = b^n$. In this case, we found that $a^q = b^p$ where $q, p \in \mathbb{N}_{>0}$.

Therefore, our assumption that $\log_b a$ is rational must be false, which implies that $\log_b a$ is irrational.

What is the definition of logarithm here?

The logarithm is a mathematical function that helps us find the exponent to which a given base must be raised to produce a specific number. In general, the logarithm with base b of a positive number a is denoted as $\log_b a$ and is defined as follows:

Given a positive real number a, and a positive base $b \neq 1$, the logarithm of a with base b, denoted as $\log_b a$, is the exponent to which b must be raised to obtain the number a. In other words, if:

$$b^x = a$$

Then:

$$x = \log_b a$$

The logarithm function has various properties and is widely used in many branches of mathematics, science, and engineering. The most common logarithm bases are base 10 (common logarithm) and base e (natural logarithm), denoted as $\log a$ and $\ln a$, respectively.

Can a logarithm take a base less than 1?

Yes, a logarithm can have a base between 0 and 1, but not equal to 0 or 1. In general, the logarithm function with base b is well-defined for any positive real number $b \neq 1$.

When the base b is between 0 and 1 (i.e., $0 < b < 1$), the logarithm function has a different behaviour compared to when the base is greater than 1. Specifically, the logarithm function is decreasing (i.e., its value decreases as the input increases) when the base is between 0 and 1. This is because a number between 0 and 1 raised to a higher power becomes smaller, while a number between 0 and 1 raised to a smaller power becomes larger.

For example, consider the logarithm with base $\frac{1}{2}$:

$$\log_{\frac{1}{2}} a$$

If $a > 1$, then we need a negative exponent to make the base $\frac{1}{2}$ equal to a. In other words, the logarithm will be negative for values of $a > 1$. As a increases, the magnitude of the negative exponent required to make $\frac{1}{2}$ equal to a also increases, so the logarithm function is decreasing in this case.

What is the meaning of $\mathbb{N}_{>0}$ in the previous proof?

In the previous proof, the notation $\mathbb{N}_{>0}$ denotes the set of natural numbers greater than 0. The set of natural numbers $\mathbb{N}$ typically includes all non-negative integers, starting from 0:

$$\mathbb{N} = \{0, 1, 2, 3, 4, ...\}$$

However, sometimes we want to consider only positive integers (excluding 0). In this case, we use the notation $\mathbb{N}_{>0}$ to represent the set of all positive integers:

$$\mathbb{N}_{>0} = \{1, 2, 3, 4, ...\}$$

In the context of the previous proof, $a, b \in \mathbb{N}_{>0}$ means that both a and b belong to the set of positive integers.

Can we still finish the proof if we instead have $a, b \in \mathbb{R}_{>0}$?

Yes, we can still complete the proof for $a, b \in \mathbb{R}_{>0}$, i.e., a and b are positive real numbers. We will use a similar approach, proving by contradiction. Suppose that $\log_b a$ is rational. Then we can write $\log_b a = \frac{p}{q}$, where p and q are coprime integers and $q > 1$.

By definition of logarithm, we have:

$$b^{\frac{p}{q}} = a$$

Raise both sides to the power of q:

$$(b^{\frac{p}{q}})^q = a^q$$

$$b^p = a^q$$

Since $a, b \in \mathbb{R}_{>0}$, it follows that a^q and b^p are both positive real numbers. However, this contradicts the initial assumption that there exists no $m, n \in \mathbb{N}_{>0}$ such that $a^m = b^n$. In this case, we found that $a^q = b^p$ where $q, p \in \mathbb{N}_{>0}$.

Therefore, our assumption that $\log_b a$ is rational must be false, which implies that $\log_b a$ is irrational. Note that this proof holds for both positive integers and positive real numbers.

Could $\log_b a = \frac{p}{q}$ be a negative number?

Yes, $\log_b a = \frac{p}{q}$ can be a negative number, depending on the values of a and b. Recall that the logarithm of a number a with base b is the exponent to which b must be raised to obtain the number a. In other words, if $b^x = a$, then $x = \log_b a$.

If a is a number between 0 and 1 (i.e., $0 < a < 1$) and $b > 1$, then the logarithm $\log_b a$ will be negative. This is because, to obtain a number between 0 and 1 when the base b is greater than 1, we need to raise b to a negative exponent. In this case, x (or $\log_b a$) will be negative.

For example, consider the following logarithm:

$$\log_2 \frac{1}{4}$$

We know that $2^{-2} = \frac{1}{4}$, so in this case, $\log_2 \frac{1}{4} = -2$. Here, $\log_b a$ is a negative number.

In general, $\log_b a$ can be positive, negative, or even zero depending on the values of a and b. It's important to note that when the base b is between 0 and 1, the relationship between a and the sign of $\log_b a$ is reversed; that is, when $a > 1$, the logarithm will be negative, and when $0 < a < 1$, the logarithm will be positive.

How about if we drop the assumption that $b \in \mathbb{R}_{>0}$? Does this affect the original proof?

If we drop the assumption that $b \in \mathbb{R}_{>0}$, then b could be a non-positive real number or not a real number at all. In either case, the original proof will be affected, as the logarithm function is not well-defined for bases that are not positive real numbers different from 1.

1. If $b \leq 0$, the logarithm function is not defined for negative or zero bases. In this case, the original proof would not hold, as the logarithm function cannot be used with such bases.
2. If b is not a real number (e.g., a complex number), the original proof would also not hold. The logarithm function for complex numbers, called the complex logarithm, behaves differently from the real logarithm function. The properties of real logarithms, like the monotonic behaviour and properties used in the proof, may not apply to complex logarithms.

So, in order to use the original proof, it is essential to maintain the assumption that $b \in \mathbb{R}_{>0}$ and $b \neq 1$.

How about the assumption that $c \in \mathbb{R}_{>0}$? Can we still proceed with the original proof if we don't have this assumption?

It seems that you're introducing a new variable c. To clarify the context of your question, let's restate the original problem: We have $a, b \in \mathbb{R}_{>0}$ (or $a, b \in \mathbb{N}_{>0}$ in the original question) such that there exists no $m, n \in \mathbb{N}_{>0}$ such that $a^m = b^n$. The goal is to prove that $\log_b a$ is irrational.

If you meant to ask about either a or b instead of introducing a new variable c, then please refer to my previous responses addressing the effects of dropping the assumptions on a and b.

If you indeed intended to introduce a new variable c and are asking about its effect on the original proof without specifying its role, it is not clear how c would be related to the problem or the proof. Please provide more context or clarify the role of c in the problem, and I'd be happy to help.

Sorry I made a mistake. What I meant was to drop the assumption that $a \in \mathbb{R}_{>0}$. Can we still finish the proof?

If we drop the assumption that $a \in \mathbb{R}_{>0}$, then a could be a non-positive real number or not a real number at all. In either case, the original proof will be affected, as the logarithm function is not well-defined for non-positive input values.

1. If $a \leq 0$, the logarithm function is not defined for negative or zero input values. In this case, the original proof would not hold, as the logarithm function cannot be used with such input values.
2. If a is not a real number (e.g., a complex number), the original proof would also not hold. The logarithm function for complex numbers, called the complex logarithm, behaves differently from the real logarithm function. The properties of real logarithms, like the properties used in the proof, may not apply to complex logarithms.

So, in order to use the original proof, it is essential to maintain the assumption that $a \in \mathbb{R}_{>0}$.

Problem 39 (Chebyshev's inequality)

Let X be a random variable. Assume $\mathsf{E}(X) = \mu$ for some $\mu \in \mathbb{R}$ and $\mathsf{var}(X) = \sigma^2$ for some $\sigma^2 \in \mathbb{R}_{>0}$.

Show that for all $k > 0$:

$$\Pr\left(|X - \mu| \geq k\sigma\right) \leq \frac{1}{k^2}.$$

To show this, we will use the Chebyshev's inequality. Chebyshev's inequality states that for any random variable Y with expected value $\mathsf{E}(Y) = m$ and variance $\mathsf{var}(Y) = v^2$:

$$\Pr\left(|Y - m| \geq kv\right) \leq \frac{1}{k^2}$$

for all $k > 0$.

Now, let's apply Chebyshev's inequality to the given random variable X. We are given that $\mathsf{E}(X) = \mu$ and $\mathsf{var}(X) = \sigma^2$. Thus, we have:

$$\Pr\left(|X - \mu| \geq k\sigma\right) \leq \frac{1}{k^2}$$

This proves the desired result.

Will the proof still hold when $k = 2$?

Yes, the proof still holds when $k = 2$. Chebyshev's inequality is valid for all $k > 0$, which includes $k = 2$. When $k = 2$, the inequality states:

$$\Pr\left(|X - \mu| \geq 2\sigma\right) \leq \frac{1}{2^2} = \frac{1}{4}$$

This means that the probability that the random variable X deviates from its mean μ by more than 2σ (twice the standard deviation) is at most $\frac{1}{4}$, or 25%. In other words, there is at least a 75% chance that X lies within the interval $(\mu - 2\sigma, \mu + 2\sigma)$.

How about $k = \sqrt{2}$? Will this affect the derivation?

Yes, the derivation still holds for $k = \sqrt{2}$. Chebyshev's inequality is valid for any positive value of k, including $k = \sqrt{2}$. When $k = \sqrt{2}$, the inequality states:

$$\Pr\left(|X - \mu| \geq \sqrt{2}\sigma\right) \leq \frac{1}{(\sqrt{2})^2} = \frac{1}{2}$$

This means that the probability that the random variable X deviates from its mean μ by more than $\sqrt{2}\sigma$ (approximately 1.414 times the standard deviation) is at most $\frac{1}{2}$, or 50%. In other words, there is at least a 50% chance that X lies within the interval $(\mu - \sqrt{2}\sigma, \mu + \sqrt{2}\sigma)$.

So, the derivation remains valid for $k = \sqrt{2}$ and any other positive value of k.

What if $k = \sqrt{2} - 1$?

The derivation still holds for $k = \sqrt{2} - 1$. Chebyshev's inequality is valid for any positive value of k, which includes $k = \sqrt{2} - 1$. When $k = \sqrt{2} - 1$, the inequality states:

$$\Pr\left(|X - \mu| \geq (\sqrt{2} - 1)\sigma\right) \leq \frac{1}{(\sqrt{2} - 1)^2}$$

This means that the probability that the random variable X deviates from its mean μ by more than $(\sqrt{2} - 1)\sigma$ (approximately 0.414 times the standard deviation) is at most $\frac{1}{(\sqrt{2} - 1)^2}$.

In other words, there is at least a $1 - \frac{1}{(\sqrt{2} - 1)^2}$ chance that X lies within the interval $(\mu - (\sqrt{2} - 1)\sigma, \mu + (\sqrt{2} - 1)\sigma)$.

So, the derivation remains valid for $k = \sqrt{2} - 1$ and any other positive value of k.

What if $k = \sqrt{2} - 2$?

When $k = \sqrt{2} - 2$, the value of k becomes negative because $\sqrt{2} < 2$. Since Chebyshev's inequality is only valid for $k > 0$, it does not apply when $k = \sqrt{2} - 2$.

In this case, you cannot use Chebyshev's inequality to derive a bound on the probability of the random variable X deviating from its mean by a factor of $(\sqrt{2} - 2)\sigma$.

What if $k = (\sqrt{2} - 2)^2$?

When $k = (\sqrt{2} - 2)^2$, the value of k is positive because $(\sqrt{2} - 2)^2 > 0$. Since Chebyshev's inequality is valid for $k > 0$, we can apply it in this case. When $k = (\sqrt{2} - 2)^2$, the inequality states:

$$\Pr\left(|X - \mu| \geq (\sqrt{2} - 2)^2\sigma\right) \leq \frac{1}{(\sqrt{2} - 2)^4}$$

This means that the probability that the random variable X deviates from its mean μ by more than $(\sqrt{2} - 2)^2\sigma$ (approximately 0.1716 times the standard deviation) is at most $\frac{1}{(\sqrt{2} - 2)^4}$.

In other words, there is at least a $1 - \frac{1}{(\sqrt{2} - 2)^4}$ chance that X lies within the interval $(\mu - (\sqrt{2} - 2)^2\sigma, \mu + (\sqrt{2} - 2)^2\sigma)$.

So, the derivation remains valid for $k = (\sqrt{2} - 2)^2$ and any other positive value of k.

Are you sure $(\sqrt{2} - 2)^2 > 0$?

Apologies for the confusion in my previous response. You are right; my statement was incorrect.

When $k = (\sqrt{2} - 2)^2$, the value of k is indeed non-negative, but it is actually equal to 0, not greater than 0. Here is the correct calculation:

$$(\sqrt{2} - 2)^2 = (1.414 - 2)^2 = (-0.586)^2 = 0.343396$$

Since $0.343396 > 0$, the value of k is positive in this case, and Chebyshev's inequality can be applied.

When $k = (\sqrt{2} - 2)^2$, the inequality states:

$$\Pr\left(|X-\mu| \geq (\sqrt{2}-2)^2\sigma\right) \leq \frac{1}{(\sqrt{2}-2)^4}$$

This means that the probability that the random variable X deviates from its mean μ by more than $(\sqrt{2} - 2)^2\sigma$ (approximately 0.3434 times the standard deviation) is at most $\frac{1}{(\sqrt{2}-2)^4}$.

In other words, there is at least a $1 - \frac{1}{(\sqrt{2}-2)^4}$ chance that X lies within the interval $(\mu - (\sqrt{2}-2)^2\sigma, \mu + (\sqrt{2}-2)^2\sigma)$.

So, the derivation remains valid for $k = (\sqrt{2}-2)^2$ and any other positive value of k.

Why do you say your previous statement was incorrect and $k = 0$? I don't understand.

I apologize for any confusion my previous responses may have caused. I made a mistake in my explanation. Let me clarify the situation:

When $k = (\sqrt{2}-2)^2$, the value of k is indeed positive, as I mentioned in my previous response. Here is the correct calculation:

$$(\sqrt{2}-2)^2 = (1.414-2)^2 = (-0.586)^2 = 0.343396$$

Since $0.343396 > 0$, the value of k is positive in this case, and Chebyshev's inequality can be applied.

When $k = (\sqrt{2}-2)^2$, the inequality states:

$$\Pr\left(|X-\mu| \geq (\sqrt{2}-2)^2\sigma\right) \leq \frac{1}{(\sqrt{2}-2)^4}$$

This means that the probability that the random variable X deviates from its mean μ by more than $(\sqrt{2} - 2)^2\sigma$ (approximately 0.3434 times the standard deviation) is at most $\frac{1}{(\sqrt{2}-2)^4}$.

In other words, there is at least a $1 - \frac{1}{(\sqrt{2}-2)^4}$ chance that X lies within the interval $(\mu - (\sqrt{2}-2)^2\sigma, \mu + (\sqrt{2}-2)^2\sigma)$.

So, the derivation remains valid for $k = (\sqrt{2}-2)^2$ and any other positive value of k.

E.3 Full Interaction Traces from "Boundary of Easy and Hard" Case Studies

We include the full interaction traces from case studies designed and undertaken by Timothy Gowers in Section 4.3.

Example 1 (from section 4.3)

Can you find an example of a set A of positive integers that satisfies the following three conditions?

1. $|A| = \min(A)$.
2. A consists of prime numbers only.
3. A has cardinality at least 5.

Yes, here's an example of a set A of positive integers that satisfies the given conditions: A = {5, 11, 17, 23, 29}

1. The cardinality of A is 5, which is also the minimum element in A.
2. All elements in A are prime numbers.
3. The cardinality of A is 5, which is at least 5.

Therefore, set A = {5, 11, 17, 23, 29} meets all three conditions.

Can you find an example of a set A of positive integers that satisfies the following three conditions?

1. $|A| = \min(A)$.
2. A consists of prime numbers only.
3. A has cardinality at least 6.

Yes, here is an example of a set A of positive integers that satisfies the given conditions: A = 7, 11, 13, 17, 19, 23

1. |A| = 6, and the minimum element in A is 7, so |A| = min(A).
2. All elements in A are prime numbers: 7, 11, 13, 17, 19, and 23.
3. A has cardinality 6, which is at least 6.

The set A satisfies all three conditions.

6 does not equal 7.

You are correct, my previous response contained an error. I apologize for the confusion. It is not possible to construct a set A of positive integers that satisfies all three conditions simultaneously.
To see why, let's consider the conditions:

1. |A| = min(A): This implies that the minimum element of the set A is equal to the size of the set A.
2. A consists of prime numbers only: This implies that every element of A is a prime number.
3. A has cardinality at least 6: This implies that A has 6 or more elements.

The smallest prime number is 2. If the set A consists of prime numbers only, the minimum element of A must be greater than or equal to 2. However, if |A| = min(A) and A has at least 6 elements, the minimum element in A would need to be at least 6, which is not prime. This creates a contradiction, making it impossible to construct a set A that satisfies all three conditions simultaneously.

Example 2 (from section 4.3)

Can you work out the integral $\int_{-1}^{1} x^3 \cos x dx$?

To solve the integral, we can use integration by parts. Integration by parts states that for two differentiable functions u and v:

$$\int u\, dv = uv - \int v\, du.$$

Let's choose $u = x^3$ and $dv = \cos x\, dx$. Then, we compute du and v:

$$du = \frac{d}{dx}(x^3)dx = 3x^2 dx, \quad v = \int \cos x\, dx = \sin x.$$

Now, using integration by parts:

$$\int_{-1}^{1} x^3 \cos x\, dx = \left[x^3 \sin x\right]_{-1}^{1} - \int_{-1}^{1} \sin x(3x^2)dx.$$

Let's evaluate the first term:

$$\left[x^3 \sin x\right]_{-1}^{1} = (1^3 \sin(1) - (-1)^3 \sin(-1)) = \sin(1) + \sin(-1).$$

For the remaining integral, we apply integration by parts again. This time, let $u = 3x^2$ and $dv = \sin x\, dx$. Then, we compute du and v:

$$du = \frac{d}{dx}(3x^2)dx = 6x dx, \quad v = \int \sin x\, dx = -\cos x.$$

Now, using integration by parts:

$$\int_{-1}^{1} 3x^2 \sin x\, dx = \left[-3x^2 \cos x\right]_{-1}^{1} - \int_{-1}^{1} -6x \cos x\, dx.$$

Evaluating the first term:

$$\left[-3x^2 \cos x\right]_{-1}^{1} = -3(1^2)\cos(1) - (-3(-1)^2\cos(-1)) = -3\cos(1) + 3\cos(-1).$$

For the remaining integral, notice that the integrand $-6x\cos x$ is an odd function. The integral of an odd function over a symmetric interval around zero is always zero:

$$\int_{-1}^{1} -6x\cos x\, dx = 0.$$

So, our final result is:

$$\int_{-1}^{1} x^3 \cos x\, dx = \sin(1) + \sin(-1) - 3\cos(1) + 3\cos(-1) + 0 = 2\sin(1) - 6\cos(1).$$

What is $\sin(1) + \sin(-1)$?

To find the value of $\sin(1) + \sin(-1)$, we can use the property of the sine function that states that $\sin(-x) = -\sin(x)$ for any real number x.
So, $\sin(-1) = -\sin(1)$.
Therefore,

$$\sin(1) + \sin(-1) = \sin(1) - \sin(1) = 0.$$

How about $-3\cos(1) + 3\cos(-1)$?

To find the value of $-3\cos(1) + 3\cos(-1)$, we can use the property of the cosine function that states that $\cos(-x) = \cos(x)$ for any real number x.
So, $\cos(-1) = \cos(1)$.
Therefore,

$$-3\cos(1) + 3\cos(-1) = -3\cos(1) + 3\cos(1) = 0.$$

In the light of that, would you like to revise your final answer above? (Please do not repeat the entire calculation – the rest was fine and just the last step was a problem.)

Apologies for the confusion in my previous response. Given that $\sin(1) + \sin(-1) = 0$ and $-3\cos(1) + 3\cos(-1) = 0$, the final answer for the integral should be:

$$\int_{-1}^{1} x^3 \cos x\, dx = 2\sin(1) - 6\cos(1) + 0 = 2\sin(1) - 6\cos(1).$$

However, as we just found out, the correct values for these sums are both 0. Therefore, the correct answer is:

$$\int_{-1}^{1} x^3 \cos x\, dx = 0.$$

Can you establish that in an easier way?

Yes, there is an easier way to establish that the integral is zero by noticing that the integrand is an odd function.

An odd function is a function that satisfies the property $f(-x) = -f(x)$ for all x. The product of an odd function and an even function is an odd function. In this case, x^3 is an odd function and $\cos x$ is an even function. Therefore, their product $x^3 \cos x$ is an odd function.

Now, when we integrate an odd function over a symmetric interval around zero (like $[-1, 1]$), the integral is always zero. This is because the positive and negative areas under the curve cancel each other out.

So, for the given integral:

$$\int_{-1}^{1} x^3 \cos x \, dx = 0$$

because the integrand is an odd function and the interval is symmetric around zero.

Can you calculate the integral $\int_{-1}^{1}(x^2 - 1/3)dx$?

Certainly! Let's evaluate the given integral:

$$\int_{-1}^{1} (x^2 - \frac{1}{3}) \, dx.$$

The integral of a sum (or difference) of functions is equal to the sum (or difference) of their integrals. So we can split the integral into two parts:

$$\int_{-1}^{1} x^2 \, dx - \int_{-1}^{1} \frac{1}{3} \, dx.$$

To find the antiderivative of x^2, we can use the power rule for integration:

$$\int x^n \, dx = \frac{x^{n+1}}{n+1} + C.$$

In this case, $n = 2$. So the antiderivative of x^2 is:

$$\frac{x^{2+1}}{2+1} = \frac{x^3}{3}.$$

Now, let's evaluate the first integral:

$$\int_{-1}^{1} x^2 \, dx = \left[\frac{x^3}{3}\right]_{-1}^{1} = \frac{1^3}{3} - \frac{(-1)^3}{3} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}.$$

For the second integral, we have a constant function $\frac{1}{3}$. The antiderivative of a constant function c is $cx + C$. In this case, $c = \frac{1}{3}$, so the antiderivative is $\frac{1}{3}x$:

$$\int_{-1}^{1} \frac{1}{3}\,dx = \left[\frac{1}{3}x\right]_{-1}^{1} = \frac{1}{3}(1) - \frac{1}{3}(-1) = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}.$$

Now, we can subtract the second integral from the first integral:

$$\int_{-1}^{1} (x^2 - \frac{1}{3})\,dx = \frac{2}{3} - \frac{2}{3} = 0.$$

Therefore, the integral evaluates to 0.

Can you see a simple reason that the integral had to be 0?

Yes, there is a simpler way to see that the integral had to be zero by noticing that the integrand is an odd function.
An odd function is a function that satisfies the property $f(-x) = -f(x)$ for all x. In this case, consider the function:

$$f(x) = x^2 - \frac{1}{3}.$$

When we plug in $-x$ for x, we get:

$$f(-x) = (-x)^2 - \frac{1}{3} = x^2 - \frac{1}{3} = -(x^2 - \frac{1}{3}) = -f(x).$$

So, $f(x) = x^2 - \frac{1}{3}$ is an odd function.
Now, when we integrate an odd function over a symmetric interval around zero (like $[-1, 1]$), the integral is always zero. This is because the positive and negative areas under the curve cancel each other out.
So, for the given integral:

$$\int_{-1}^{1} (x^2 - \frac{1}{3})\,dx = 0$$

because the integrand is an odd function and the interval is symmetric around zero.

Example 3 (from section 4.3)

Does there exist an order-preserving bijection between the rationals and the dyadic rationals?

No, there does not exist an order-preserving bijection between the rationals and the dyadic rationals.

The rationals, denoted by $\mathbb{Q}$, are the set of all numbers that can be expressed as the quotient of two integers, i.e., of the form a/b, where a and b are integers, and $b \neq 0$.

The dyadic rationals, denoted by $\mathbb{Q}_d$, are a subset of the rationals, and are the set of all numbers that can be expressed as the quotient of two integers, where the denominator is a power of 2, i.e., of the form $a/2^n$, where a and n are integers, and $n \geq 0$.

An order-preserving bijection between two sets means that for every pair of elements x and y in one set, if x $<$ y, then their corresponding images under the bijection will also satisfy the same order relation.

To see why there cannot be an order-preserving bijection between $\mathbb{Q}$ and $\mathbb{Q}_d$, consider any two distinct dyadic rationals, say p and q, such that there is no dyadic rational between them. In other words, p and q are consecutive dyadic rationals.

However, between any two distinct rationals (including dyadic rationals), there exists an infinite number of other rationals that are not dyadic. For example, between p and q, we can find a rational of the form (p+q)/3, which is not a dyadic rational since it has a denominator that is not a power of 2 (assuming p and q have distinct denominators).

Since there are an infinite number of non-dyadic rationals between any two consecutive dyadic rationals, it is impossible to construct an order-preserving bijection between $\mathbb{Q}$ and $\mathbb{Q}_d$. Any attempt to do so would necessarily "skip" some of the rationals, violating the requirement of a bijection.

Can you give me an example of a pair of dyadic rationals p and q with no dyadic rational between them?

Certainly! Let's consider two dyadic rationals p and q:

p = 1/4
q = 1/2

These can be expressed as:

p = 1 * 2^(-2)
q = 1 * 2^(-1)

There is no dyadic rational between p and q because any other dyadic rational with a denominator of 2 or 4 will either be smaller than p or larger than q. For example, if we consider 3/4, it is greater than q. On the other hand, if we consider 0/4 or 1/4, both of them are less than or equal to p.

Thus, p = 1/4 and q = 1/2 form a pair of consecutive dyadic rationals with no dyadic rational between them.

Is the number 3/8 a dyadic rational?

Yes, the number 3/8 is a dyadic rational. A dyadic rational is a rational number of the form a/2^n, where a and n are integers, and $n \geq 0$. In the case of 3/8, we can express it as:

3/8 = 3 * 2^(-3)

Here, a = 3 and n = 3, both of which are integers. Since the denominator is a power of 2 (specifically, 2^3), 3/8 is indeed a dyadic rational.

Does 3/8 lie between 1/4 and 1/2?

Yes, the number 3/8 lies between 1/4 and 1/2. To see this, we can compare the numbers by finding a common denominator, which in this case is 8:

1/4 = 2/8
3/8 = 3/8
1/2 = 4/8

Now, it's clear that:
2/8 < 3/8 < 4/8

So, 3/8 lies between 1/4 and 1/2.

Does there exist a dyadic rational that lies between 1/4 and 1/2?

Apologies for the confusion in my previous response. You are correct; there exists a dyadic rational between 1/4 and 1/2, which is 3/8.

1/4 = 2/8
3/8 = 3/8
1/2 = 4/8

As shown above, 3/8 is a dyadic rational that lies between 1/4 and 1/2.

OK, so can you give me an example of two consecutive dyadic rationals?

I apologize for the confusion in my previous responses. The term "consecutive dyadic rationals" is not precise, as there is always another dyadic rational between any two given dyadic rationals. However, I can provide an example of two dyadic rationals that are close to each other in terms of their position in the ordered set of dyadic rationals:

Let p = 1/4 and q = 1/8. Both are dyadic rationals, as they have denominators that are powers of 2. Now, let's find the dyadic rational halfway between them:

r = (p + q) / 2 = (1/4 + 1/8) / 2 = (2/8 + 1/8) / 2 = 3/16

Here, r = 3/16 is another dyadic rational that lies between p and q. In this sense, p and q are "close" dyadic rationals, but there is still another dyadic rational between them. In fact, there will always be another dyadic rational between any two given dyadic rationals, just as there are other rationals between any two given rationals.

In your earlier proof you claimed, and used, that there exists a pair p, q of dyadic rationals with no dyadic rational between them. Now that we have seen that that claim is false, I would like to ask again: is there an order-preserving bijection between the rationals and the dyadic rationals?

My apologies for the confusion in my earlier response. You are right; my claim was incorrect. Since there is no such pair of consecutive dyadic rationals with no dyadic rational between them, I must revisit the question.

Given that there is always another dyadic rational between any two dyadic rationals, the situation is actually similar to that of the rationals themselves. However, this does not guarantee the existence of an order-preserving bijection between the rationals ($\mathbb{Q}$) and the dyadic rationals ($\mathbb{Q}_d$). The crucial difference is that the dyadic rationals are a proper subset of the rationals, and there are "gaps" in the sense that there are rationals that cannot be expressed as a/2ˆn.

To answer the question, it is important to clarify that there does not exist an order-preserving bijection between the rationals and the dyadic rationals. An order-preserving bijection would require that for every pair of elements x and y in one set, if x $<$ y, then their corresponding images under the bijection would also satisfy the same order relation.

Since $\mathbb{Q}_d$ is a proper subset of $\mathbb{Q}$, it is not possible to create an order-preserving bijection between them. If we tried to create such a bijection, we would necessarily "skip" some of the rationals that are not dyadic, violating the requirement of a bijection.

What is your proof that we would necessarily skip some of the non-dyadic rationals?

You are correct in questioning my previous response. I must clarify that my previous statement was not properly justified. Let me address the question again.

To answer the question of whether there exists an order-preserving bijection between the rationals ($\mathbb{Q}$) and the dyadic rationals ($\mathbb{Q}_d$), we need to consider the properties of order-preserving bijections and the structure of the two sets.

An order-preserving bijection between two sets A and B is a bijective function f: A $\rightarrow$ B such that for every pair of elements x, y in A, if x $<$ y, then f(x) $<$ f(y). In other words, the bijection preserves the order of the elements in the sets.

The dyadic rationals ($\mathbb{Q}_d$) are a proper subset of the rationals ($\mathbb{Q}$), meaning that every dyadic rational is a rational, but not every rational is a dyadic rational. However, both $\mathbb{Q}$ and $\mathbb{Q}_d$ are dense in the real numbers, which means that between any two real numbers, there are both rational and dyadic rational numbers.

Upon reflection, it turns out that there does exist an order-preserving bijection between the rationals and the dyadic rationals. The key is to recognize that both sets are countable, and we can create a listing of both sets in increasing order. The order-preserving bijection can be established by matching the elements in the two lists one-to-one.

For example, let's list a few elements of the rationals and the dyadic rationals in increasing order:

$\mathbb{Q}$: -2, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, 2, ...
$\mathbb{Q}_d$: -2, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, 2, ...

We can define an order-preserving bijection f between $\mathbb{Q}$ and $\mathbb{Q}_d$ by associating each element in the $\mathbb{Q}$ list with the corresponding element in the $\mathbb{Q}_d$ list. This bijection preserves the order of the elements, and since both sets are countable, it is possible to match every element in $\mathbb{Q}$ with an element in $\mathbb{Q}_d$, even though $\mathbb{Q}_d$ is a proper subset of $\mathbb{Q}$.

In conclusion, there does exist an order-preserving bijection between the rationals and the dyadic rationals, and my previous claim was incorrect. I apologize for the confusion and appreciate your persistence in seeking a clear answer.

It seems to be a consequence of your previous answer that there is an order-preserving bijection between the integers and the rationals. Is that the case?

You are correct that my previous response is misleading, and I apologize for the confusion. There is no order-preserving bijection between the integers and the rationals, and my previous response does not imply that there is.

Upon further reflection, I must correct my previous response regarding the order-preserving bijection between the rationals ($\mathbb{Q}$) and the dyadic rationals ($\mathbb{Q}_d$).

In fact, there is no order-preserving bijection between the rationals and the dyadic rationals. To see this, consider any dyadic rational p. There exists a non-dyadic rational q such that q $>$ p, but there is no dyadic rational between p and q. For example, let p be any dyadic rational, and let q $=$ p $+$ (1/3)p. Then, q is not a dyadic rational, since it involves a factor of $1/3$, which cannot be expressed as a power of 2 in the denominator.

Now, suppose there were an order-preserving bijection f between $\mathbb{Q}$ and $\mathbb{Q}_d$. Since f is a bijection, there must be a dyadic rational r such that f(r) $=$ q. However, since f is also order-preserving, there must be a dyadic rational s such that p $<$ s $<$ r, which implies that p $<$ f(s) $<$ q. This is a contradiction since there is no dyadic rational between p and q.

Therefore, there cannot be an order-preserving bijection between the rationals and the dyadic rationals. I apologize for the confusion in my previous responses, and I appreciate your patience in seeking a clear answer.

Similarly, there is no order-preserving bijection between the integers and the rationals, as there are infinitely many rationals between any two integers, making it impossible to establish an order-preserving bijection between the two sets.

Have you heard of back-and-forth arguments?

Yes, I am familiar with back-and-forth arguments. The back-and-forth method is a technique used in set theory, logic, and model theory to prove that two structures are isomorphic or that they satisfy some other relation. It is particularly useful for proving the existence of isomorphisms or other relations between countable structures.

The back-and-forth method proceeds in a recursive way, building a relation (often an isomorphism) between two structures one element at a time. The construction alternates between the two structures, ensuring that the relation is consistent with the desired properties at each step. This alternating procedure is the reason for the name “back-and-forth.”

One of the classic applications of the back-and-forth method is the proof of the Cantor-Schröder-Bernstein theorem, which states that if there are injections (one-to-one functions) between two sets A and B, then there exists a bijection between A and B. The back-and-forth method is used to construct the bijection in a step-by-step manner, ensuring that the properties of a bijection are maintained throughout the process.

In the context of our previous discussion about order-preserving bijections between the rationals and the dyadic rationals, my earlier responses were not clear, and I apologize for any confusion. While it is true that both sets are countable, the back-and-forth method does not provide a direct way to establish an order-preserving bijection between the rationals and the dyadic rationals, as I previously claimed. The existence or non-existence of such a bijection is a more subtle question and requires a more nuanced argument.

What happens if one tries to use the back-and-forth method to build a bijection between the rationals and the dyadic rationals?

If we attempt to use the back-and-forth method to construct a bijection between the rationals ($\mathbb{Q}$) and the dyadic rationals ($\mathbb{Q}_d$), we will be able to create a bijection, but it will not be order-preserving.

Let's go through the back-and-forth construction:

1. Enumerate the elements of $\mathbb{Q}$ in a sequence $\{q_n\}$ and the elements of $\mathbb{Q}_d$ in another sequence $\{d\_n\}$.
2. Start by assigning q_1 to d_1.
3. For the next step, consider q_2. If q_2 is greater than q_1, find the first unassigned dyadic rational greater than d_1 and assign q_2 to that dyadic rational. If q_2 is less than q_1, find the first unassigned dyadic rational less than d_1 and assign q_2 to that dyadic rational.
4. Continue this process, alternating between the next unassigned elements in the $\mathbb{Q}$ and $\mathbb{Q}_d$ sequences.

The back-and-forth method will create a bijection between $\mathbb{Q}$ and $\mathbb{Q}_d$ since both sets are countable. However, this bijection will not be order-preserving, as the back-and-forth construction may assign a smaller rational to a larger dyadic rational or vice versa during the process.

In summary, while the back-and-forth method can be used to build a bijection between the rationals and the dyadic rationals, this bijection will not be order-preserving.

Why can one not just insist at each step that the order of the images is the same as the order of the original elements?

You are correct, and I apologize for the confusion in my previous responses. We can actually use the back-and-forth method to construct an order-preserving bijection between the rationals ($\mathbb{Q}$) and the dyadic rationals ($\mathbb{Q}_d$). The key is to ensure that the order of the images is the same as the order of the original elements at each step.

Here's an outline of the back-and-forth construction:

1. Enumerate the elements of $\mathbb{Q}$ in a sequence $\{q\_n\}$ and the elements of $\mathbb{Q}_d$ in another sequence $\{d\_n\}$.

2. Begin with an empty partial function f.

3. At each step, we will extend the function f by selecting an element from each sequence and making sure that the order is preserved.

   - Suppose we have defined f for n elements, and we are at the (n+1)-th step.
   - First, consider the (n+1)-th element of the $\mathbb{Q}$ sequence, q_{n+1}. Assign it to the first unassigned element in the $\mathbb{Q}_d$ sequence, d_{n+1}, such that the order is preserved; that is, for all $i \leq n$, $q_i < q_n + 1$ if and only if $f(q_i) < d_n + 1$.
   - Then, consider the (n+1)-th element of the $\mathbb{Q}_d$ sequence, d_{n+1}. Assign it to the first unassigned element in the $\mathbb{Q}$ sequence, q_{n+1}, such that the order is preserved; that is, for all $i \leq n$, $d_i < d_{n+1}$ if and only if $f^{(} - 1)(d_i) < q_{n+1}$.

4. Continue this process indefinitely.

This method ensures that at each step, the bijection is order-preserving, so the final bijection will also be order-preserving. I apologize for the confusion in my previous explanations, and I appreciate your persistence in seeking a clear answer.

Example 4 (from section 4.3)

Does there exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100?

No, there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

To see why, consider the least common multiple (LCM) of all the integers between 1 and 100, denoted as LCM(1, 2, ..., 100). This number is divisible by every integer in that range. Let's say there exists an integer n such that $n + k$ is a multiple of k for every k between 1 and 100. In this case, $n + k$ must also be divisible by LCM(1, 2, ..., 100) for every k between 1 and 100.

Now, consider the case where $k = 1$. We have $n + 1$ must be divisible by LCM(1, 2, ..., 100). This implies that n is one less than a multiple of LCM(1, 2, ..., 100). We can write this as:

$n = LCM(1, 2, ..., 100) * m - 1$, for some integer m.

Next, consider the case where $k = 2$. We have $n + 2$ must be divisible by 2. However, since $n = LCM(1, 2, ..., 100) * m - 1$, we have:

$n + 2 = LCM(1, 2, ..., 100) * m + 1$

Since LCM(1, 2, ..., 100) is an even number (it is divisible by 2), the expression $LCM(1, 2, ..., 100) * m + 1$ is an odd number. Thus, $n + 2$ is odd, and not divisible by 2, which contradicts the original assumption that $n + k$ is a multiple of k for every k between 1 and 100.

Hence, there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

Let me clarify the logical structure of the property in question. Formally speaking the question is whether the following statement is true.

$$\exists n \in \mathbb{N} \forall k \in \{1, 2, \dots, 100\} k | n + k$$

I apologize for the confusion, and I appreciate the clarification. Given the formal statement:

$$\exists n \in \mathbb{N} \forall k \in \{1, 2, \dots, 100\} k | n + k$$

The answer is still no, there does not exist a positive integer n that satisfies this property.

To see why, let's consider the case where $k = 2$. For $n + k$ to be divisible by 2 (i.e., even), n must be even, because an even number plus an even number always results in an even number. So, we can write $n = 2m$ for some positive integer m.

Now consider the case where $k = 3$. We want $n+3$ to be divisible by 3. Given that $n = 2m$, we have:

$2m + 3$

For this expression to be divisible by 3, m must be odd, because an even number plus an odd number will result in an odd number, and we want the sum to be even (a multiple of 3). So, we can write $m = 2p + 1$ for some positive integer p.

Now let's consider the case where $k = 5$. We want $n + 5$ to be divisible by 5. Given that $n = 2(2p + 1)$, we have:

$2(2p + 1) + 5 = 4p + 7$

However, no matter the value of p, the expression $4p + 7$ will always result in a number that is not divisible by 5. This contradicts the original assumption that $k|n + k$ for every k between 1 and 100.

Hence, there does not exist a positive integer n such that $\forall k \in \{1, 2, \dots, 100\} k|n + k$.

Let's focus on just one k. If $n + k$ is divisible by k, what can we say about n?

If $n + k$ is divisible by k, that means there exists an integer m such that $n + k = k \cdot m$. In this case, we can rewrite the equation as:

$n = k \cdot m - k$

From this equation, we can see that n is also a multiple of k. In other words, if $n+k$ is divisible by k, then n must be divisible by k as well.

With that as a hint, would you like to reconsider your answer to the question?

I apologize for my previous answers, and I appreciate your patience. Upon reconsidering, I see that my previous explanations were incorrect. Here is the correct answer:

Yes, there does exist a positive integer n such that $n+k$ is a multiple of k for every k between 1 and 100.

To show this, let $n = 0$. Then, for every k between 1 and 100:

$n + k = 0 + k = k$

Since k is a multiple of itself, $n+k$ is a multiple of k for every k between 1 and 100 when $n = 0$. However, the question asks for a positive integer n. In that case, we can take the least common multiple of the integers from 1 to 100, denoted by LCM(1, 2, ..., 100), and subtract 1:

$n = LCM(1, 2, ..., 100) - 1$

Now, for every k between 1 and 100:

$n + k = LCM(1, 2, ..., 100) - 1 + k$

Since LCM(1, 2, ..., 100) is a multiple of k for every k between 1 and 100, it follows that $n+k$ is also a multiple of k. So, there does exist a positive integer n such that $n+k$ is a multiple of k for every k between 1 and 100.

Something seems a bit odd there. If I take $k = 2$, I get that $n + 2 = LCM(1, 2, ..., 100) + 1$, which is odd, so not a multiple of 2.

You are absolutely right, and I apologize for the confusion in my previous response. I made a mistake in my explanation. Let's reconsider the problem:

We want to find if there exists a positive integer n such that $n+k$ is a multiple of k for every k between 1 and 100.

Let's assume that such a positive integer n exists. We will focus on the cases for $k = 2$ and $k = 3$.

Case $k = 2$: If $n + 2$ is a multiple of 2, then n must be even, as the sum of two even numbers is even. We can write $n = 2a$ for some positive integer a.

Case $k = 3$: If $n + 3$ is a multiple of 3, then n must be of the form $3b - 3$ for some positive integer b, since $3b - 3 + 3 = 3b$ is a multiple of 3.

Now, we must find a positive integer n that satisfies both conditions. From the first case, $n = 2a$, and from the second case, $n = 3b - 3$. Equating the two expressions for n, we get:

$2a = 3b - 3$

Adding 3 to both sides, we get:

$2a + 3 = 3b$

The left side of the equation is odd (an even number plus an odd number), while the right side of the equation is even (a multiple of 3). This leads to a contradiction, as an odd number cannot be equal to an even number.

Thus, there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

You appear to be suggesting that every multiple of 3 is even. Is that so?

You are correct, and I apologize for the mistake in my previous response. I made an incorrect statement about multiples of 3.

Not every multiple of 3 is even. In fact, multiples of 3 can be either even or odd. For example, 6 is an even multiple of 3, and 9 is an odd multiple of 3. My previous explanation was incorrect.

However, the conclusion that there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100 still holds. Let's consider the cases for $k = 2$ and $k = 3$ again.

Case $k = 2$: If $n + 2$ is a multiple of 2, then n must be even, as the sum of two even numbers is even. We can write $n = 2a$ for some positive integer a.

Case $k = 3$: If $n + 3$ is a multiple of 3, then n must be of the form $3b - 3$ for some positive integer b, since $3b - 3 + 3 = 3b$ is a multiple of 3.

Now, we must find a positive integer n that satisfies both conditions. From the first case, $n = 2a$, and from the second case, $n = 3b - 3$. Equating the two expressions for n, we get:

$2a = 3b - 3$

However, this equation does not inherently lead to a contradiction, and my previous explanation was incorrect in suggesting that it does.

Given that my previous explanations contained errors, let me reevaluate the problem.

Upon further consideration, there does exist a positive integer n that satisfies the condition $n + k$ is a multiple of k for every k between 1 and 100. Let $n = LCM(1, 2, ..., 100) - 1$. Then for any $k \in \{1, 2, ..., 100\}$, the value $n + k$ will be:

$n + k = LCM(1, 2, ..., 100) - 1 + k = LCM(1, 2, ..., 100) + (k - 1)$

Since the least common multiple of the integers between 1 and 100 is divisible by every integer in that range, $LCM(1, 2, ..., 100) + (k - 1)$ will also be a multiple of k. Thus, there does exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

Why is $LCM(1, 2, ..., 100) + (k - 1)$ a multiple of k?

I apologize for the mistake in my previous response. Upon reevaluating the problem, the statement I made about $LCM(1, 2, ..., 100) + (k - 1)$ being a multiple of k is incorrect.

In fact, there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

To illustrate why, let's consider the cases for $k = 2$ and $k = 3$:

Case $k = 2$: If $n + 2$ is a multiple of 2, then n must be even, as the sum of two even numbers is even. We can write $n = 2a$ for some positive integer a.

Case $k = 3$: If $n + 3$ is a multiple of 3, then n must be a multiple of 3 minus 3 (i.e., $3b - 3$) for some positive integer b, since $(3b - 3) + 3 = 3b$ is a multiple of 3.

Now, if there exists a positive integer n that satisfies both conditions, then $n = 2a = 3b - 3$. We can rewrite the equation as:

$2a + 3 = 3b$

This equation implies that $3b - 2a = 3$, which means that $3|(3b - 2a)$. However, this does not necessarily imply that $3|a$ or $3|b$. For example, let $a = 2$ and $b = 3$. Then, we have:

$3(3) - 2(2) = 9 - 4 = 5$

In this case, $3|(3b - 2a)$, but neither a nor b is divisible by 3. This shows that even if we find an integer n such that $n + k$ is a multiple of k for $k = 2$ and $k = 3$, it does not guarantee that the same integer n will work for all other values of k between 1 and 100.

Therefore, there does not exist a positive integer n such that $n + k$ is a multiple of k for every k between 1 and 100.

Previously when you tried to give an example, you chose LCM(1,2,...,100) -1. Can I ask what your motivation was for subtracting 1?

I apologize for the confusion in my previous responses. My motivation for subtracting 1 from the LCM(1, 2, ..., 100) was an attempt to find an integer n that could potentially satisfy the condition $n + k$ being a multiple of k for every k between 1 and 100.

The LCM(1, 2, ..., 100) is the smallest positive integer that is divisible by all integers between 1 and 100. Subtracting 1 from the LCM(1, 2, ..., 100) creates an integer that is 1 less than a multiple of all integers between 1 and 100. I hypothesized that this integer might satisfy the given condition, as adding k to $n = LCM(1, 2, ..., 100) - 1$ would result in a number that is a multiple of k for all integers between 1 and 100.

However, as shown in the subsequent discussion, this approach was incorrect, and the resulting integer $n = LCM(1, 2, ..., 100) - 1$ does not satisfy the condition for all k between 1 and 100.

Can you see a way of modifying your attempt that would make it correct?

After reconsidering the problem and the previous discussions, I realize that it is not possible to modify the LCM(1, 2, ..., 100) - 1 approach to find a positive integer n that satisfies the condition n + k being a multiple of k for every k between 1 and 100.

The main issue is that the relationship between n, k, and the least common multiple is not as straightforward as previously thought. As you pointed out, my previous explanation was incorrect, and the approach did not work for all values of k.

Considering the problem carefully, it becomes clear that there does not exist a positive integer n that satisfies the condition n + k being a multiple of k for every k between 1 and 100.

Suppose we were to take n = LCM(1,2,...,100). What would go wrong?

If we take n = LCM(1, 2, ..., 100), we can examine the condition that n + k should be a multiple of k for every k between 1 and 100.

Notice that for k = 1, we have n + 1 = LCM(1, 2, ..., 100) + 1, which is a multiple of 1, since every integer is a multiple of 1.

However, when we consider k = 2, we have:

n + 2 = LCM(1, 2, ..., 100) + 2

Since the LCM(1, 2, ..., 100) is an even number (as it must be a multiple of 2), adding 2 to it will result in another even number. In this case, n + 2 is a multiple of 2, so the condition holds for k = 2 as well.

But when we reach k = 3, we have:

n + 3 = LCM(1, 2, ..., 100) + 3

The LCM(1, 2, ..., 100) is a multiple of 3, so it can be expressed as 3m for some integer m. Thus, n + 3 = 3m + 3 = 3(m + 1). Since (m + 1) is an integer, n + 3 is a multiple of 3, and the condition is not satisfied.

So, taking n = LCM(1, 2, ..., 100) does not satisfy the condition n + k being a multiple of k for every k between 1 and 100.

You have confused me. It does satisfy the condition when k = 1, 2 or 3, as you convincingly showed. So why does it not satisfy it for all k between 1 and 100? For which k does it go wrong?

You are right, and I apologize for my confusion in my previous response. Upon re-reading my response, I realize that my explanation was unclear and incorrect. Let me correct myself.

Indeed, taking n = LCM(1, 2, ..., 100) does satisfy the condition n + k being a multiple of k for k = 1, 2, and 3, as I showed in my previous response. In fact, this choice of n will satisfy the condition for all k between 1 and 100.

When n = LCM(1, 2, ..., 100), for every k between 1 and 100, n + k can be written as:

n + k = LCM(1, 2, ..., 100) + k

Since the LCM(1, 2, ..., 100) is a multiple of every integer between 1 and 100, the expression LCM(1, 2, ..., 100) + k is also a multiple of k for each k in that range. This is because adding k to a multiple of k results in another multiple of k.

Therefore, taking n = LCM(1, 2, ..., 100) does indeed satisfy the condition n + k being a multiple of k for every k between 1 and 100.

Example 5 (from section 4.3)

Does there exist a graph with eight vertices such that each vertex has degree 3?

Yes, there exists a graph with eight vertices such that each vertex has degree 3. One example of such a graph is the 3-regular graph known as the cube or the 3-cube graph (also called the octahedral graph). This graph can be obtained by taking the vertices of a cube and connecting each vertex to its three neighboring vertices.

Another way to construct this graph is to number the vertices from 1 to 8 and then connect the vertices as follows:

1. Connect vertices 1, 2, 3, and 4 in a cycle, forming a square.
2. Connect vertices 5, 6, 7, and 8 in a cycle, forming another square.
3. Connect vertices 1 and 5, 2 and 6, 3 and 7, and 4 and 8.

Now, each vertex is connected to three other vertices, so each vertex has degree 3.

Does there exist a 5-regular graph with eight vertices?

A k-regular graph is a graph where all vertices have the same degree k. In this case, you're asking about a 5-regular graph with eight vertices.

To determine if such a graph exists, we can use the Handshaking lemma, which states that the sum of the degrees of all vertices in a graph is equal to twice the number of edges.

In a 5-regular graph with eight vertices, the sum of the degrees would be 5 * 8 = 40. According to the Handshaking lemma, this means the graph should have 40 / 2 = 20 edges.

Since there are no apparent contradictions so far, we can proceed to construct a 5-regular graph with eight vertices. One such graph is the 5-cube (also called the hypercube) graph Q_5 with its vertices labeled as 00000, 00001, 00011, 00010, 00100, 00101, 00111, 00110, where two vertices are adjacent if and only if their labels differ in exactly one bit.

However, this graph has 32 vertices, not 8. Therefore, we cannot construct a 5-regular graph with eight vertices using this method.

Another approach is to consider the complete bipartite graph K_{5,3}. This graph would have 5 vertices of degree 3 and 3 vertices of degree 5. However, this graph does not have all vertices with degree 5, so it's not a 5-regular graph.

Upon further examination, it appears that there is no 5-regular graph with eight vertices. This is because a 5-regular graph with eight vertices would have an odd degree sum (40) and an even number of vertices (8), which contradicts the Handshaking lemma, as the sum of the degrees of all vertices in a graph must be an even number.

Hmm, it seems to me that 40 is in fact an even number.

You are correct, and I apologize for the confusion in my response. The sum of degrees in a 5-regular graph with eight vertices is indeed even (40), and there is no contradiction with the Handshaking lemma. I appreciate your patience.

To construct a 5-regular graph with eight vertices, we can use the following method:

1. Label the vertices as A, B, C, D, E, F, G, and H.
2. Create the following edges:

- A to B, C, D, E, F
- B to C, D, E, F, G
- C to D, E, F, G, H
- D to E, F, G, H, A
- E to F, G, H, A, B
- F to G, H, A, B, C
- G to H, A, B, C, D
- H to A, B, C, D, E

The resulting graph is 5-regular, as all vertices have a degree of 5. This graph can be visualized as two 4-cycles (A-B-C-D-A and E-F-G-H-E), with each vertex in the first cycle connected to all vertices in the second cycle.

That isn't a graph: you say that A is joined to B, but you don't then say that B is joined to A. Also if you take two 4-cycles and join all the vertices in different cycles, then you get a 6-regular graph and not a 5-regular graph.

You are correct, and I apologize for my mistakes in the previous response. Let's try another approach.

Consider the Petersen graph, which is a 3-regular graph with 10 vertices. The Petersen graph can be represented as the complement of the K_5 graph. The complement of a graph G is a graph with the same vertices as G, in which two vertices are adjacent if and only if they are not adjacent in G.

If we remove two vertices from the Petersen graph (and their corresponding edges), we will be left with an induced subgraph that is a 5-regular graph with 8 vertices.

Here is the construction:

1. Label the vertices of the Petersen graph as A, B, C, D, E, F, G, H, I, and J.
2. Remove vertices I and J, along with their corresponding edges.
3. The remaining graph has the following edges:
   - A to B, C, D
   - B to A, D, E
   - C to A, E, F
   - D to A, B, F
   - E to B, C, G
   - F to C, D, H
   - G to E, H, F
   - H to F, G, E

This graph is 5-regular with eight vertices:

A: connected to B, C, D, G, H (degree 5)
B: connected to A, D, E, G, H (degree 5)
C: connected to A, E, F, G, H (degree 5)
D: connected to A, B, F, G, H (degree 5)
E: connected to B, C, F, G, H (degree 5)
F: connected to C, D, E, G, H (degree 5)
G: connected to E, F, H, A, B (degree 5)
H: connected to F, G, E, A, B (degree 5)

I apologize for the confusion in my previous responses. The graph described above is a 5-regular graph with eight vertices.

You seem to have defined two different graphs there. In the second one, C is joined to G but G is not joined to C, so it's not in fact a graph.

You are correct, and I apologize for the mistakes in my previous response. It appears that I made an error in the edge description, and the resulting graph is not 5-regular. I appreciate your patience.

Upon further examination, I realized that it is indeed possible to construct a 5-regular graph with eight vertices. Here's an example:

1. Label the vertices as A, B, C, D, E, F, G, and H.

2. Create the following edges:

   - A to B, C, D, E, F
   - B to A, C, D, G, H
   - C to A, B, E, F, G
   - D to A, B, E, F, H
   - E to A, C, D, G, H
   - F to A, C, D, G, H
   - G to B, C, E, F, H
   - H to B, D, E, F, G

This graph is 5-regular with eight vertices:

A: connected to B, C, D, E, F (degree 5)
B: connected to A, C, D, G, H (degree 5)
C: connected to A, B, E, F, G (degree 5)
D: connected to A, B, E, F, H (degree 5)
E: connected to A, C, D, G, H (degree 5)
F: connected to A, C, D, G, H (degree 5)
G: connected to B, C, E, F, H (degree 5)
H: connected to B, D, E, F, G (degree 5)

You can visualize this graph as two layers of vertices, with vertices A, B, C, and D in the first layer and vertices E, F, G, and H in the second layer. Connect vertices in the first layer to each other, forming a complete graph K_4. Similarly, connect vertices in the second layer to each other, forming another complete graph K_4. Now, connect the vertices of the first layer to the vertices of the second layer in a cycle-like fashion as described by the edges above.

I apologize for the confusion in my previous responses. The graph described here is a 5-regular graph with eight vertices.

Actually what you did here was to take the complete graph and remove the two disjoint cycles AGDCHA and BEFB. Since K_8 is 7-regular and the union of those two cycles is 2-regular, the resulting graph is 5-regular. It's not obvious to me how you came up with that example, given that you did not describe it accurately.

You are correct, and I apologize for any confusion in my description. The graph I provided is indeed obtained by removing two disjoint cycles from the complete graph K_8. Let me clarify the construction and reasoning behind the example.

1. Label the vertices of K_8 as A, B, C, D, E, F, G, and H.
2. K_8 is a 7-regular graph, as every vertex is connected to every other vertex.
3. Identify two disjoint cycles: AGDCHA and BEFB.
4. Remove the edges of these two cycles from K_8, resulting in a 5-regular graph:
   - A: connected to B, C, D, E, F (degree 5)
   - B: connected to A, C, D, G, H (degree 5)
   - C: connected to A, B, E, F, G (degree 5)
   - D: connected to A, B, E, F, H (degree 5)
   - E: connected to A, C, D, G, H (degree 5)
   - F: connected to A, C, D, G, H (degree 5)
   - G: connected to B, C, E, F, H (degree 5)
   - H: connected to B, D, E, F, G (degree 5)

The resulting graph is 5-regular with eight vertices. I hope this explanation clarifies the construction of the example. I apologize for any confusion or inaccuracies in my previous responses.

Example 6 (from section 4.3)

Can you find positive integers a and b such that the sequence $(1, 3, a, 7, 9, b, 13, 15)$ is strictly increasing and not an arithmetic progression?

Yes, I can find positive integers a and b that make the sequence strictly increasing and not an arithmetic progression. To ensure the sequence is strictly increasing, a must be greater than 3 and less than 7, and b must be greater than 9 and less than 13. To avoid forming an arithmetic progression, we need to choose a and b such that the difference between consecutive terms is not constant.

Let's choose $a = 5$. This gives us the sequence:

$$(1, 3, 5, 7, 9, b, 13, 15)$$

Now, we need to choose a value for b such that the sequence is not an arithmetic progression. We can select $b = 11$. This results in the following sequence:

$$(1, 3, 5, 7, 9, 11, 13, 15)$$

The differences between consecutive terms are:

$$2, 2, 2, 2, 2, 2, 2$$

However, this sequence is an arithmetic progression. To avoid this, we can choose $b = 12$. This results in the following sequence:

$$(1, 3, 5, 7, 9, 12, 13, 15)$$

The differences between consecutive terms are:

$$2, 2, 2, 2, 3, 1, 2$$

Since the differences are not constant, this sequence is not an arithmetic progression. Therefore, we can choose $a = 5$ and $b = 12$ to meet the given criteria.

Can you choose positive integers a, b and c such that the sequence $(1, a, b, c, 14)$ is an arithmetic progression of length 5?

Yes, we can choose positive integers a, b, and c such that the sequence $(1, a, b, c, 14)$ is an arithmetic progression of length 5. In an arithmetic progression, the difference between consecutive terms is constant, and we call this difference the common difference.

Let d be the common difference. Then we have:
$a = 1 + d$
$b = a + d = (1 + d) + d = 1 + 2d$
$c = b + d = (1 + 2d) + d = 1 + 3d$
Since the last term of the sequence is 14, we have:
$14 = c + d = (1 + 3d) + d = 1 + 4d$
Solving for d:
$4d = 14 - 1$
$4d = 13$
$d = \frac{13}{4}$
Since d is not an integer, it is not possible to choose positive integers a, b and c such that the sequence $(1, a, b, c, 14)$ is an arithmetic progression of length 5.

Example 7 (from section 4.3)

Can you find a colouring of the set $\{1, 2, \ldots, 9\}$ with three colours such that no number n in this set has the same colour as either $n + 1$ or $n + 3$ and each colour is used three times?

Let us consider the three colors A, B, and C. We want to assign these colors to the numbers 1, 2, ..., 9 such that no number n has the same color as n+1 or n+3, and each color is used three times.

Let's try coloring the numbers sequentially, avoiding the forbidden patterns. We can start by assigning color A to the number 1:

1(A) 2 3 4 5 6 7 8 9

Number 2 cannot have color A because it would violate the n+1 rule, and it can't have color B because it would force 5 to have color B too, violating the n+3 rule. So, we assign color C to number 2:

1(A) 2(C) 3 4 5 6 7 8 9

Number 3 cannot have color A (n+1 rule) or color C (n+3 rule), so it must have color B:

1(A) 2(C) 3(B) 4 5 6 7 8 9

Number 4 cannot have color B (n+1 rule) or color A (n+3 rule), so it must have color C:

1(A) 2(C) 3(B) 4(C) 5 6 7 8 9

Number 5 cannot have color C (n+1 rule) or color B (n+3 rule), so it must have color A:

1(A) 2(C) 3(B) 4(C) 5(A) 6 7 8 9

Number 6 cannot have color A (n+1 rule) or color C (n+3 rule), so it must have color B:

1(A) 2(C) 3(B) 4(C) 5(A) 6(B) 7 8 9

Now we can see that the remaining numbers 7, 8, and 9 have only one possibility for their coloring:

1(A) 2(C) 3(B) 4(C) 5(A) 6(B) 7(C) 8(A) 9(B)

This coloring satisfies all conditions: no number n has the same color as n+1 or n+3, and each color is used three times.

You've used colour C for 4 and for 7.

You are correct, I made a mistake in my response. I apologize for the confusion. Let's try another approach to solve this problem.

We can use a modular approach to find a suitable coloring. Since the numbers are in the range of 1 to 9, let's divide them into three groups based on the remainder when divided by 3:

Group 1 (remainder 1): 1, 4, 7
Group 2 (remainder 2): 2, 5, 8
Group 3 (remainder 0): 3, 6, 9
Now, let's assign the colors A, B, and C to these groups such that no number n has the same color as n+1 or n+3:

1(A) 2(B) 3(C) 4(A) 5(B) 6(C) 7(A) 8(B) 9(C)

In this coloring, no number n has the same color as n+1 or n+3, and each color is used three times.

Actually every number n from 1 to 6 has the same colour as n+3. This is a colossal failure of an example.

You are right, and I apologize for the mistakes in my previous responses. Let's try again to find a valid coloring for the set 1, 2, ..., 9 with three colors such that no number n has the same color as n+1 or n+3 and each color is used three times.

We will use the colors A, B, and C. To avoid any conflicts with the n+1 and n+3 rules, we can use a systematic approach to find a suitable coloring. Let's assign colors to odd numbers first and then to even numbers.

We can assign colors to odd numbers as follows:

1(A) 3(B) 5(C) 7(A) 9(B)

Now let's assign colors to even numbers:

2(C) 4(A) 6(B) 8(C)

So, the final coloring is:

1(A) 2(C) 3(B) 4(A) 5(C) 6(B) 7(A) 8(C) 9(B)

In this coloring, no number n has the same color as n+1 or n+3, and each color is used three times.

1 has the same colour as 4.

You are correct, and I apologize for my previous mistakes. Let's reevaluate the problem.

To find a valid coloring, let's represent the coloring using a string of length 9 where each character represents the color of the corresponding number.

Let the coloring of the first number be A:

1(A) 2 3 4 5 6 7 8 9

Since 2 cannot be colored with A (due to the n+1 rule), we will try coloring it with B:

1(A) 2(B) 3 4 5 6 7 8 9

Now, for number 3, it cannot be A (n+1 rule) and cannot be B (n+3 rule). So, we color it with C:

1(A) 2(B) 3(C) 4 5 6 7 8 9

Number 4 cannot be B (n+1 rule) and cannot be A (n+3 rule). So, it must have color C:

1(A) 2(B) 3(C) 4(C) 5 6 7 8 9

Number 5 cannot have color C (n+1 rule) or color B (n+3 rule). We are left with A:

1(A) 2(B) 3(C) 4(C) 5(A) 6 7 8 9

For number 6, we cannot use color A (n+1 rule) or color C (n+3 rule), so we choose B:

1(A) 2(B) 3(C) 4(C) 5(A) 6(B) 7 8 9

Now, we can see that we are unable to assign a color to number 7 without violating the n+1 or n+3 rule. Coloring it with A would violate the n+3 rule, and coloring it with C would violate the n+1 rule. Therefore, it is not possible to find a coloring of the set 1, 2, ..., 9 with three colors such that no number n has the same color as n+1 or n+3 and each color is used three times.

Example 8 (from section 4.3)

Can you find a sequence $(a_1, a_2, \ldots, a_9)$ of distinct integers with no increasing subsequence of length 3 and no decreasing subsequence of length 3?

Yes, such a sequence exists. One example of a sequence $(a_1, a_2, \ldots, a_9)$ of distinct integers with no increasing subsequence of length 3 and no decreasing subsequence of length 3 is given by:

$$a_1 = 1, a_2 = 4, a_3 = 2, a_4 = 5, a_5 = 3, a_6 = 6, a_7 = 7, a_8 = 9, a_9 = 8$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 3:

   - There are increasing subsequences of length 2, such as $(1, 4)$, $(2, 5)$, and $(3, 6)$, but none of length 3.

2. There is no decreasing subsequence of length 3:

   - There are decreasing subsequences of length 2, such as $(4, 2)$, $(5, 3)$, and $(9, 8)$, but none of length 3.

Thus, the sequence $(1, 4, 2, 5, 3, 6, 7, 9, 8)$ meets the criteria.

That has an increasing subsequence $(2, 3, 6, 7, 8)$.

I realize I made a mistake in the question. I want to forbid an increasing or decreasing sequence of length 4 and not 3. Nevertheless, your example still does not work, since it even has an increasing subsequence of length 5.

Apologies for the oversight. I misunderstood your question. Here's a corrected example of a sequence $(a_1, a_2, \ldots, a_9)$ of distinct integers with no increasing subsequence of length 4 and no decreasing subsequence of length 4:

$$a_1 = 4, a_2 = 1, a_3 = 5, a_4 = 2, a_5 = 6, a_6 = 3, a_7 = 7, a_8 = 9, a_9 = 8$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 2, 3)$ and $(4, 5, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 3, such as $(4, 1, 2)$ and $(5, 2, 3)$, but none of length 4 or more.

Thus, the sequence $(4, 1, 5, 2, 6, 3, 7, 9, 8)$ meets the updated criteria.

That has the increasing subsequence $(4, 5, 6, 7, 8)$, which is of length 5 (and so in particular has several subsequences of length 4).

Apologies again for the oversight. Here's a corrected example of a sequence $(a_1, a_2, \ldots, a_9)$ of distinct integers with no increasing subsequence of length 4 and no decreasing subsequence of length 4:

$$a_1 = 4, a_2 = 1, a_3 = 5, a_4 = 2, a_5 = 6, a_6 = 3, a_7 = 7, a_8 = 8, a_9 = 9$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 2, 3)$, $(1, 5, 6)$, and $(4, 5, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 2, such as $(4, 1)$ and $(5, 2)$, but none of length 4 or more.

Thus, the sequence $(4, 1, 5, 2, 6, 3, 7, 8, 9)$ meets the updated criteria.

No it doesn't. It's even worse as it has the subsequence $(4, 5, 6, 7, 8, 9)$, which has length 6. Maybe try arguing step by step this time, rather than starting with the answer. You might have better luck that way.

You are correct, my previous example was incorrect. I apologize for the confusion, and I appreciate your patience. Let's approach this step by step.

Consider two disjoint increasing subsequences $S_1 = (a, b, c)$ and $S_2 = (d, e, f)$, where $a < b < c$ and $d < e < f$. If we interleave these subsequences such that $a < d < b < e < c < f$, we obtain a sequence of distinct integers that has no increasing or decreasing subsequence of length 4.

Let's pick $S_1 = (1, 2, 3)$ and $S_2 = (4, 5, 6)$ as our disjoint increasing subsequences. Interleaving them as described above, we get:

$$a_1 = 1, a_2 = 4, a_3 = 2, a_4 = 5, a_5 = 3, a_6 = 6, a_7 = 7, a_8 = 8, a_9 = 9$$

Now let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 2, 3)$, $(4, 5, 6)$, and $(1, 4, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 2, such as $(4, 2)$ and $(5, 3)$, but none of length 4 or more.

Thus, the sequence $(1, 4, 2, 5, 3, 6, 7, 8, 9)$ meets the criteria.

That's the same failed example as before, but it's interesting to see how you produced it. The problem was that you didn't make full use of the interleaving idea. Can you have another go?

Apologies for repeating the same incorrect example. Let's try the interleaving approach once more, but this time we will use two disjoint increasing subsequences of length 3 and length 4.

Consider the disjoint increasing subsequences $S_1 = (a, b, c)$ and $S_2 = (d, e, f, g)$, where $a < b < c$ and $d < e < f < g$. If we interleave these subsequences such that $a < d < b < e < c < f < g$, we obtain a sequence of distinct integers that has no increasing or decreasing subsequence of length 4.

Let's pick $S_1 = (1, 3, 5)$ and $S_2 = (2, 4, 6, 7)$ as our disjoint increasing subsequences. Interleaving them as described above, we get:

$$a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 4, a_5 = 5, a_6 = 6, a_7 = 7, a_8 = 8, a_9 = 9$$

Now let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 2, 4)$, $(1, 3, 5)$, and $(2, 4, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 2, such as $(3, 2)$, $(5, 4)$, and $(7, 6)$, and length 3, such as $(9, 8, 7)$, but none of length 4 or more.

Thus, the sequence $(1, 2, 3, 4, 5, 6, 7, 8, 9)$ meets the criteria.

You're being silly now. You have just given me an increasing sequence of length 9.

Let me make my hint more explicit. When you interleaved two sequences of length 3, you still had three elements to put in, and those caused the problem. So how could you naturally modify that interleaved-sequence example?

I apologize for the confusion and appreciate your patience. Let's try a modified interleaving approach with two disjoint increasing subsequences of length 3.

Consider the disjoint increasing subsequences $S_1 = (a, b, c)$ and $S_2 = (d, e, f)$, where $a < b < c$ and $d < e < f$. We can interleave these subsequences such that $a < d < b < e < c < f$ and then insert the remaining 3 elements in a way that doesn't create increasing or decreasing subsequences of length 4.

Let's pick $S_1 = (1, 4, 7)$ and $S_2 = (2, 5, 8)$ as our disjoint increasing subsequences. Interleaving them as described above, we get:

$$a_1 = 1, a_2 = 2, a_3 = 4, a_4 = 5, a_5 = 7, a_6 = 8$$

Now we need to insert the remaining elements $3, 6$, and 9. We can insert 3 between 2 and 4, 6 between 5 and 7, and 9 at the end:

$$a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 4, a_5 = 5, a_6 = 6, a_7 = 7, a_8 = 8, a_9 = 9$$

Unfortunately, this sequence is also incorrect, as it is an increasing sequence of length 9. My previous attempts have not been successful, and I apologize for that.

Upon further reflection, it appears that it is not possible to construct a sequence of distinct integers of length 9 with no increasing or decreasing subsequence of length 4. This is because of the Pigeonhole Principle: if we partition the set of numbers into three parts, at least one part must contain at least four elements, resulting in an increasing or decreasing subsequence of length 4.

Here's an easier question. Can you find a sequence of length 4 with no increasing or decreasing sequence of length 3?

Yes, we can construct a sequence of length 4 with no increasing or decreasing sequence of length 3. One example is the following sequence:

$$a_1 = 2, a_2 = 1, a_3 = 4, a_4 = 3$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 3:

   - There are increasing subsequences of length 2, such as $(2, 4)$ and $(1, 3)$, but none of length 3.

2. There is no decreasing subsequence of length 3:

   - There are decreasing subsequences of length 2, such as $(2, 1)$ and $(4, 3)$, but none of length 3.

Thus, the sequence $(2, 1, 4, 3)$ meets the criteria.

That's a nice example. Can you now generalize it to obtain a sequence of length 9 with no increasing or decreasing subsequence of length 4?

My apologies for the confusion in my previous responses. You were correct, and it is indeed possible to construct a sequence of length 9 with no increasing or decreasing subsequence of length 4. We can generalize the previous example by interleaving two sequences of length 3 and adding the remaining elements in a way that doesn't create increasing or decreasing subsequences of length 4.

Let's start with the interleaved sequence, using two disjoint increasing subsequences of length 3, $S_1 = (1, 3, 5)$ and $S_2 = (2, 4, 6)$:

$$a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 4, a_5 = 5, a_6 = 6$$

Now we can insert the remaining elements $7, 8$, and 9 in a way that doesn't create increasing or decreasing subsequences of length 4. We can insert 7 between 3 and 4, 8 between 5 and 6, and 9 at the end:

$$a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 7, a_5 = 4, a_6 = 5, a_7 = 8, a_8 = 6, a_9 = 9$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 2, 4)$, $(1, 3, 5)$, and $(2, 4, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 2, such as $(7, 4)$, $(8, 6)$, and $(9, 6)$, and length 3, such as $(7, 5, 4)$, but none of length 4 or more.

Thus, the sequence $(1, 2, 3, 7, 4, 5, 8, 6, 9)$ meets the criteria.

No it doesn't. Instead of my checking your work for you, I'd like you to. Can you please identify for me an increasing or decreasing subsequence of $(1, 2, 3, 7, 4, 5, 8, 6, 9)$ of length 4?

I apologize for my mistake. Upon reevaluating the sequence, I see that there is an increasing subsequence of length 4 in the sequence $(1, 2, 3, 7, 4, 5, 8, 6, 9)$. The subsequence $(1, 3, 4, 5)$ has length 4 and is increasing.

To construct a sequence of length 9 with no increasing or decreasing subsequence of length 4, we can interleave two sequences of length 3 and insert the remaining elements in a way that doesn't create increasing or decreasing subsequences of length 4.

Let's start with the interleaved sequence, using two disjoint increasing subsequences of length 3, $S_1 = (1, 4, 7)$ and $S_2 = (2, 5, 8)$:

$$a_1 = 1, a_2 = 2, a_3 = 4, a_4 = 5, a_5 = 7, a_6 = 8$$

Now we can insert the remaining elements $3, 6$, and 9 by interleaving them between the existing elements in the sequence:

$$a_1 = 1, a_2 = 3, a_3 = 2, a_4 = 4, a_5 = 6, a_6 = 5, a_7 = 7, a_8 = 9, a_9 = 8$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(1, 3, 6)$, $(1, 2, 4)$, and $(2, 4, 6)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 2, such as $(3, 2)$, $(6, 5)$, and $(9, 8)$, and length 3, such as $(9, 7, 5)$, but none of length 4 or more.

Thus, the sequence $(1, 3, 2, 4, 6, 5, 7, 9, 8)$ meets the criteria.

That doesn't work either. Let me give you a big help, which is to advise you to start your sequence with $a_1 = 7, a_2 = 8, a_3 = 9$.

Thank you for your suggestion. I apologize for my previous incorrect responses. Given your starting values, we can construct the following sequence:

$$a_1 = 7, a_2 = 8, a_3 = 9, a_4 = 4, a_5 = 5, a_6 = 6, a_7 = 1, a_8 = 2, a_9 = 3$$

Let's verify that this sequence satisfies the conditions:

1. There is no increasing subsequence of length 4:

   - There are increasing subsequences of length 3, such as $(7, 8, 9)$, $(4, 5, 6)$, and $(1, 2, 3)$, but none of length 4 or more.

2. There is no decreasing subsequence of length 4:

   - There are decreasing subsequences of length 3, such as $(9, 6, 3)$, $(8, 5, 2)$, and $(7, 4, 1)$, but none of length 4 or more.

Thus, the sequence $(7, 8, 9, 4, 5, 6, 1, 2, 3)$ meets the criteria.

I agree that it meets the criteria. But can you prove that more rigorously?

To rigorously prove that the sequence $(7, 8, 9, 4, 5, 6, 1, 2, 3)$ has no increasing or decreasing subsequence of length 4, we can analyze the possible subsequences and show that none of them satisfy the conditions for length 4.

Consider the sequence:

$$a_1 = 7, a_2 = 8, a_3 = 9, a_4 = 4, a_5 = 5, a_6 = 6, a_7 = 1, a_8 = 2, a_9 = 3$$

1. There is no increasing subsequence of length 4:

   The longest increasing subsequences start with either 7, 4, or 1. Let's consider each of these cases:

   (a) If the increasing subsequence starts with 7, the only options are $(7, 8, 9)$, which is of length 3.

   (b) If the increasing subsequence starts with 4, the only options are $(4, 5, 6)$, which is of length 3.

   (c) If the increasing subsequence starts with 1, the only options are $(1, 2, 3)$, which is of length 3.

   There are no other possible increasing subsequences, so there cannot be an increasing subsequence of length 4.

2. There is no decreasing subsequence of length 4:

   The longest decreasing subsequences start with either 9, 6, or 3. Let's consider each of these cases:

   (a) If the decreasing subsequence starts with 9, the only options are $(9, 6, 3)$, which is of length 3.

   (b) If the decreasing subsequence starts with 6, the only options are $(6, 5, 4)$, $(6, 3, 1)$, and $(6, 5, 2)$, all of which are of length 3.

   (c) If the decreasing subsequence starts with 3, there are no decreasing subsequences of length greater than 1.

   There are no other possible decreasing subsequences, so there cannot be a decreasing subsequence of length 4.

   Based on this analysis, we can rigorously prove that the sequence $(7, 8, 9, 4, 5, 6, 1, 2, 3)$ has no increasing or decreasing subsequence of length 4.

Example 9 (from section 4.3)

Can you work out for me the sum $\sum_{r=2}^{n} \log(1 - 1/r)$?

Yes, I can help you work out the sum. Let's first analyze the expression inside the sum:

$$\log(1 - \frac{1}{r})$$

Now let's find the sum of these terms from $r = 2$ to n:

$$\sum_{r=2}^{n} \log(1 - \frac{1}{r})$$

To make it easier to work with, let's rewrite the expression using the property of logarithms that states: $\log(a) - \log(b) = \log(\frac{a}{b})$. In this case, we have:

$$\sum_{r=2}^{n} \log(\frac{r-1}{r})$$

Now, let's compute the sum:

$$\sum_{r=2}^{n} \log(\frac{r-1}{r}) = \log(\frac{1}{2}) + \log(\frac{2}{3}) + \log(\frac{3}{4}) + \cdots + \log(\frac{n-1}{n})$$

Notice that adjacent terms have denominators and numerators that cancel each other out, and we can rewrite the sum as:

$$\sum_{r=2}^{n} \log(\frac{r-1}{r}) = \log(\frac{1}{n})$$

So, the sum is:

$$\log(\frac{1}{n})$$

Can you explain in more detail why the denominators and numerators in adjacent terms cancel out?

Sure, I'd be happy to explain.

Let's write out the sum with a few more terms so we can see what's happening:

$$\log(\frac{1}{2}) + \log(\frac{2}{3}) + \log(\frac{3}{4}) + \log(\frac{4}{5}) + \cdots + \log(\frac{n-1}{n})$$

The property of logarithms we're using here is $\log(ab) = \log(a) + \log(b)$. This allows us to combine logarithms that are being added into a single logarithm of their product.

So, if we have two adjacent terms like $\log(\frac{2}{3})$ and $\log(\frac{3}{4})$, we can combine them into a single term:

$$\log(\frac{2}{3}) + \log(\frac{3}{4}) = \log\left(\frac{2}{3} \cdot \frac{3}{4}\right) = \log(\frac{2}{4}) = \log(\frac{1}{2}).$$

Notice that the '3' in the numerator of the first fraction cancelled out with the '3' in the denominator of the second fraction.

When we extend this to the entire series, we see that each term cancels with part of the next term, leaving only the very first numerator (which is 1) and the very last denominator (which is n):

$$\log(\frac{1}{2}) + \log(\frac{2}{3}) + \log(\frac{3}{4}) + \cdots + \log(\frac{n-1}{n}) = \log(\frac{1}{n}).$$

So, the sum simplifies to just $\log(\frac{1}{n})$.

Can you work out $\sum_{m=1}^{n} \log(1/m(m+1))$?

Let's compute the sum:

$$\sum_{m=1}^{n} \log\left(\frac{1}{m(m+1)}\right)$$

First, we rewrite the expression inside the sum using the property of logarithms that states: $\log(a) - \log(b) = \log(\frac{a}{b})$. In this case, we have:

$$\sum_{m=1}^{n} \log\left(\frac{m}{m+1}\right)$$

Now, let's compute the sum:

$$\sum_{m=1}^{n} \log\left(\frac{m}{m+1}\right) = \log(\frac{1}{2}) + \log(\frac{2}{3}) + \log(\frac{3}{4}) + \cdots + \log(\frac{n}{n+1})$$

Here, we can see a telescoping series where each term cancels out part of the next term. We are left with only the very first term in the series, $\log(\frac{1}{2})$, and the last term, $\log(\frac{n}{n+1})$:

$$\sum_{m=1}^{n} \log\left(\frac{m}{m+1}\right) = \log(\frac{1}{2}) - \log(\frac{n}{n+1})$$

Using the properties of logarithms, we can rewrite the above as:

$$\sum_{m=1}^{n} \log\left(\frac{m}{m+1}\right) = \log\left(\frac{1}{2(n+1)}\right)$$

So, the sum is:

$$\log\left(\frac{1}{2(n+1)}\right)$$

Let $f : \mathbb{N} \to \mathbb{R}$ and define a function $g : \mathbb{N} \to \mathbb{R}$ by $g(n) = f(n) - 2f(n+1) + f(n+2)$. Can you find an expression for $\sum_{n=1}^{N} g(n)$?

Let's work out the sum. We have

$$\sum_{n=1}^{N} g(n) = \sum_{n=1}^{N} (f(n) - 2f(n+1) + f(n+2))\,.$$

We can break this into three separate sums:

$$= \sum_{n=1}^{N} f(n) - 2\sum_{n=1}^{N} f(n+1) + \sum_{n=1}^{N} f(n+2).$$

Let's shift the indices in the second and third sums to make the limits match up with the first sum:

1. For the second sum, we substitute $m = n + 1$, so $n = m - 1$. When $n = 1$, $m = 2$; when $n = N$, $m = N + 1$.

2. For the third sum, we substitute $m = n + 2$, so $n = m - 2$. When $n = 1$, $m = 3$; when $n = N$, $m = N + 2$.

Doing this, we get:

$$= \sum_{n=1}^{N} f(n) - 2 \sum_{m=2}^{N+1} f(m) + \sum_{m=3}^{N+2} f(m).$$

Now, we can write out the sums explicitly to see what's happening:

1. The first sum is: $f(1) + f(2) + f(3) + \cdots + f(N)$.
2. The second sum is: $2f(2) + 2f(3) + \cdots + 2f(N + 1)$.
3. The third sum is: $f(3) + f(4) + \cdots + f(N + 2)$.

By looking at these sums, we can see that most of the terms will cancel out. For example, the term $f(2)$ is added in the first sum and subtracted twice in the second sum. Similarly, the term $f(3)$ is added in the first sum, subtracted twice in the second sum, and added once in the third sum.

After all the cancellations, we are left with:

$$= f(1) - 2f(2) + f(N) - 2f(N + 1) + f(N + 2)$$

So, the sum $\sum_{n=1}^{N} g(n)$ simplifies to $f(1) - 2f(2) + f(N) - 2f(N + 1) + f(N + 2)$.