

НЕЙРОМЕРЕЖЕВІ МЕТОДИ ФОРМАЛІЗАЦІЇ МАТЕМАТИЧНИХ ТЕКСТІВ

У цій статті досліджуються можливості великих мовних моделей (ВММ) для автоматизації перетворення математичних тверджень із природної мови у формальні вирази, що спрощує процес пошуку та генерації розв'язань. Завдяки досягненням в області штучного інтелекту, такі моделі як GPT-4 і LLaMA-3 представляють перспективні можливості для вирішення задачі математичної формалізації текстів. Ці моделі демонструють значні здібності до розуміння та перекладу текстів, що зменшує невизначеності у природномовній версії математичних задач та поліпшує процес збільшення математичної бази знань відповідних систем. Ефективність цих моделей оцінюється в переході від природної мови до математичного формалізму. Хоча ВММ показують значні результати у семантичному аналізі, точні обчислення і символічні маніпуляції потребують подальших удосконалень. Інтеграція з системами автоматичних доведень, такими як LEAN, досліджується для поліпшення процесу побудови формальних доведень. Отримані результати вказують на великий потенціал використання моделей штучного інтелекту в автоматизації математичного міркування.

Ключові слова: штучний інтелект; математичні задачі; формальні мови; обробка природної мови; великі мовні моделі; системи автоматичного пошуку доведень.

NIKOLAIEV ANDRII

Taras Shevchenko National University of Kyiv

NEURAL NETWORK METHODS FOR MATHEMATICAL TEXTS FORMALISATION

This paper examines the potential of large language models (LLMs) to automate the conversion of mathematical statements from natural language into formal expressions, thereby streamlining the process of solution generation. With advancements in artificial intelligence, particularly in natural language processing, models such as GPT-4 and LLaMA-3 present promising applications for mathematical formalisation tasks. These models showcase significant capabilities in text comprehension and translation, which help reduce ambiguities in natural language descriptions of mathematical problems. The effectiveness of these models is evaluated in the context of transitioning from natural language to mathematical formalism. While LLMs excel at semantic analysis and text translation, challenges persist in their application to precise computations and symbolic manipulation, which are critical for solving complex mathematical problems with accuracy. To address these limitations, integration with formal proof systems such as LEAN is explored, enhancing the process of generating formal proofs and ensuring logical consistency in mathematical arguments. Furthermore, the integration of LLMs with formal systems is investigated for its potential to facilitate the development of systems that automatically generate proofs from mathematical texts. By leveraging the strengths of LLMs in linguistic understanding and the precision of formal systems, a hybrid approach is proposed to improve the reliability of automated problem-solving processes. The findings indicate that with continued development, LLMs can significantly contribute to automating mathematical reasoning. However, ongoing refinement is necessary to fully realise their potential in academic and educational contexts, particularly in handling complex mathematical reasoning and extensive problem sets.

Keywords: artificial intelligence; mathematical problems; formal representation; natural language processing; large language models; automated theorem proving.

Постановка проблеми

Останніми роками застосування штучного інтелекту у розв'язанні математичних задач привернуло увагу багатьох науковців. Великі мовні моделі (ВММ), відомі своєю здатністю розуміти та генерувати текст, подібний до людського, відкрили нові можливості у сфері математичної формалізації текстів. Ця стаття розглядає потенціал ВММ у автоматизації перетворення математичних висловлювань із природної (людської) мови на формальні вирази для подальшої генерації розв'язань.

Аналіз досліджень та публікацій

Автоматизація розв'язання математичних задач викликає інтерес вже понад 50 років. Останні досягнення в галузі штучного інтелекту (ШІ), особливо в обробці природної мови, запропонували перспективні інструменти для полегшення процесів розв'язання математичних задач.

У 1970 році, академік В.М. Глушков опублікував статтю, в якій обговорювалися різноманітні виклики штучного інтелекту і представив дослідницьку програму під назвою "Алгоритм Очевидності" [9]. У цій програмі він окреслив своє бачення у процесі пошуку доказів теорем з використанням комп'ютерів. Глушков запропонував зосередитися на створенні системи, яка автоматизує пошук доказів теорем, зокрема – розробку формальних мов для написання математичних текстів у зрозумілій для людини формі [3], створення процедур пошуку доведень на основі комп'ютерного розуміння очевидності, використовуючи вже існуючі або щойно отримані знання, а також забезпечення користувацьких інтерфейсів для підтримки системи. Після публікації програми було зроблено дві основні спроби її реалізації. Перша призвела до створення системи автоматизації доказів SAD у 1980 році [10], а друга – до її англійської модифікації в 2002 році [11]. Також з'явилися інші автоматизовані довідники теорем, які використовували системи формалізації, такі як доказові асистенти Isabelle [13] і LEAN [14]. Ці інструменти є високоточними і забезпечують логічно несуперечливу структуру для математичних доведень. Вони, як правило, вимагають повністю формалізованих даних, які можуть бути зрозумілими для автоматизованих довідників теорем, але не для зручного розуміння людиною. У статті [12] автори запропонували формалізувати дані мовою ForTheL, яка націлена на відбиття природного (людського)

стилю та структури математичного аргументу, але підходить для автоматизованої обробки. Однак, будь-яка підготовка даних у формалізованій формі, як вказують автори FIMO [16], вимагає значних зусиль від експертів і високих витрат.

Різноманітність тем задач та їх перетини підкреслюють важливість дослідження найкращих практик та моделей для подальшого розвитку обчислювальних систем. Великі мовні моделі, такі як GPT-4 [4], PaLM-2 [1] досягли значного прогресу у виконанні широкого спектра завдань, пов'язаних із мовою, особливо у вирішенні викликів математичного міркування. Не є виключенням нещодавні відкриті мовні моделі, як наприклад, LLaMA-3 [6]. Дані моделі являють собою значний крок вперед у обробці та розумінні природної мови; проте вони часто стикаються з труднощами при виконанні складних математичних міркувань. Математичне міркування мовних моделей було покращено в існуючих результатах або за рахунок поетапного природномовного обґрунтування [8], або шляхом синтезу й виконання програм для отримання правильних відповідей [2].

Слід зазначити, що природномовне обґрунтування за допомогою цих моделей підходить для завдань, що потребують семантичного аналізу, перекладу текстів та генерації ідей. Але цей підхід часто виявляється недостатнім для задач, що потребують точних обчислень, символічної маніпуляції або алгоритмічної обробки. Автори роботи [17] продемонстрували, як невеликі зміни у тексті з навіть нескладними математичними задачами можуть призвести до суттєво погіршеної роботи мовних моделей.

Синтез і виконання програм забезпечують необхідну стійкість для таких завдань, що точно виконують операції та передають складність обчислення інструментам, таким як системи автоматичного розв'язання рівнянь. Автори статті [5] продемонстрували як за допомогою мовних моделей було успішно перекладено задачі рівня математичних олімпіад до Isabelle з послідуочим використанням системи для пошуку доведень.

Показовим прикладом даної ідеї є нещодавні зусилля Google DeepMind, які запропонували моделі AlphaGeometry та AlphaProof, що включають нейро-символічні системи для пошуку розв'язань та ефективний спосіб автоформалізації умов задач за допомогою великої мовної моделі Gemini [15]. Ці системи успішно формалізували задачі рівня міжнародної математичної олімпіади [7] та створили велику бібліотеку формальних задач різної складності. Хоча відповідні моделі та дані, використані для навчання, не є доступними, видно, що різноманітність проблем вимагає застосування різних технік. У поєднанні з оновленою версією вже оприлюдненої AlphaGeometry, система змогла вирішити 4 з 6 математичних задач останньої олімпіади 2024 року.

Формулювання цілей статті

Метою роботи є вивчення можливостей великих мовних моделей для автоматизованої формалізації математичних текстів, зосередивши увагу на їх здатності поліпшувати точність і ефективність процесу створення формальних математичних доказів. Дослідження включає розробку процедури, яка інтегрує природномовне розуміння та алгоритмічну обробку для подолання труднощів, пов'язаних із семантичною складністю та структурною різноманітністю математичних виразів, а також обґрунтування практичного застосування таких інструментів у розв'язанні математичних задач.

Виклад основного матеріалу

Опис підходу. Запропонований підхід складається з етапів описаних на рис. 1. Система отримує на вхід задачу описану природною мовою, після чого проходить етап формалізації за допомогою ВММ та подається у систему автоматичного пошуку доведень. Результати роботи програми деформулізуються зворотню у природню мову за допомогою ВММ задля комфортного перегляду людиною.

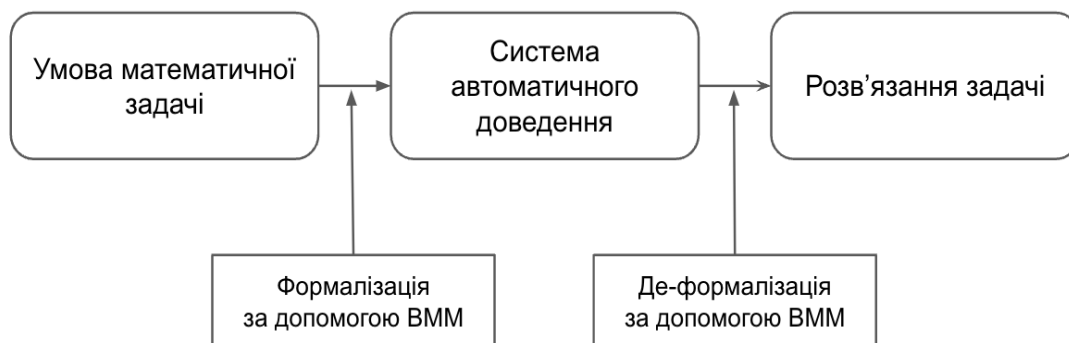


Рис. 1. Опис процедури для формалізації та де-формалізації заданої умови та розв'язання задачі

Вхідний набір даних. У якості вхідних даних використовуються математичні тексти у вільному доступі. Значна кількість математичної літератури доступна у оцифрованих книжках та онлайн ресурсах.

Вибір та робота з моделями. Для експериментування з математичними текстами було обрано моделі GPT-4o та LLaMA-3.1, які показали високий рівень роботи у вмінні працювати з текстами, а також генерувати код різними мовами програмування, що є необхідним для виконання поставленої задачі у контексті формалізації математичних задач. Звернення до GPT-4o виконувалися за допомогою OpenAI Playground (рис.2), до моделі LLaMA-3.1 – через LM Studio (рис.3). Обидва середовища дозволяють налаштувати

параметри моделей, таких як, наприклад, *temperature* задля подальшої підготовки до поставленої задачі.

Процес формалізації. Формалізація включає перетворення тверджень природною мовою у символічні математичні представлення. На цьому етапі моделі допомагають усунути неточності текстів представлених заданою мовою, перетворивши вхідний текст у точні математичні вирази, з якими можна працювати за допомогою відповідних систем.

Генерація розв'язань. Задля пошуку та перевірки коректності розв'язань використовується система LEAN. Після завершення формалізації, дана система перевіряє розв'язання задачі на основі згенерованих символічних виразів.

Процес деформалізації. Процес деформалізації перетворює математичні розв'язки назад у зрозумілу людині природню мову задля отримання якісних пояснень до розв'язань задач. Хоча цей етап не є обов'язковим для успішного пошуку результатів, він є важливим для забезпечення кінцевого розуміння результатів роботи системи користувачем.

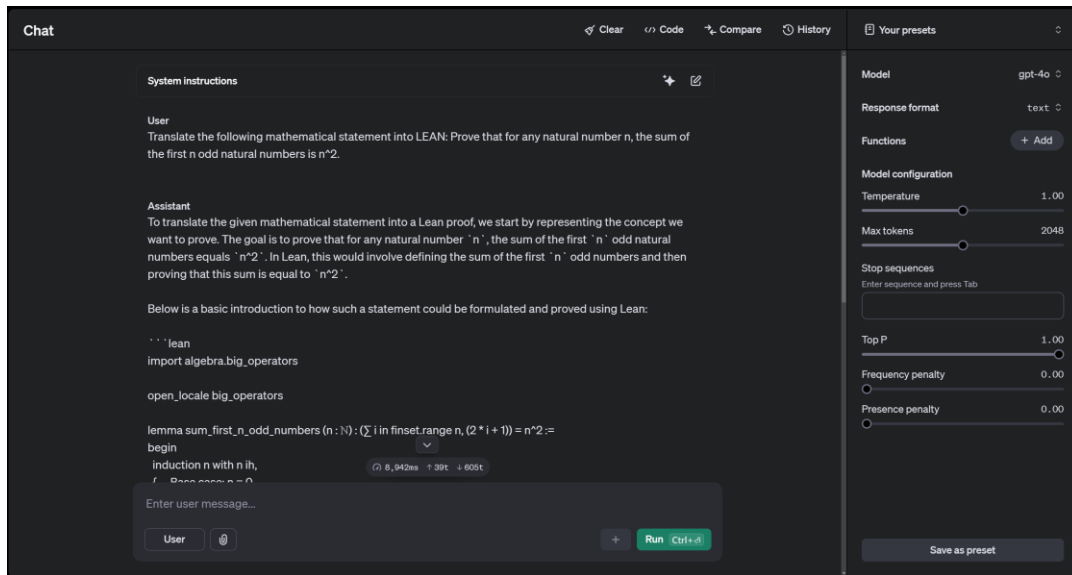


Рис. 2. OpenAI Playground для роботи з GPT-4o

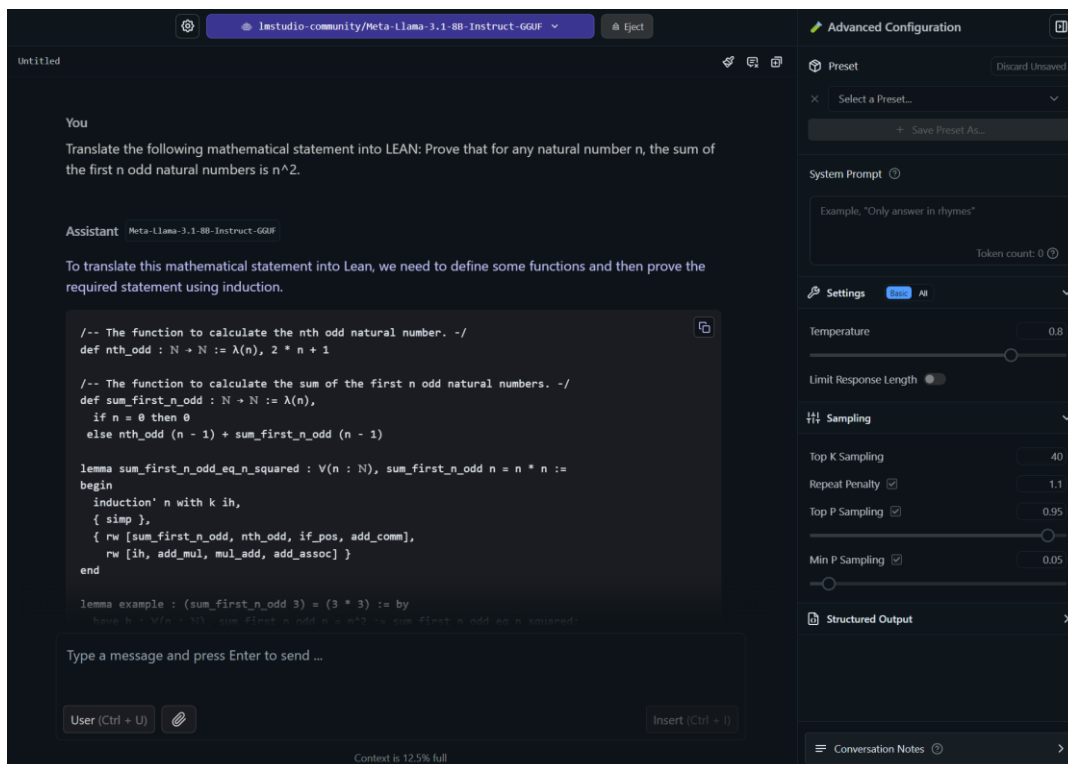


Рис. 3. LM Studio для роботи з LLaMA 3.1

Приклад використання запропонованої процедури у розв'язанні математичної задачі наведено у таблиці 1. Слід окремо зауважити, що не зважаючи на те, що наведений приклад описано англійською мовою, запропонований підхід є робочим у випадку обробки текстів заданих іншими мовами, наприклад,

українською. Якщо обрана ВММ була навчена на великій кількості даних відповідної мови, це не буде перешкодою для задовільної якості роботи системи.

Таблиця 1

Етапи здійснення формалізації задачі за допомогою мовної моделі на прикладі задачі

<p>1. Математична задача англійською мовою: Prove that for any natural number n, the sum of the first n odd natural numbers is n^2.</p>
<p>2. Проміжний крок – звернення до GPT-4o: <u>Prompt:</u> Translate the following mathematical statement into LEAN: {Умова задачі подана природньою мовою}.</p>
<p>3. Формалізація умови та розв’язання задачі за допомогою системи LEAN:</p> <pre>import data.nat.parity open nat -- The statement of the theorem theorem sum_of_first_n_odd_numbers (n : ℕ) : ∑ i in range n, (2 * i + 1) = n^2 := begin -- We will prove this by induction on n induction n with n ih, -- Base case: n = 0 { -- The sum from 0 to 0 is 0, and 0^2 is also 0 simp, }, -- Induction step: assume it is true for n, prove for n+1 { -- We need to show ∑ i in range (n + 1), (2 * i + 1) = (n + 1)^2 -- The sum can be broken down into the sum for range n plus the last term (2 * n + 1) rw sum_range_succ, -- Using the induction hypothesis ih: ∑ i in range n, (2 * i + 1) = n^2 rw ih, -- Thus, n^2 + (2 * n + 1) = (n + 1)^2 -- Simplifying the right hand side: expand (n + 1)^2 to n^2 + 2 * n + 1 simp, } end</pre> <p>У наведеному коді задля доведення теореми було використано математичну індукцію. Початкова умова перевіряється за допомогою `simp`, яка підтверджує, що сума перших нульових непарних чисел дорівнює нулю. Для переходу індукції ми використовуємо факт, що сума перших $n + 1$ непарних чисел дорівнює $(n + 1)^2$ за припущення $\sum_{i=0}^{n-1} (2i + 1) = n^2$.</p>
<p>4. Проміжний крок – звернення до GPT-4o: <u>Prompt:</u> Translate the following mathematical text into natural language: {Умова задачі подана мовою LEAN}.</p>
<p>5. Розв’язання англійською мовою: To prove that the sum of the first n odd natural numbers is n^2, we use mathematical induction: 1) Base case: For $n = 0$, the sum is 0, which equals 0^2. 2) Inductive step: Assume the sum of the first n odd natural numbers is n^2. We need to prove it for $n + 1$.</p> <p>The $(n+1)$-th odd number is $2n + 1$. Thus, the sum of the first $n + 1$ odd numbers is $(1 + 3 + 5 + \dots + (2n - 1)) + (2n + 1)$</p> <p>By the inductive hypothesis, this equals to $n^2 + (2n + 1) = n^2 + 2n + 1 = (n + 1)^2.$</p> <p>Thus, by induction, for any natural number n, the sum of the first n odd natural numbers is n^2.</p>

Висновки

Дослідження демонструє значний потенціал великих мовних моделей у автоматизації перетворення математичних тверджень з природньої мови у формальні математичні вирази. Завдяки прогресу у напрямку обробки природньої мови, GPT-4 і LLaMA-3, моделі які досліджувалися у даній статті, демонструють високу здатність до інтерпретації та трансляції текстів, що суттєво знижує неоднозначності, притаманні природномовним описам математичних задач та збільшує якість підготовки даних. Це не тільки робить дані системи ефективними у роботі, а й зменшує витрати на формалізацію текстів, які у традиційних підходах потребують професійних фахівців для роботи з відповідними формальними мовами.

Застосування мовних моделей для точних обчислень та символічних маніпуляцій, необхідних для роботи з математичними твердженнями, є обмеженим. Усунення цих обмежень можливе через інтеграцію моделей із системами формальних доведень, такими як LEAN, що покращує послідовність і точність у побудові формальних доведень. Залишається нагальна потреба в удосконаленні моделей для забезпечення їх ефективності в обробці завдань високої складності та великих обсягів даних, які досягаються за рахунок збільшення бази формальних знань.

Слід додатково зазначити, що у якості вхідних даних для роботи системи, можливе використання будь-яких природніх текстів написаних людською мовою, що робить можливість задіяння даного підходу поширеним задля розширення математичної бази знань системи та швидкого опрацювання нових математичних задач. Додатковий крок деформалізації згенерованих розв'язань задля перевірки точності розв'язку наданого у формальному вигляді, робить результати роботи системи доступними для сприйняття не лише фахівцями, а й широкою аудиторією.

Запропонований метод відкриває можливості для створення вдосконалених систем автоматизованого пошуку доведень. Зокрема, одним зі шляхів для подальшого удосконалення запропонованої системи є інтеграція елементу, який дозволяє обирати складові підсистеми для виконання проміжних етапів роботи системи, керуючись наявними у доступі ВММ та систем автоматичного пошуку доведень в залежності від тематики обраної математичної задачі.

Література

1. Anil, Rohan, and et al. 2023. PaLM 2 Technical Report: <https://arxiv.org/abs/2305.10403>.
2. Gao, Luyu, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-Aided Language Models: <https://arxiv.org/abs/2211.10435>.
3. Glushkov, V. M., K. P. Vershinin, Yu. V. Kapitonova, et al. 1974. About a Formal Language for Recording Mathematical Texts: Automation of the Search for Proofs of Theorems in Mathematics.
4. OpenAI. 2024. GPT-4 Technical Report: <https://arxiv.org/abs/2303.08774>.
5. Wu, Y., Jiang, A. Q., Li, W., Rabe, M. N., Staats, C., Jamnik, M., & Szegedy, C. (2022). Autoformalization with Large Language Models: <http://arxiv.org/abs/2205.12615>.
6. Dubey, A. et al. (2024). The Llama 3 Herd of Models: arxiv.org/abs/2407.21783.
7. Trinh, Trieu, Yuhuai Wu, Quoc Le, He He, and Thang Luong. 2024. Solving Olympiad Geometry Without Human Demonstrations: <https://doi.org/10.1038/s41586-023-06747-5>.
8. Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models: <https://arxiv.org/abs/2201.11903>.
9. Victor M. Glushkov. Problems in the theory of automata and artificial intelligence. *Journal of Cybernetics*, 1(1):97–113, 1971.
10. Victor M. Glushkov. The SAD automated proving system: A brief informal presentation. *Automated Processing of Mathematical Texts*, pages 3–30, 1980.
11. Konstantin Verchinine, Alexander Lyaletski, and Andrei Paskevich. System for automated deduction (SAD): A tool for proof verification. In Frank Pfenning, editor, *Automated Deduction – CADE-21*, pages 398–403, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
12. Konstantin Verchinine, Alexander Lyaletski, Andrei Paskevich, and Anatoly Anisimov. On correctness of mathematical texts from a logical and practical point of view. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics*, pages 583–598, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
13. Lawrence C. Paulson. 1986. Natural deduction as higher-order resolution. *The Journal of Logic Programming*, 3(3):237–258.
14. Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. 2015. The lean theorem prover (system description). In *Automated Deduction - CADE-25*, pages 378–388, Cham. Springer International Publishing.
15. Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., and others (2024). Gemini: A Family of Highly Capable Multimodal Models: <http://arxiv.org/abs/2312.11805>.
16. Chengwu Liu, Jianhao Shen, Huajian Xin, Zhengying Liu, Ye Yuan, Haiming Wang, Wei Ju, Chuanyang Zheng, Yichun Yin, Lin Li, Ming Zhang, and Qun Liu. 2023. Fimo: A challenge formal dataset for automated theorem proving: <https://arxiv.org/abs/2309.04295>.

17. Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models: <https://arxiv.org/abs/2410.05229>.