AI における自動証明について[*1]

作成日 2025 年 3 月 1 日
最終更新 2025 年 3 月 26 日
理化学研究所　園田翔[*2]

1　雑談

本稿執筆中に OpenAI の「ChatGPT (o3-mini-high) 」が東京大学の 2025 年度入試問題の理系数学を 5.5 問 (全 6 問) 解いたというポストが飛び込んできた [*3] [*4].

> 以下の問に答えてください．日本の大学入試の問題で，TeX 形式で与えられます．可能な限り慎重に，じっくり，よくよく考えて，くれぐれもミスや勘違いをしないようにお願いします．

という巧妙なプロンプトに続けて問題文を 1 問ずつ TeX 形式で入力したところ，計 9 分 59 秒ですべての解答が出力されたそうである．ポストに共有されていた ChatGPT のログを見れば，模範解答のような解答が書き連ねてあった．もとの問題文に図形は描かれていないが，図形問題にはまるで図形が見えているかのように解答している様子を目の当たりにして，器用なものだと感心した．5.5 問正答という成績は，公式の採点結果ではないが，上位合格者のレベルに匹敵する．「東ロボくん」が 2016 年に終了してから 8 年が経ち，我々は気づかぬうちに，東ロボくんが超えられなかった技術的障壁の向こう側に来ていたのである．

もう一つ，本稿執筆中に ChatGPT の調査機能「Deep Research」も使えるようになった．早速拙いプロンプトを駆使して「AI における自動証明」について調査させたところ，ものの 10 分程度で，熱心な学生が一晩粘ったような力作の記事が現れた．この体験にはクラっとした．計算機を意味する「computer」という言葉は，20 世紀前半までは文字通り「計算する人間」のことだったと聞くが，人間の研究者を指す「researcher」という言葉も，21 世紀後半には「研究機」と訳されるようになるのであろうか．研究機が超人の速さで次々と論文を書き，人間の研究者は「研究機の書いた内容に責任を持ちます」と言ってハンコをつくだけになるような未来を想像した．

あまりにも良い出来なのでそのまま草稿を準備させようと思い，プロンプトを試行錯誤しているうちに，50 ページ程度の資料ができていた．しかし結局，草稿まで書かせるのは**まだ**時期尚早と判断して諦めた．生成された文章のファクトチェックに時間がかかりすぎたためである．Deep Research の調査能力は素晴らしいが，出てきた原稿をよく見てみると，嘘をついている箇所が散見された．あまりにも理路整然としているので，注意深く読まないと見過ごしてしまいそうになる．一文ずつファクトチェックをしていくと，10 分で出てきた文章の校正に何時間もかかる．これは大変疲れる．どうやら，数値や固有名詞の “写し間違い” や，ロジックを要約する際に大胆な “記憶違い” をすることがまだ多い．ニュース記事やまとめ記事のような二次文献を出典としていることもあり，自社 (OpenAI) と他社の違いを強調するような記述もある．もとの一次文献と突き合わてファクトチェックをするのであれば，はじめからもとの文献を当たるほうが速く正確で情報量も多い．こういうわけで，正確な文章を書くという仕事も，もうしばらくはなくなりそうにないことを体感した．もっと

[*1] 最終版（v2.5）に cite を復活させたもの
[*2] sho.sonoda@riken.jp
[*3] @kaitou_ryaku，2025 年 2 月 26 日午前 3:25．https://x.com/kaitou_ryaku/status/1894453591735505353
[*4] 本節で言及したデータは筆者の GitHub リポジトリにて公開している．https://github.com/shosonoda/susemi2025aitp/

も，文献を突き合わせるような機械的なファクトチェックの方法は一日も速く自動化して，我々の仕事を奪ってほしいと願っている．

さて，本稿では，最近の定理証明 AI をフォローするために必要となる背景知識や機械学習技術について解説する．定理証明 AI に関する文献は 2019 年頃から矢継ぎ早に発表されており，個別の文献情報はたちどころに古くなるため，本稿ではあまり踏み込まない．オンラインには「AI による自動証明と数学的推論」について活発にメンテナンスされている論文リストが複数存在する．例えば調査論文 (Lu et al., 2023; Li et al., 2024; Yang et al., 2024a; Kumar et al., 2025; Li et al., 2025) および連携する GitHub リポジトリを参照されたい．

2 定理証明 AI の変遷

AI による**自動定理証明 (Automated Theorem Proving, ATP)** とは，数学や論理の問題をコンピュータが自動的に証明することを目指す分野である．自動定理証明の研究は計算機科学の黎明期から存在し，初期には論理式の形式操作による証明探索が中心であった．1956 年に Newell らは「Logic Theorist」(Newell and Simon, 1956) を発表し，Whitehead と Russell の『プリンキピア・マテマティカ』に示された定理を証明して，AI における推論システムの可能性を示した．1965 年には Robinson による「一階述語論理の導出原理」(Robinson, 1965) が登場し，論理推論をコンピュータで自動化する基盤が整った．1980–90 年代には人手で定義した推論規則と検索アルゴリズムを組み合わせた自動定理証明器 (「Otter」，「E」，「Vampire」など) が発達した．ATP と並行して，証明支援系 (「Mizar」,「Isabelle」,「Rocq」[*5]，「Lean」など) を用いた**対話型定理証明 (Interactive Theorem Proving, ITP)** も発達した．これは人間が**証明戦略 (tactic)** や**前提 (premise)** を提示し，コンピュータが機械的に証明を検証するシステムである．

2010 年代後半になると，深層学習による画像生成やゲーム AI が飛躍的進歩を遂げたことを受け，数学分野への機械学習応用が注目されるようになった．特に,「AlphaGo」を成功に導いた**モンテカルロ木探索 (Monte Carlo Tree Search, MCTS)** や,「Transformer」に始まる**大規模言語モデル (Large Language Model, LLM)** は，定理証明 AI の研究を強く動機付けた．

LLM 以前の機械学習を用いる先駆的な結果として，**前提選択 (premise selection)** に対する「TacticToe」(Gauthier et al., 2017, 2021) や「DeepMath」(Irving et al., 2016), 強化学習による**証明戦略予測 (tactic prediction)** や**証明探索 (proof search)** に対する「GamePad」(Huang et al., 2019) や「HOList」(Bansal et al., 2019) が挙げられる．

一方,LLM を用いた定理証明 AI の先駆けは,2019 年に登場した OpenAI の「**GPT-f**」(Polu and Sutskever, 2020) である．GPT-f は「Metamath」で形式化された問題に対してそれまで知られていなかった新しい証明を生成し，発見された証明は Metamath の定理ライブラリに採択された．2022 年に Meta が発表した「**Hyper Tree Proof Search (HTPS)**」(Lample et al., 2022) は AlphaZero 流の MCTS を用いて Lean や Metamath における定理証明性能を大きく向上させた．また同年，Google は「**Minerva**」(Lewkowycz et al., 2022) を発表し，自然言語で記述された文章題形式の数学問題において当時最高性能を達成した．Minerva のように自然言語を用いた**数学的推論 (mathematical reasoning)** は，LLM の登場によって初めて実用レベルになったタスクである．それまでは専ら，「TPTP」や Isabelle，Lean のような専用の形式言語で記述された問題が対象であった．LLM による定理証明 AI はその後も次々と登場し，現在に至っている．

[*5] Coq Proof Assistant は Rocq Prover に改名した (2025 年 3 月 12 日完了宣言).

3　定理証明 AI における機械学習技術

一般に LLM の学習は，大規模コーパスにおける次トークン予測を行う**事前学習 (pre-training)** と，目的のタスクに応じた学習を行う**事後学習 (post-training)** の 2 段階に分けられる．さらに，事後学習は訓練時 (training-time) と推論時 (inference-time) またはテスト時 (test-time) [*6] の 2 段階に細分される．訓練時の事後学習とは**教師あり微調整 (supervised fine-tuning, SFT)** と**強化学習 (reinforcement learning, RL)** のことである．一方，推論時における事後学習とは**推論時計算 (inference-time compute)** または**テスト時計算 (test-time -)** と呼ばれる，パラメータを更新せずに計算効率や探索性能を改善させる技術の総称である．以下では事後学習について説明する．

3.1　人間のフィードバックを用いた強化学習 (RLHF)

ChatGPT の成功で知られる「**Reinforcement Learning from Human Feedback (RLHF)**」は，人間の**好み (選好・嗜好，preference)** に対して LLM を**整合 (alignment)** させる手法である．定理証明においては，例えば「証明が論理的に一貫しているか」「読みやすいか」「解答様式に沿っているか」など，人間が好む性質を学習させる目的で RLHF が利用できる．

以下では RLHF の基本形として「Instruct GPT」(Ouyang et al., 2022) の例を説明する．

- **Step 1．教師あり微調整 (SFT)**：事前学習済みの LLM をベースモデルとして，人間が用意した高品質な回答例を用いた教師あり学習を行う．微調整後の LLM を SFT モデルと呼ぶ．定理証明 AI の場合，SFT は AI に基本的な論理推論パターンや定理ライブラリの知識を与えるステップであり，この段階での性能がその後の強化学習の効果を左右する．例えば GPT-f では，Metamath の膨大な形式証明データで GPT-3 を訓練し，文法的に正しい証明文を生成できるようにした．また「DeepSeek-Prover-V1.5」(Xin et al., 2024, 2025) では，DeepSeekMath-Base という事前学習モデルに対し，蓄積された証明データを教師あり学習した．
- **Step 2．報酬モデルの学習 (Reward Model Training, RMT)**：入力プロンプト x に対する出力 y の好ましさ (選好) を数値化する関数 (報酬モデル) $r(x, y)$ を学習する．まず，あるプロンプト x に対する応答を複数 (例えば y_1, y_2) 生成し，人間の評価者が選好順位 (例えば $y_1 \succ y_2$) を付与する．次に，得られた選好データから報酬モデルを学習する．具体的には，選好と報酬の関係を「**Bradley-Terry モデル**」に基づいて定式化し，報酬モデルを最尤推定する．Bradley-Terry モデルは勝敗データから選手のスコアを推定するために用いられるモデルである．2 つの出力 y_1, y_2 について「y_1 が好ましい」という確率を

$$P_r(y_1 \succ y_2 \mid x) := \frac{\exp r(x, y_1)}{\exp r(x, y_1) + \exp r(x, y_2)} = \sigma(r(x, y_1) - r(x, y_2))$$

  という r の関数として定式化し，尤度最大化によって r を調整する．ここで $\sigma(r) := \dfrac{1}{1 + \exp(-r)}$ はシグモイド関数である．
- **Step 3．強化学習 (RL)**：Step 2 で得られた報酬モデル r を用いて，Step 1 で得られた SFT モデルをさらに訓練する．これは入力プロンプト x を状態，出力 y を行動，報酬モデル r を報酬，SFT モ

[*6] 学習フェーズ (learning phase) と推論フェーズ (inference phase) ともいう．

デルを方策 $\pi(y \mid x)$ の初期値とする強化学習である．学習には「**近接方策最適化 (Proximal Policy Optimization, PPO)**」を用いる．この際，初期方策 π_0 との KL ダイバージェンス $KL(\pi_0 \| \pi)$ を罰則項に加え，方策モデルが急激に変化しないよう制御する．

3.2 RLHF の拡張

定理証明は遅延報酬型の問題であり，証明の途中における報酬や選好順位付けは困難である．このため，RLHF の代替や拡張として以下のような手法が提案されている．

- **直接選好最適化 (Direct Preference Optimization, DPO)**(Rafailov et al., 2023)：強化学習 (Step 3) を明示的に行わずに人間の選好データから方策を直接学習する方法である．“RLHF without RL” とも呼ばれる．Step 3 の (KL 正則化付き期待報酬最大化) 問題の解は

$$\pi(y \mid x) = \frac{1}{Z(x)} \pi_0(y \mid x) \exp\left(\frac{r(x,y)}{\beta}\right)$$

という閉形式で書ける．これを r について解き，Step 2 の BT モデルに代入して報酬 r を消去すると，方策 π を変関数とする BT モデルが得られる．π に関して BT モデルの対数尤度最大化を行うことで，報酬学習や KL の調整を経由せず，直接的に，選好データに整合した方策が得られる．なお，報酬モデルは方策から得られる．DPO を発端として多数の “xPO” が開発された．
- **過程報酬モデル (Process Reward Model, PRM)**(Uesato et al., 2022; Lightman et al., 2024)：証明の途中ステップに対して報酬が与えられるようにした報酬モデルを**過程報酬モデル (Process -, PRM)** と呼ぶ．（これに対し，最終的な解答 (証明ステップの列) の正しさに対して報酬が与えられるような報酬モデルを**結果報酬モデル (Outcome -, ORM)** と呼ぶ．）PRM を構築するには，証明の途中ステップに対して適当なスコアを付与した訓練データが必要である．しかし，このスコアを人手で付与（アノテーション）するのは難しい．「Math-Shepherd」(Wang et al., 2024) では，MCTS を利用して，人手を介さずに PRM を自動構築できるようにした．
- **群相対方策最適化 (Group Relative Policy Optimization, GRPO)**(Shao et al., 2024)：DeepSeek が開発した PPO の確率的近似アルゴリズム．同社の「DeepSeek-R1」でも重要技術として採用された．通常の PPO では行動価値 Q と状態価値 V を別に用意して，相対的な価値 (advantage)$A(y,x) = Q(y,x) - V(x)$ を計算する．GRPO では，一組の出力をまとめて評価し，A をモンテカルロ近似することにより，価値モデルを省略した．また，GRPO では報酬モデルを使わずに，一群の生成結果に対して例えば「証明が完了したか」「形式エラーがないか」などルールベースの評価関数でスコアを与え，それらの相対差分のみで勾配を計算することもできる．

3.3 推論時スケーリング (TTS)

推論時スケーリング (Test-Time Scaling, TTS) は，推論時の計算 (test-time compute) に追加の計算資源や計算上の工夫を投入し，推論性能を向上 (スケール) させる技術の総称である．例えば，“Let’s think step by step” というフレーズで一躍有名となった「**思考連鎖 (Chain-of-Thought, CoT)**」をはじめ，AlphaGo で脚光を浴びた**モンテカルロ木探索 (MCTS)** などのサンプリング・探索技術，「**検索拡張生成 (Retrieval-Augmented Generation(RAG)**」などの検索技術，さらには「**低ランク適応 (Low-Rank**

Adaptation, LoRA)」などの軽量化技術も TTS に分類される．ChatGPT の**推論モデル (o1, o3)** は TTS を効果的に活用しているとされ，従来のモデルサイズ・データサイズ・学習時間に関するスケーリング則 (言わば，訓練時スケーリング) に代わる新しいスケーリング概念として 2024 年頃から注目を集めている (Snell et al., 2025)．TTS は大きく並列スケーリングと逐次スケーリングに分けられる．並列とは，Best-of-N に代表される繰り返しサンプリング・探索技術である．すなわち，複数の解を同時に生成し，その中から最良の解を選ぶ手法である．一方，逐次とは，思考連鎖 (CoT) に代表される逐次的な推論技術である．数学の定理証明は論理的思考の多ステップ推論が要求される難問であり，TTS 手法の導入が特に有効である．モデルサイズを増やすことなく推論プロセスを工夫して高性能化できるため，近年の定理証明 AI ではさまざまな TTS 手法が活用されている．

- **モンテカルロ木探索 (MCTS)**：MCTS は確率的に効率よく木を探索するアルゴリズムである．各ノードで方策に沿ったロールアウトを繰り返してノードの価値を推定し，逐次的に良さそうな枝を深く掘り下げる．計算コストはロールアウト回数 × 深さと非常に高いが，探索効率は高く，証明のように大きい探索空間でも有望手順に確率集中できる．前述の通り，HTPS は，AlphaZero にヒントを得た HyperTree 探索アルゴリズムを導入し，Metamath における定理証明の成功率を飛躍的に向上させた．DeepSeek-Prover-V1.5 では，RMaxTS という独自の MCTS を導入し，内在的報酬による多様な証明経路探索で性能向上を図っている．「**rStar-Math**」(Guan et al., 2025) では，7B 規模の**小規模言語モデル (Small -, SLM)** [*7] に対して，自然言語による推論と Python コードによる計算を組み合わせたステップを MCTS で展開する方法で成功率を飛躍的に向上させることに成功した．
- **検索拡張生成 (RAG)**：RAG とは，LLM の埋め込みベクトルを用いたベクトル検索のことである．定理証明においては，証明すべき定理に関連する既知の定理や補題を検索するために利用される．「**LeanDojo**」(Yang et al., 2023) では，「ByT5」をエンコーダとして Lean の証明ライブラリ「Mathlib」をベクトル化し，目的の定理に関連する過去の定理や証明をベクトル検索することで，次の証明ステップを提案できるようにした．
- **外部装置の統合**：「**Thor**」(Jiang et al., 2022) では，LLM と Isabelle の「Sledgehammer」(複数の ATP による補題探索ツール) を連携させることで，LLM の生成力に伝統的検索アルゴリズムの網羅力を統合した．「**Qwen2.5-Math**」(Yang et al., 2024b) では，CoT による段階的解法に加えて，外部計算ツールとの連携機能 (**Tool-Integrated Reasoning, TIR**) を持つ．例えば，計算や方程式を解くために Python を呼び出して正確に処理できる．計算コストはツール実行分増えるが，計算ミスなどを防げる利点があり，DeepSeek と比肩する性能を示した．

4　技術的観点の整理

定理証明 AI の技術的観点を比較整理する．

- **訓練時スケーリングと推論時スケーリング**：大規模モデルは一般に知識量とパターン認識能力が高く，GPT-3 (175B) や DeepSeek-V3 (671B) のようにスケールで殴る戦略は高い性能をもたらしてきた．一方，rStar-Math (7B) はそれだけでは解けない問題に対して，MCTS と自己検証を組み合わせたア

[*7] LLM のパラメータサイズは B(billion = 10 億) または M(million = 100 万) で数える．GPT-3 (175B) や DeepSeek-V3 (671B) と比較して，7B は小規模である．

プローチが有効であることを示した．小規模モデルであっても推論時計算量を増やせば，結果的に大規模モデル並みの網羅性が持てることが分かってきた．とはいえ，伝統的なスケーリング則もまだ健在である．例えば形式言語を扱う「**Goedel-Prover**」(Lin et al., 2025) は小規模モデル (7B) + 大量の合成データを用いた段階学習によって，高度な推論を行うことなく，DeepSeek-Prover-V1.5 (7B) よりも高い成績を収めることに成功した．

- **形式言語と自然言語**：形式言語による証明は，証明支援系を用いて証明の正しさを機械的に検査できるという強みがある一方，証明ステップの生成が難しいという弱点がある．これに対し，自然言語による証明は，証明の正しさを人手によって確認する必要があり，全体のボトルネックとなるという弱点がある一方，例えば ChatGPT の推論モデルでは大学レベルの証明問題にも比較的正しく解答できるようになってきており，証明生成が相対的に容易という意外な強みが明らかとなりつつある．
- **単独型と複合型**：一般に単独の LLM は代数的な計算が苦手である．Python を呼び出して検算をする rStar-Math のように，外部装置と連携する複合型 LLM は合理的である．既に「Toolformer」のように計算機や定理データベースへのアクセスを組み込む仕組みがある．特に定理証明においては形式検証器との連携が有効である．Thor や LeanDojo のように，LLM の提案を ATP や証明支援系がチェックする構造は，誤答を原理的に排除できる強みがある．一方，対話の頻度が増えると探索効率が落ちるため，Goedel-Prover のように単独で正しい証明を出力する能力も必要である．
- **逐次生成と一括生成**：HTPS や，LeanDojo,「**Self-Taught Reasoner (STaR)**」(Zelikman et al., 2022) のように証明をステップ毎に構築する逐次生成方式は, 途中で間違えても巻き戻したり方針転換したりできる一方, 探索コストが高いという欠点がある．これに対し DeepSeek-Prover や Goedel-Prover のような一括生成方式は，一度の生成で失敗すればアウトだが，LLM の文脈保持力を最大限に活かせる利点がある．「**Draft, Sketch, and Prove (DSP)**」(Jiang et al., 2023) は両者のハイブリッド方式といえる．DSP では，非形式的な証明 (draft) から形式的証明の骨子 (sketch) を自動抽出し，それを Isabelle に入力して証明を自動探索する (prove)．
- **分かりやすさと厳密さ**：生成された解答の分かりやすさと論理的飛躍の少なさはいずれも重要である．しかし，単なる RLHF では分かりやすさを優先して数学的厳密さを損なう危険性がある．Math-Shepherd のように人ではなく論理規則に基づいて報酬を設計する仕組みや，「Constitutional AI」のようにあらかじめ「数学では厳密性を優先する」といった原則を定める方法が模索されている．
- **オープン開発とクローズド開発**：2025 年 3 月現在，ChatGPT のようなクローズドソースモデルはオープンソースモデルに先行している．しかし同時に，Llemma や Goedel-Prover などのオープンソースモデルが次々と生まれ，DeepSeek や Qwen のような企業オープンモデルも登場している．オープンソースの強みはコミュニティによる機動的な検証可能性や連携可能性である．特に数学分野は研究者コミュニティが大きいため，オープンモデルの性能向上サイクルが今後も回り続けると期待できる．

5　まとめ

本稿では，2025 年 3 月時点における定理証明 AI をフォローするための知識を概観した．紙数の都合上，ごく簡単な説明に終始することとなったが，興味のある読者は参考文献から詳細を辿ってほしい．(冒頭触れた筆者の GitHub でも未完の詳細版を公開している．)　改めて俯瞰してみると，AI による自動証明は，複数の分野——機械学習・言語処理・計算機科学・数理論理学・数学など——を繋ぐ境界領域の問題として，古くから

人々を魅了し続けてきた．

参考文献を眺めれば一目瞭然だが，LLM を用いた定理証明 AI の研究は国外の研究グループがリードしており，日本の影は薄い．論文の著者欄を見ると，LLM 以前の定理証明 AI 技術や競技数学の技術を知っている人たちと，LLM を訓練する技術を備えた人たちが上手く協力する体制を築いている．筆者が知る限りにおいても，日本にはそれぞれの専門分野において深い理解と技術力を持った人たちがいる．今後，日本国内から本格的な定理証明 AI が登場することは十分にあり得ると考えている．

定理証明 AI は既に人間を越えたように思われるかもしれないが，今のところは (まだ) 数学オリンピックや大学の数学のように「正解」もあり「教科書」もあるよく整備された数学において，人間よりも素早く解答できるようになってきたという段階である．最先端の数学の未解決問題を解いたとか，その過程で新しい数学を創ったという段階には至っていない．もし万が一ここで AI の発展が止まれば，LLM による定理証明 AI の正体は，人類が産出した知識を素早く検索する新手の検索アルゴリズムだったということになる．AI は真に新しい知識を創れるようになるのだろうか．筆者としては，数学や機械学習だけでなく，数理論理学や計算機科学の教科書をも書き換えるような，21 世紀を代表する大発見に化けることを期待している．

参考文献

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. A Survey of Deep Learning for Mathematical Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

Zhaoyu Li, Jialiang Sun, Logan Murphy, Qidong Su, Zenan Li, Xian Zhang, Kaiyu Yang, and Xujie Si. A Survey on Deep Learning for Theorem Proving. In *First Conference on Language Modeling*, 2024.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. Formal Mathematical Reasoning: A New Frontier in AI. *arXiv preprint: 2412.16075*, 2024a.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H S Torr, Salman Khan, and Fahad Shahbaz Khan. LLM Post-Training: A Deep Dive into Reasoning Large Language Models. *arXiv preprint: 2502.21321*, 2025.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. From System 1 to System 2: A Survey of Reasoning Large Language Models. *arXiv preprint: 2502.17419*, 2025.

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

J A Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of ACM*, 12(1): 23–41, 1965.

Thibault Gauthier, Cezary Kaliszyk, and Josef Urban. TacticToe: Learning to Reason with HOL4 Tactics. In *LPAR-21: 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 46, pages 125–143. EasyChair, 2017.

Thibault Gauthier, Cezary Kaliszyk, Josef Urban, Ramana Kumar, and Michael Norrish. TacticToe: Learning to Prove with Tactics. *J. Autom. Reason.*, 65(2):257–286, 2021.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Een, Francois Chollet, and Josef Ur-

ban. DeepMath - Deep Sequence Models for Premise Selection. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Daniel Huang, Prafulla Dhariwal, Dawn Song, and Ilya Sutskever. GamePad: A Learning Environment for Theorem Proving. In *International Conference on Learning Representations*, 2019.

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. HOList: An Environment for Machine Learning of Higher Order Logic Theorem Proving. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 454–463. PMLR, 2019.

Stanislas Polu and Ilya Sutskever. Generative Language Modeling for Automated Theorem Proving. *arXiv preprint: 2009.03393*, 2020.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. HyperTree Proof Search for Neural Theorem Proving. In *Advances in Neural Information Processing Systems*, volume 35, pages 26337–26349, 2022.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems with Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. DeepSeek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data. *arXiv preprint: 2405.14333*, 2024.

Huajian Xin, Z Z Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z F Wu, Fuli Luo, and Chong Ruan. Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. In *2nd MATH-AI Workshop at NeurIPS'22*, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, 2024.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui.

Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y K Li, Y Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint: 2402.03300*, 2024.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. *arXiv preprint: 2501.04519*, 2025.

Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Albert Qiaochu Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Miłoś, Yuhuai Wu, and Mateja Jamnik. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. In *Advances in Neural Information Processing Systems*, volume 35, pages 8360–8373, 2022.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv preprint: 2409.12122*, 2024b.

Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-Prover: A Frontier Model for Open-Source Automated Theorem Proving. *arXiv preprint: 2502.07640*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Self-Taught Reasoner Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022.

Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Timothee Lacroix, Jiacheng Liu, Wenda Li, Mateja Jamnik, Guillaume Lample, and Yuhuai Wu. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. In *The Eleventh International Conference on Learning Representations*, 2023.