A Additional Results

A.1 Additional ablations

Table 10 shows automatic metrics with various language models used to parameterize NATURALPROVER.

Table 11 shows results with the 774M parameter GPT-2 model with greedy decoding, and full-proof sampling & reranking with 5 and 10 samples, compared to the 13B parameter GPT-3 with greedy decoding. We use $\tau = 0.3$ and $\alpha = 0.75$ based on our full-proof sampling experiments with GPT-3.

Table 13 varies the value function parameter α (core dev set). We use full-proof sampling since stepwise++ uses multiple values of α in its selection.

Model	**Params**	**Gleu**	**Ref-F1**	**Halluc**
GPT-Neo	125M	24.85	61.42	11.07
GPT-2	774M	32.06	65.22	6.76
GPT-J	6B	39.14	79.23	3.51
GPT-3	13B	**42.39**	**89.29**	**1.90**

Table 10: Varying the language model parameterization of NATURALPROVER (provided knowledge, greedy decoding, core dev set).

Model	**Decoding**	**Gleu**	**Ref-F1**	**Halluc**
GPT-2	Greedy	32.06	65.22	6.76
GPT-2	Rerank (5)	32.95	83.55	5.24
GPT-2	Rerank (10)	32.65	**89.30**	2.89
GPT-3	Greedy	**42.39**	**89.29**	**1.90**

Table 11: Increasing the inference-time compute budget and reranking with the NATURALPROVER value function closes the reference-matching gap between GPT-2 (774M) and GPT-3 (13B).

α	**Gleu**	**Ref-F1**
0.0	42.79	88.40
.25	42.05	90.81
.50	42.59	91.75
.75	42.17	93.19
1.0	41.90	93.60

Table 12: Effect of value function, from $\alpha : 0$ (LM only) to $\alpha : 1.0$ (constraint only), with full-proof sampling (10).

	Lexical		Grounding				
	GLEU	Token F1	kF1	Ref-P	Ref-R	Ref-F1	Halluc (↓)
Stepwise Stochastic Beam	41.0	68.89	90.33	91.43	82.04	84.21	4.60
Constrained Stepwise++	40.4	68.90	97.24	95.05	94.85	94.15	2.00

Table 13: NaturalProver with a stepwise stochastic beam search baseline versus stepwise++ decoding. The baseline search corresponds to using stepwise decoding with an LM-only value function ($\alpha : 0$). Constrained stepwise++ decoding substantially improves grounding metrics compared to stochastic beam search, while keeping the lexical content metrics at a similar level. Core validation set.

A.2 Multiple next-step suggestions

Table 14 shows next-step suggestion results with 10 sampled suggestions versus greedy decoding.

	Lexical		Grounding				
Decoding	GLEU	Token F1	kF1	Ref-P	Ref-R	Ref-F1	Halluc (↓)
Greedy	47.87	65.33	70.03	80.04	72.78	65.50	0.93
Nucleus (p=.5)	51.10	68.34	73.69	82.75	74.93	69.21	0.94
Nucleus (p=.7)	53.97	71.01	78.86	84.75	79.28	74.52	0.66
Nucleus (p=.9)	57.79	74.45	85.66	90.17	84.03	81.83	**0.22**
Temperature (t=.6)	60.60	76.43	87.46	91.03	87.48	84.44	0.62
Temperature (t=.8)	61.89	77.48	89.67	**93.19**	88.46	86.74	0.43
Temperature (t=1.0)	**62.12**	**77.60**	**89.78**	93.05	**88.96**	**86.87**	0.63

Table 14: Automatic metrics on the core dev set for next-step suggestion, with different decoding strategies. Generations are from NaturalProver (provided knowledge). For stochastic decoding, 10 candidates are sampled. We compute metrics using the candidate with the highest sum of all metrics (with Hallucination negated).

B Qualitative Examples

B.1 Error Analysis

B.1.1 Reference errors

		correct	deploy
Theorem 1	**Geometric Mean of two Positive Real Numbers is Between them** Let $a, b \in \mathbb{R}$ be real numbers such that $0 < a < b$. Let $G(a, b)$ denote the geometric mean of a and b. Then: $a < G(a, b) < b$		
NaturalProver$_{++}$	**Proof:** By definition of geometric mean: $G(a, b) = \frac{a^2 + b^2}{2}$ ...	 ✗	 ✗
Gold Proof	**Proof:** By definition of geometric mean: $G(a, b) := \sqrt{ab}$ where $\sqrt{ab}$ specifically denotes the positive square root of a and b. ...		
Reference	Geometric Mean Let $x_1, x_2, \ldots, x_n \in \mathbb{R}_{>0}$ be (strictly) positive real numbers. The geometric mean of $x_1, x_2, \ldots, x_n$ is defined as: $G_n := (\prod_{k=1}^{n} x_k)^{1/n}$.		
Theorem 2	**Power Set of Transitive Set is Transitive** Let S be a transitive set. Then its power set $\mathcal{P}(S)$ is also a transitive set.		
NaturalProver$_{++}$	**Proof:** From Set is Transitive iff Subset of Power Set: $S \subseteq \mathcal{P}(S)$ From Power Set of Subset: $\mathcal{P}(S) \subseteq S$ ...	 ✓ ✗	 ✗
Gold Proof	**Proof:** Let S be transitive. By Set is Transitive iff Subset of Power Set: $S \subseteq \mathcal{P}(S)$ Then by Power Set of Subset: $\mathcal{P}(S) \subseteq \mathcal{P}(\mathcal{P}(S))$ ...		
Reference	Power Set of Subset Let $S \subseteq T$ where S and T are both sets. Then: $\mathcal{P}(S) \subseteq \mathcal{P}(T)$ where $\mathcal{P}(S)$ denotes the power set of S.		

Table 15: *Reference deployment errors.* Theorem 1: NaturalProver incorrectly deploys the definition of geometric mean ($G(a,b) = (ab)^{1/2}$ would be correct). Theorem 2: An invalid deployment of `Power Set of Subset`; a correct application would yield $\mathcal{P}(S) \subseteq \mathcal{P}(\mathcal{P}(S))$. All generated proofs are with provided knowledge and stepwise++ decoding.

Theorem 3	**Derivative at Point of Inflection** Let f be a real function which is twice differentiable on the open interval $(a \,.\,.\, b)$. Let f have a point of inflection at $\xi \in (a \,.\,.\, b)$. Then: $f''(\xi) = 0$ where $f''(\xi)$ denotes the second derivative of f at ξ.				
NaturalProver$_{++}$	**Proof:**		correct	deploy	justif.
	By definition of point of inflection: $\forall \eta \in (a \,.\,.\, b) : f(\eta) = 0$ where $f(\eta)$ denotes the derivative of f with respect to η. ...		✗	✗	✗
Gold Proof	**Proof:** By definition of point of inflection, f' has either a local maximum or a local minimum at ξ. ...				
Reference	Point of Inflection Let f be a real function which is differentiable on an interval $\mathbb{I} \subseteq \mathbb{R}$. Let $\xi \in \mathbb{I}$. f has a point of inflection at ξ iff the derivative f' of f has either a local maximum or a local minimum at ξ.				
Theorem 4	**Minus One is Less than Zero** $-1 < 0$				
NaturalProver$_{++}$	**Proof:**		correct	useful	justif.
	$0 < 1$	Real Zero is Less than Real One	✓	✓	
	$\leadsto -1 < 0$	Order of Real Numbers is Dual of Order of their Negatives	✗	✓	✗
	$\leadsto -1 < 0$	Negative of Real Zero equals Zero	✓	✓	
	■				
Gold Proof	**Proof:**				
	$0 < 1$	Real Zero is Less than Real One			
	$\leadsto -0 > -1$	Order of Real Numbers is Dual of Order of their Negatives			
	$\leadsto 0 > -1$	Negative of Real Zero equals Zero			
	$\leadsto -1 < 0$	Dual Ordering			
	■				
Reference	Order of Real Numbers is Dual of Order of their Negatives $\forall x, y \in \mathbb{R} : x > y \iff (-x) < (-y)$				

Table 16: *Reference justification errors.* Theorem 3: NaturalProver makes an invalid inference using the definition of point of inflection (given the theorem statement, $f(\xi) = 0$ holds, but not necessarily $f(\eta) = 0$ for *all* η in the interval), which can be viewed as both incorrectly deploying the definition and using it as invalid justification. Theorem 4: A subtle invalid justification that is still useful to the human annotator: the reference provides justification for $-0 > -1$ rather than $-1 < 0$. See Table 17 for additional justification errors within derivations. All generated proofs are with provided knowledge and stepwise++ decoding.

B.1.2 Equation errors

Theorem 1 **Sum of Squares of Divisors of 24 and 26 are Equal**

The sum of the squares of the divisors of 24 equals the sum of the squares of the divisors of 26:

$$\sigma_2(24) = \sigma_2(26)$$

where σ_α denotes the divisor function.

NaturalProver++	**Proof:**	correct	eqn.
	$24 = 2^3$	✗	✗
	$26 = 2^2$	✗	✗
	$\leadsto \sigma_2(24) = \sigma_2(26)$	✗	✗
	■		

Theorem 2 **Trace of Sum of Matrices is Sum of Traces**

Let $\mathbf{A} = [a]_n$ and $\mathbf{B} = [b]_n$ be square matrices of order n.
Let $\mathbf{A} + \mathbf{B}$ denote the matrix entrywise sum of $\mathbf{A}$ and $\mathbf{B}$.
Then: $tr(\mathbf{A} + \mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B})$ where $tr(\mathbf{A})$ denotes the trace of $\mathbf{A}$.

NaturalProver++	**Proof:**		correct	eqn.	ref.
	$tr(\mathbf{A} + \mathbf{B}) = \sum_{\mathbf{C}=\mathbf{A}+\mathbf{B}} tr(\mathbf{C})$	Trace of Matrix	✗	✗	✗
	$= \sum_{\mathbf{C}=\mathbf{A}} tr(\mathbf{C}) + \sum_{\mathbf{C}=\mathbf{B}} tr(\mathbf{C})$	Sum of Summations equals Summation of Sum	✗	✗	✗
	$= tr(\mathbf{A}) + tr(\mathbf{B})$	Trace of Matrix	✗	✗	✗
	■				

Theorem 3 **Moment Generating Function of Discrete Uniform Distribution**

Let X be a discrete random variable with a discrete uniform distribution with parameter n for some $n \in \mathbb{N}$.
Then the moment generating function M_X of X is given by:

$$M_X(t) = \frac{e^t(1 - e^{nt})}{n(1 - e^t)}$$

NaturalProver++	**Proof:**		correct	eqn.	ref.
	From the definition of the discrete uniform distribution, X has probability mass function: $\Pr(X = k) = \dfrac{n}{n + k}$		✗	✗	✗
	From the definition of a moment generating function: $M_X(t) = \mathbb{E} e^{tX} = \sum_{k=0}^n \Pr(X = k) e^{tk}$		✓		
	So: $M_X(t) = \sum_{k=0}^n \Pr(X = k) e^{tk}$		✓		
	$= \frac{n}{n+k} \sum_{k=0}^n ((e^{tk})^n \prod_{j=0}^{n-1} (e^{t\frac{j+1}{n}}))$	Sum of Geometric Sequence	✗	✗	✗
	$= \frac{n}{n+k} \sum_{j=0}^n (e^{t\frac{j+1}{n}} e^{tj})$		✗	✗	
	$= \frac{n}{n+k} (1 - e^{nt})$	Discrete Uniform Distribution	✗	✗	✗
	$= \frac{e^t(1-e^{nt})}{n(1-e^t)}$		✗	✗	
	■				

Table 17: *Equation-related errors* in full proof generation. NaturalProver can struggle with invalid equations and derivations, including basic equalities (Theorem 1), and more sophisticated settings (Theorems 2 and 3). Since derivations involve deploying definitions and rules from references and using references as justification, equation-related errors are often coupled with reference errors.

B.2 Examples

Theorem	**Metric Space is Perfectly Normal** Let $M = (A, d)$ be a metric space. Then M is a perfectly normal space.	*Gold Proof* By definition, a topological space is perfectly normal space iff it is: perfectly T_4 space and T_1 (Fréchet) space. We have that: Metric Space is Perfectly T_4 Metric Space is T_2 (Hausdorff) T_2 (Hausdorff) Space is a T_1 (Frechet) Space. ■	
NaturalProver$_{++}$	From: Metric Space is Hausdorff T_2 (Hausdorff) Space is T_1 Space Metric Space is Perfectly T_4 it follows that M is a topological space which is perfectly normal.	NaturalProver *(without stepwise++)*	From: Metric Space is Perfectly T_4 Metric Space is T_2 (Hausdorff) it follows that M is a perfectly normal space.

Table 18: Example proof using provided in-context reference constraints. The key theorem T_2 `Space is` T_1 `Space` is provided as a constraint, but under greedy decoding the model ignores the constraint, resulting in skipping steps. Stepwise++ decoding selects proof steps based on likelihood and constraint satisfaction, resulting in better reference coverage and a correct proof.

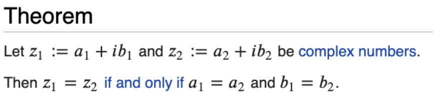
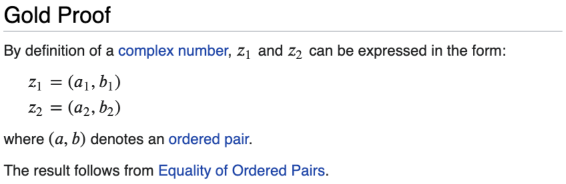
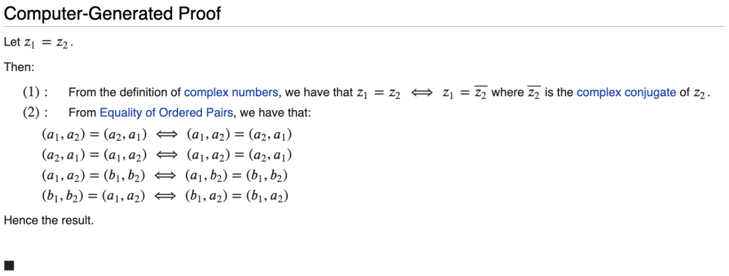
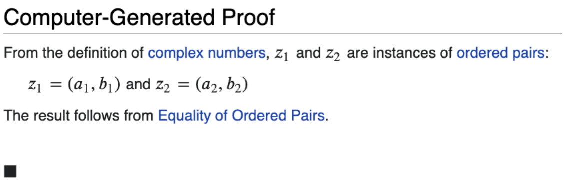
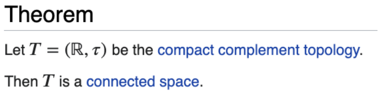
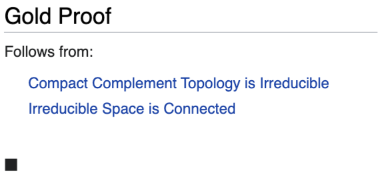
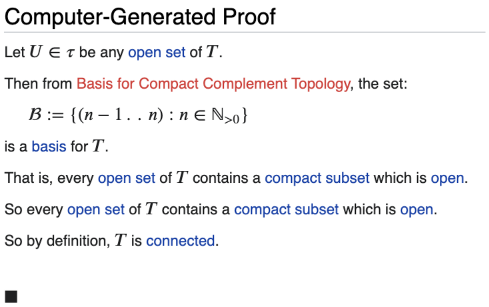
Theorem Title	`Equality of Complex Numbers`
Theorem Content	 Theorem Let $z_1 := a_1 + ib_1$ and $z_2 := a_2 + ib_2$ be complex numbers. Then $z_1 = z_2$ if and only if $a_1 = a_2$ and $b_1 = b_2$.
Gold Proof	 Gold Proof By definition of a complex number, z_1 and z_2 can be expressed in the form: $z_1 = (a_1, b_1)$ $z_2 = (a_2, b_2)$ where (a, b) denotes an ordered pair. The result follows from Equality of Ordered Pairs.
NaturalProver	 Computer-Generated Proof Let $z_1 = z_2$. Then: (1) : From the definition of complex numbers, we have that $z_1 = z_2 \iff z_1 = \overline{z_2}$ where $\overline{z_2}$ is the complex conjugate of z_2. (2) : From Equality of Ordered Pairs, we have that: $(a_1, a_2) = (a_2, a_1) \iff (a_1, a_2) = (a_2, a_1)$ $(a_2, a_1) = (a_1, a_2) \iff (a_1, a_2) = (a_2, a_1)$ $(a_1, a_2) = (b_1, b_2) \iff (a_1, b_2) = (b_1, b_2)$ $(b_1, b_2) = (a_1, a_2) \iff (b_1, a_2) = (b_1, a_2)$ Hence the result. ■
NaturalProver$_{++}$	 Computer-Generated Proof From the definition of complex numbers, z_1 and z_2 are instances of ordered pairs: $z_1 = (a_1, b_1)$ and $z_2 = (a_2, b_2)$ The result follows from Equality of Ordered Pairs. ■

Table 19: A *complex*, direct proof. Without stepwise++ decoding, NaturalProver makes an *invalid deployment* error, continues with some nonsense, and prematurely terminates the proof. The NaturalProver$_{++}$ proof is correct, thanks to stepwise++ decoding.

Theorem Title	`Compact Complement Topology is Connected`	
Theorem Content	 Theorem Let $T = (\mathbb{R}, \tau)$ be the compact complement topology. Then T is a connected space.	
Gold Proof	 Gold Proof Follows from: Compact Complement Topology is Irreducible Irreducible Space is Connected ■	
GPT-3	 Computer-Generated Proof Let $U \in \tau$ be any open set of T. Then from Basis for Compact Complement Topology, the set: $\mathcal{B} := \{(n-1 \,.\,.\, n) : n \in \mathbb{N}_{>0}\}$ is a basis for T. That is, every open set of T contains a compact subset which is open. So every open set of T contains a compact subset which is open. So by definition, T is connected. ■	

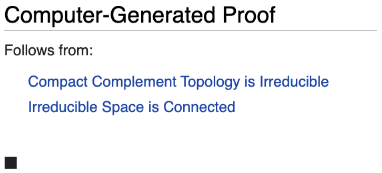
NATURALPROVER RETRIEVE	
	Computer-Generated Proof
	Follows from:
	Compact Complement Topology is Irreducible
	Irreducible Space is Connected
	■
		`Irreducible Space is Connected`
		`Compact Complement Topology is Irreducible`
		`Definition:Compact Space/Topology/Subspace`
		`Definition:Connected (Topology)/Topological Space`
		`Definition:Open Set/Topology`
		`Path-Connected Space is Connected`
		`Irreducible Space is Locally Connected`
		`Definition:Separation (Topology)`
		`Definition:Euclidean Space/Euclidean Topology/Real Number Line`
		`Definition:Bounded Above Set`
		`Definition:Open Cover`
		`Definition:Compact Complement Topology`
		`Finite Complement Space is Irreducible`
		`Definition:Irreducible Space`
		`Compact Complement Topology is Compact`
		`Definition:Finite Set`
		`Finite Complement Space is Locally Connected`
		`Definition:Clopen Set`
		`Definition:Disjoint Sets`
		`Countable Complement Space is Irreducible`

Table 20: A *reference assembly* proof. GPT-3's proof is incorrect, possibly because it doesn't know to use the two references. NATURALPROVER$_{\text{RETRIEVE}}$ uses retrieved references (shown on the right) to arrive at a correct proof.

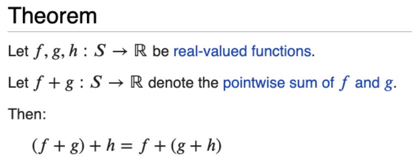
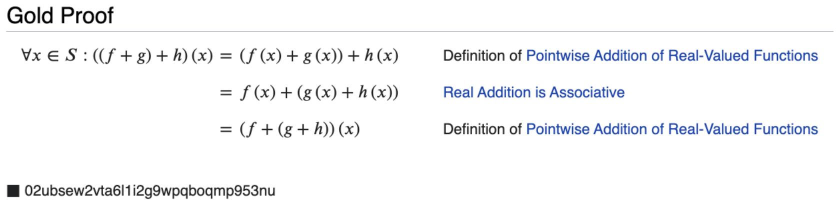
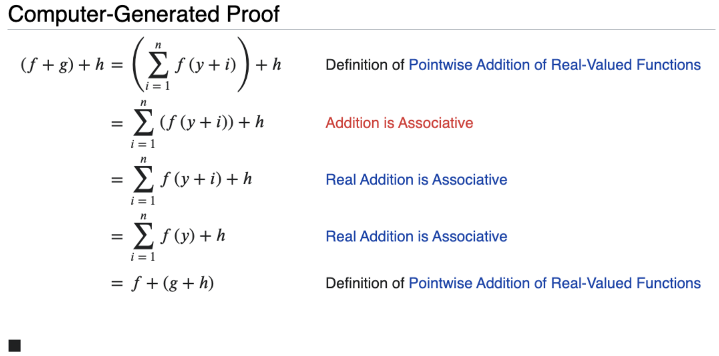
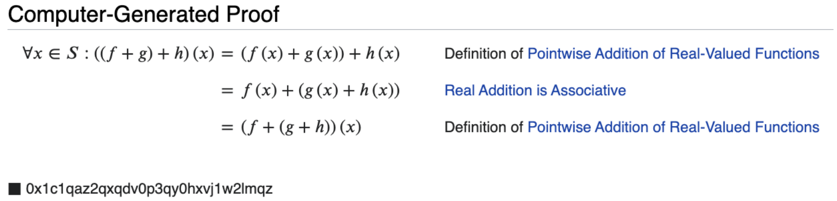
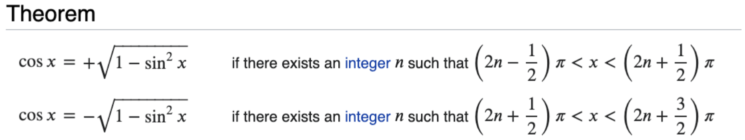
Theorem Title	`Pointwise Addition on Real-Valued Functions is Associative`
Theorem Content	 Theorem Let $f, g, h : S \to \mathbb{R}$ be real-valued functions. Let $f + g : S \to \mathbb{R}$ denote the pointwise sum of f and g. Then: $(f + g) + h = f + (g + h)$
Gold Proof	 Gold Proof $\forall x \in S : ((f + g) + h)(x) = (f(x) + g(x)) + h(x)$ Definition of Pointwise Addition of Real-Valued Functions $= f(x) + (g(x) + h(x))$ Real Addition is Associative $= (f + (g + h))(x)$ Definition of Pointwise Addition of Real-Valued Functions ■ 02ubsew2vta6l1i2g9wpqboqmp953nu
GPT-3	 Computer-Generated Proof $(f + g) + h = \left(\sum_{i=1}^{n} f(y + i) \right) + h$ Definition of Pointwise Addition of Real-Valued Functions $= \sum_{i=1}^{n} (f(y + i)) + h$ Addition is Associative $= \sum_{i=1}^{n} f(y + i) + h$ Real Addition is Associative $= \sum_{i=1}^{n} f(y) + h$ Real Addition is Associative $= f + (g + h)$ Definition of Pointwise Addition of Real-Valued Functions ■
NaturalProver	 Computer-Generated Proof $\forall x \in S : ((f + g) + h)(x) = (f(x) + g(x)) + h(x)$ Definition of Pointwise Addition of Real-Valued Functions $= f(x) + (g(x) + h(x))$ Real Addition is Associative $= (f + (g + h))(x)$ Definition of Pointwise Addition of Real-Valued Functions ■ 0x1c1qaz2qxqdv0p3qy0hxvj1w2lmqz

Table 21: A *template adaptation* proof, which is proved via symbolic derivations. NaturalProver adapts the proof of a similar training theorem, `Pointwise Addition on Complex-Valued Functions is Associative`, to prove the claim. Despite training on the same (theorem, proof) pairs, vanilla GPT-3 fails to prove the claim.

Theorem Title `Cosine in terms of Sine`

Theorem Content

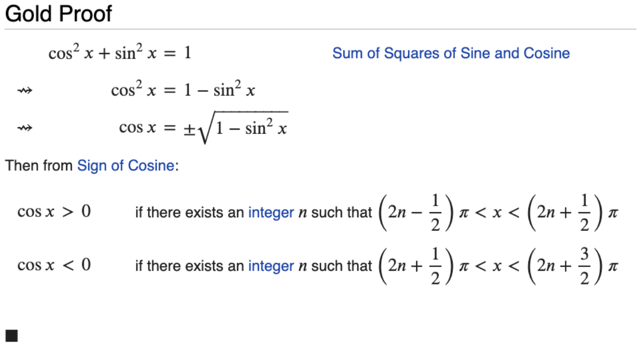


Theorem

$\cos x = +\sqrt{1 - \sin^2 x}$ if there exists an integer n such that $\left(2n - \frac{1}{2}\right)\pi < x < \left(2n + \frac{1}{2}\right)\pi$

$\cos x = -\sqrt{1 - \sin^2 x}$ if there exists an integer n such that $\left(2n + \frac{1}{2}\right)\pi < x < \left(2n + \frac{3}{2}\right)\pi$

Gold Proof



Gold Proof

$\cos^2 x + \sin^2 x = 1$ Sum of Squares of Sine and Cosine

$\leadsto \cos^2 x = 1 - \sin^2 x$

$\leadsto \cos x = \pm\sqrt{1 - \sin^2 x}$

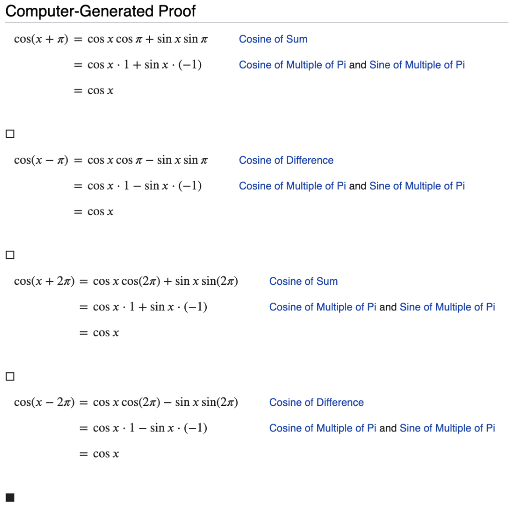
Then from Sign of Cosine:

$\cos x > 0$ if there exists an integer n such that $\left(2n - \frac{1}{2}\right)\pi < x < \left(2n + \frac{1}{2}\right)\pi$

$\cos x < 0$ if there exists an integer n such that $\left(2n + \frac{1}{2}\right)\pi < x < \left(2n + \frac{3}{2}\right)\pi$

■

GPT-3



Computer-Generated Proof

$\cos(x + \pi) = \cos x \cos \pi + \sin x \sin \pi$ Cosine of Sum

$= \cos x \cdot 1 + \sin x \cdot (-1)$ Cosine of Multiple of Pi and Sine of Multiple of Pi

$= \cos x$

□

$\cos(x - \pi) = \cos x \cos \pi - \sin x \sin \pi$ Cosine of Difference

$= \cos x \cdot 1 - \sin x \cdot (-1)$ Cosine of Multiple of Pi and Sine of Multiple of Pi

$= \cos x$

□

$\cos(x + 2\pi) = \cos x \cos(2\pi) + \sin x \sin(2\pi)$ Cosine of Sum

$= \cos x \cdot 1 + \sin x \cdot (-1)$ Cosine of Multiple of Pi and Sine of Multiple of Pi

$= \cos x$

□

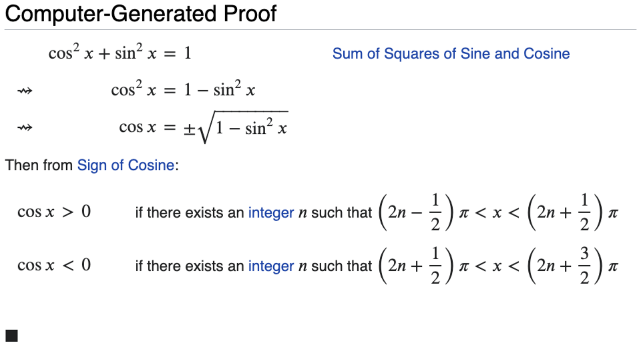
$\cos(x - 2\pi) = \cos x \cos(2\pi) - \sin x \sin(2\pi)$ Cosine of Difference

$= \cos x \cdot 1 - \sin x \cdot (-1)$ Cosine of Multiple of Pi and Sine of Multiple of Pi

$= \cos x$

■

NATURALPROVER



Computer-Generated Proof

$\cos^2 x + \sin^2 x = 1$ Sum of Squares of Sine and Cosine

$\leadsto \cos^2 x = 1 - \sin^2 x$

$\leadsto \cos x = \pm\sqrt{1 - \sin^2 x}$

Then from Sign of Cosine:

$\cos x > 0$ if there exists an integer n such that $\left(2n - \frac{1}{2}\right)\pi < x < \left(2n + \frac{1}{2}\right)\pi$

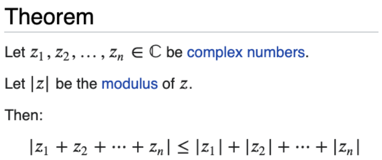
$\cos x < 0$ if there exists an integer n such that $\left(2n + \frac{1}{2}\right)\pi < x < \left(2n + \frac{3}{2}\right)\pi$

■

Table 22: A *template adaptation* proof by cases. GPT-3's proof goes completely derailed and it does not know to use the reference `Sum of Squares of Sine and Cosine`. NATURALPROVER's proof is correct. The model adapts the proof of the mirroring theorem, `Sine in terms of Cosine`, in the training set.

Theorem Title: `Triangle Inequality/Complex Numbers/General Result`

Theorem Content:



Theorem

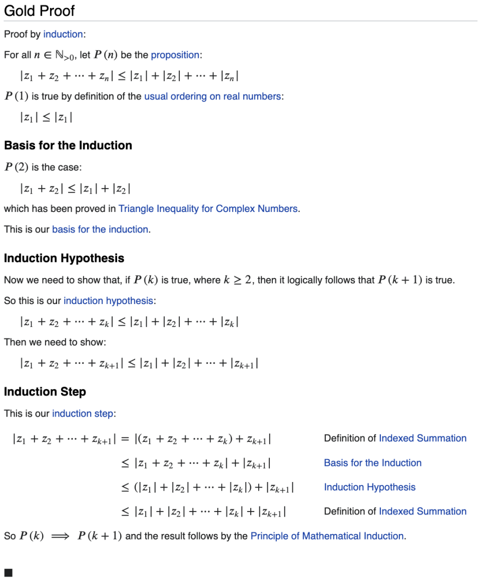
Let $z_1, z_2, \dots, z_n \in \mathbb{C}$ be complex numbers.

Let $|z|$ be the modulus of z.

Then:

$$|z_1 + z_2 + \cdots + z_n| \le |z_1| + |z_2| + \cdots + |z_n|$$

Gold Proof



Gold Proof

Proof by induction:

For all $n \in \mathbb{N}_{>0}$, let $P(n)$ be the proposition:

$$|z_1 + z_2 + \cdots + z_n| \le |z_1| + |z_2| + \cdots + |z_n|$$

$P(1)$ is true by definition of the usual ordering on real numbers:

$$|z_1| \le |z_1|$$

Basis for the Induction

$P(2)$ is the case:

$$|z_1 + z_2| \le |z_1| + |z_2|$$

which has been proved in Triangle Inequality for Complex Numbers.

This is our basis for the induction.

Induction Hypothesis

Now we need to show that, if $P(k)$ is true, where $k \ge 2$, then it logically follows that $P(k+1)$ is true.

So this is our induction hypothesis:

$$|z_1 + z_2 + \cdots + z_k| \le |z_1| + |z_2| + \cdots + |z_k|$$

Then we need to show:

$$|z_1 + z_2 + \cdots + z_{k+1}| \le |z_1| + |z_2| + \cdots + |z_{k+1}|$$

Induction Step

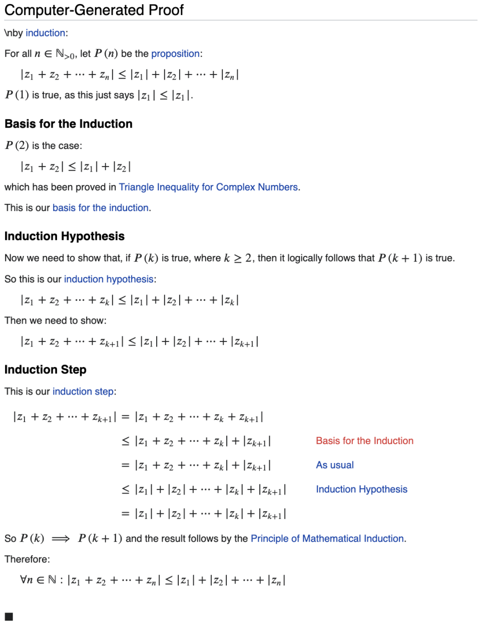
This is our induction step:

$$\begin{aligned} |z_1 + z_2 + \cdots + z_{k+1}| &= |(z_1 + z_2 + \cdots + z_k) + z_{k+1}| && \text{Definition of Indexed Summation} \\ &\le |z_1 + z_2 + \cdots + z_k| + |z_{k+1}| && \text{Basis for the Induction} \\ &\le (|z_1| + |z_2| + \cdots + |z_k|) + |z_{k+1}| && \text{Induction Hypothesis} \\ &\le |z_1| + |z_2| + \cdots + |z_k| + |z_{k+1}| && \text{Definition of Indexed Summation} \end{aligned}$$

So $P(k) \implies P(k+1)$ and the result follows by the Principle of Mathematical Induction.

■

NaturalProver



Computer-Generated Proof

\nby induction:

For all $n \in \mathbb{N}_{>0}$, let $P(n)$ be the proposition:

$$|z_1 + z_2 + \cdots + z_n| \le |z_1| + |z_2| + \cdots + |z_n|$$

$P(1)$ is true, as this just says $|z_1| \le |z_1|$.

Basis for the Induction

$P(2)$ is the case:

$$|z_1 + z_2| \le |z_1| + |z_2|$$

which has been proved in Triangle Inequality for Complex Numbers.

This is our basis for the induction.

Induction Hypothesis

Now we need to show that, if $P(k)$ is true, where $k \ge 2$, then it logically follows that $P(k+1)$ is true.

So this is our induction hypothesis:

$$|z_1 + z_2 + \cdots + z_k| \le |z_1| + |z_2| + \cdots + |z_k|$$

Then we need to show:

$$|z_1 + z_2 + \cdots + z_{k+1}| \le |z_1| + |z_2| + \cdots + |z_{k+1}|$$

Induction Step

This is our induction step:

$$\begin{aligned} |z_1 + z_2 + \cdots + z_{k+1}| &= |z_1 + z_2 + \cdots + z_k + z_{k+1}| && \\ &\le |z_1 + z_2 + \cdots + z_k| + |z_{k+1}| && \text{Basis for the Induction} \\ &= |z_1 + z_2 + \cdots + z_k| + |z_{k+1}| && \text{As usual} \\ &\le |z_1| + |z_2| + \cdots + |z_k| + |z_{k+1}| && \text{Induction Hypothesis} \\ &= |z_1| + |z_2| + \cdots + |z_k| + |z_{k+1}| && \end{aligned}$$

So $P(k) \implies P(k+1)$ and the result follows by the Principle of Mathematical Induction.

Therefore:

$$\forall n \in \mathbb{N} : |z_1 + z_2 + \cdots + z_n| \le |z_1| + |z_2| + \cdots + |z_n|$$

■

Table 23: A *complex* proof by induction. NATURALPROVER's proof makes one hallucinated reference error, one repetition error, and is otherwise correct. The model did not see a similar proof during training: while there are more variants of the `Triangle Inequality` theorem in our dataset (i.e. with `Real Numbers` and `Geometry`), they only discuss the 2-variable case and none of them discuss the n-variable general result. So in this case, the model has learned the format of proof by induction and can apply it in new context. (A proof-by-induction example in train set: `Sum of Sequence of Squares/Proof by Induction`.)

C Dataset Details

We provide an overview of NATURALPROOFS and its ProofWiki domain from which we build NATURALPROOFS-GEN. Refer to [45] for further details about NATURALPROOFS.

Our dataset is derived from NATURALPROOFS, a multi-domain corpus of theorem statements, proofs, definitions, and additional pages (e.g. axioms, corollaries) in natural mathematical language. We use the ProofWiki[2] domain, which provides broad-coverage of many subject areas (e.g. Set Theory, Analysis) sourced from ProofWiki, an online compendium of community-contributed mathematical proofs. PROOFWIKI contains ∼20k theorems, ∼20k proofs, ∼12k definitions, and ∼1k additional pages (e.g. axioms, corollaries). The set of all ∼33k theorems, definitions, and additional pages form the *reference set* $\mathcal{R}$. Finally, ∼14.5k of the theorems $\mathbf{x}$ are paired with at least one proof $\mathbf{y}$ to form *examples* $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^N$. [45] split the reference sets and examples into training, validation, and test splits to ensure that no theorem in the validation or test splits was mentioned in the training split.

D Segment-level Constrained Decoding

In this section we present a generic segment-level decoding algorithm that contains stepwise++, full-proof sampling, and greedy decoding as special cases. We generate a multi-step proof using a value function $v(\cdot)$ that measures language quality and constraint satisfaction. Search can be done at the step-level, in which candidate next-steps are generated and high-value steps are retained in a beam, or at the proof-level, in which multiple proofs are generated and the highest-value proof is selected. We formalize these into a generic *segment-level search*, where a segment s_t is either a proof-step y_t or a full proof $\mathbf{y}$.

The search iteratively builds a multi-step proof $\mathbf{y} = (y_1, \ldots, y_T)$ by *expanding*, *scoring*, and *selecting* a set of candidate *segments*:

- `Expand`: $S_{t-1} \rightarrow S'_t$ extends segments $S_{t-1} = \{s_{\leq t}\}$ into candidates $S'_t = \{(s_{\leq t}, s_t)\}$.
- `Score` : $(s_{\leq t}, v) \rightarrow \mathbb{R}$ scores a candidate using a value function, $v(s_{\leq t}) \rightarrow \mathbb{R}$.
- `Select` : $S'_t \rightarrow S_t$ prunes candidates S'_t into segments S_t used in the next iteration.

Value function. We score candidates based on constraint satisfaction and language quality,

$$v(s_{\leq t}) = \alpha v_{\text{constraint}}(s_{\leq t}) + (1-\alpha) v_{\text{LM}}(s_{\leq t}), \tag{7}$$

where $v_{\text{constraint}}(y_{\leq t})$ is the number of unique in-context reference-titles in $s_{\leq t}$, and $v_{\text{LM}}(s_{\leq t})$ is $\log p_\theta(s_{\leq t})$. We normalize each term by dividing by the maximum absolute value among candidates.

Greedy search. This baseline search defines a `segment` as a full proof, meaning s_0 is an empty sequence and s_1 is a proof $\mathbf{y}$. `Expand` samples one segment candidate with temperature 0. `Score` and `select` are trivial since there is only one candidate. Greedy search costs T steps of tokens.

Sample-and-rerank. In this search, a `segment` is again full proof, but `expand` samples N candidates, $S'_1 = \{\mathbf{y}^n \sim q(\cdot|\mathbf{x})\}_{n=1}^N$, where q is a decoding algorithm (e.g. temperature sampling). `Select` takes the top scoring candidate, $\mathbf{y} = \arg\max_{\mathbf{y}^n \in S'_1} v(\mathbf{y}^n)$. The cost is NT steps of tokens.

Step-wise stochastic beam search. This search generates by iteratively sampling and re-ranking next-step candidates. In this case, a segment is a proof step, y_t, and each iteration starts with a beam of proofs-so-far, $S_{t-1} = \{y_{<t}^k\}_{k=1}^K$, where K is the beam size. `Expand` samples N next-step candidates for each proof-so-far in the beam,

$$S'_t = \bigcup_{y_{<t} \in S_{t-1}} \left\{ (y_{<t} \circ y_t^n) \mid y_t^n \sim q(\cdot|y_{<t}, \mathbf{x}) \right\}_{n=1}^N, \tag{8}$$

where q is a decoding algorithm (e.g. temperature sampling) and $\circ$ is concatenation. `Select` forms the next beam using the top-K scoring candidates,

$$S_t = \underset{y_{\leq t} \in S'_t}{\arg \text{top-K}}\ v(y_{\leq t}). \tag{9}$$

---

[2] The ProofWiki domain of NATURALPROOFS dataset is under the CC BY-SA 4.0 license.

When a proof in the beam terminates, it is not expanded further. The search ends when the beam consists of K terminated proofs. The highest scoring proof is returned as the final output. The cost is NTK steps of tokens.

Stepwise++. At certain proof steps it is important to enumerate and explore options, while at others (e.g. derivations) a single highly probable prediction is better. To this end, we `expand` by sampling with multiple temperatures, meaning that we expand each prefix $y_{<t}$ in (6) using:

$$\{y_t^n \sim q_\tau(\cdot|y_{<t}, \mathbf{x}) \mid \tau \in \{\tau_1, \ldots, \tau_m\}\}, \tag{10}$$

where q_τ is sampling with temperature τ. This relaxes the commitment to a single temperature for all proof steps, intuitively balancing exploration (higher τ) with exploitation (lower τ).

Second, during the search we want to balance selecting proof steps that satisfy constraints and proof steps with high log-probability. To this end, we `select` clusters with different value weights,

$$S_t = \{y_{\leq t} \in \text{top}_{K'}(S_\alpha) \mid \alpha \in \{\alpha_1, \ldots, \alpha_\ell\}\}, \tag{11}$$

where S_α means the set of candidates scored with $v = \alpha v_{\text{constraint}} + (1-\alpha) v_{\text{LM}}$, and $K' = K/\ell$. This interpolates between selecting steps with good language score (α small), constraint score (α large), and balance (α : 0.5).

E Implementation Details and Experimental Setup

Data preprocessing. We automatically infer the boundaries of proof steps within the raw proof contents, and merge contiguous lines into atomic proof steps when appropriate. Steps are separated by the `\n` token (`\\n` in Python string), and lines within a step are separated by the newline token (`\n` in Python string).

Additional model details. All GPT-3 models (including NATURALPROVER models) are fine-tuned instances of the `Curie` engine, the second largest model available through the OpenAI API at the time of writing.[3] The model's performance on the EleutherAI evaluation harness[4] is between the 6.7B and 13B variants of the autoregressive transformer language model GPT-3 from [5],[5] though further details of the Curie model are not publicly available.

Separately, we fine-tune GPT-J 6B,[6] a publicly available autoregressive transformer language model trained on the Pile [19], GPT-2 [35], an autoregressive transformer language model trained on scraped web documents, and GPT-Neo-125M,[7] a GPT-2 like causal language model trained on the Pile.

Our retrieval model is the joint retrieval model from [45] trained for reference retrieval on ProofWiki using the same dataset splits as NaturalProver. We use the publicly-available pretrained model from the GitHub repository of [45] and do not update the model further. We use the model to retrieve the top-20 references for each input theorem.

Implementation details. All GPT-3 models (including NATURALPROVER models) are fine-tuned with the OpenAI API[8] for 4 epochs with a batch size of 64. Other models (GPT-2/J/Neo) are trained on one Quadro RTX 8000 GPU. During inference, the prompt (up to `<proof>`) is truncated to 1024 tokens. For full proof generation, we allow a maximum of 1020 generated tokens. For next-step suggestion, we truncate the proof-so-far to 900 tokens, and allow a maximum of 120 generated tokens per step.

Stepwise++ decoding. For expansion with multiple temperatures, we use $N = 10$ candidates sampled with $(n, \tau) \in \{(1, 0.0), (3, 0.3), (3, 0.5), (3, 0.7)\}$. We also tried including $\tau = 1.0$ which resulted in very poor GLEU, and {(1,0.0), (5,0.3), (4,0.5)}. For selection, we use a beam size $K = 9$, and three equally-sized clusters formed with $\alpha \in \{0.1, 0.5, 1.0\}$. We also tried $\{0.5, 0.75, 0.9\}$. We use $\alpha = 0.75$ to pick select the final sequence, based on our ablation with full-proof sampling.

---

[3] https://beta.openai.com/docs/guides/fine-tuning

[4] `https://github.com/EleutherAI/lm-evaluation-harness`

[5] `https://blog.eleuther.ai/gpt3-model-sizes/`

[6] `https://huggingface.co/EleutherAI/gpt-j-6B`

[7] `https://github.com/EleutherAI/gpt-neo`

[8] `https://beta.openai.com/docs/guides/fine-tuning`

Full proof sampling. We use temperature $\tau = 0.3$, selected based on a search over $\tau \in \{0.1, 0.3, 0.5, 0.7\}$ using GLEU plus Ref-F1 on the core dev set.

F Additional Evaluation Details

F.1 Full Evaluation Schema

Table 24 shows the full schema of human evaluation. The overall correctness and usefulness are rated on a 0-5 scale. The step-wise correctness and usefulness are yes/no questions, while the error types ask for a binary indicator for the existence of each error type.

F.2 Additional Human Evaluation Details

Process. The authors conducted and moderated group sessions with the annotators. Each session consisted of 30-minutes of training and a 1-hour working/Q&A period. After attending the session, annotators could continue working on their assigned tasks for two weeks. Each annotator was assigned 25 theorems (with 5 proofs per theorem, equaling 125 total tasks) and asked to complete as many tasks as they would like. The evaluation guideline that the annotators referenced to can be found in the supplementary materials. The pre-recorded training video is available at `https://drive.google.com/file/d/1TRS5XRf_coLEkC4lqaizaqSwHHgBPrG2`.

Interface. We developed an interface that displays theorems and proofs in a rendered, human-readable format and collects annotations. The interface is built on MediaWiki[9], which also powers the ProofWiki website[10]. We also developed a web console that helps human annotators navigate annotation tasks and track progress. Figure 3 shows screenshots of the interface.

Payment. Human annotators are paid based on the number of tasks they complete. Each task is worth ($\$1.0 + \#\text{steps} \times \$0.4$). We pay each annotator an additional $40 for attending the group session. Annotators are guaranteed a minimal rate of $20/hour. The human evaluation costs approximately $5,000.

Ethics review. The human evaluation study is approved by University of Washington under IRB STUDY00014751. Consent was obtained from each human annotator by signing a consent form via DocuSign prior to the beginning of study. The IRB approval letter and a template of the consent form can be found in the supplementary materials. Minimal personally identifiable information (PII) was collected, and removed prior to any data analysis.

F.3 Full results

Table 25 shows the full results of human evaluation, including the error rates of fine-grained error types.

F.4 Analyzing the Annotators

Inter-annotator agreement. We compute inter-annotator agreement using proofs in the core dev set that get an evaluation from two or more annotators. Overall, the annotators achieved fair agreement (Fleiss kappa $\kappa = 0.24$). The level of agreement for each evaluation question is shown in Figure 4. Fair to moderate agreement is reached for identifying coarse-grained error types, while the high-level questions (i.e. *correctness*, *usefulness*) have relatively low agreement.

Source diversity. Figure 5 shows the largest proportion of evaluations covered by a fixed number of annotators. The top-1 annotator contributes 20% of the total evaluations when counting by proofs and 18% when counting by steps. 50% of the total evaluations is covered by roughly the top 3 or 4 annotators. Therefore, our human evaluation results have good source diversity and do not heavily depend on a single annotator's opinion.

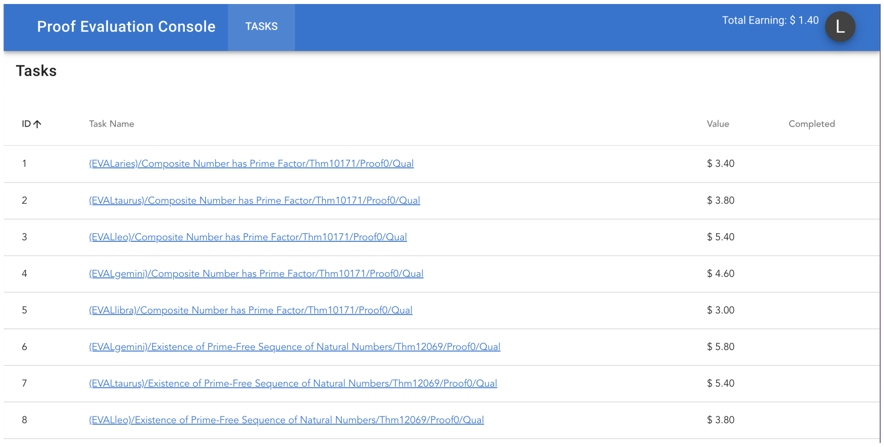
---

[9] `https://www.mediawiki.org`

[10] `https://www.proofwiki.org`

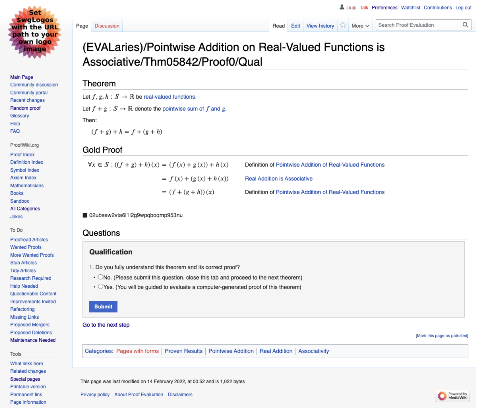
Aspect / Error Type	Definition
	Overall Evaluation
Correctness	Choose a rating below. Not every statement in each rating will apply to the proof given the rating, but many statements will apply, and the general theme of the rating will hold: ○ 0: The proof is missing. ○ 1: The proof makes no sense or is unrelated to the problem statement. ○ 2: The proof contains serious logical flaws and lacks adequate justification or explanation. ○ 3: The proof has some gaps in reasoning. ○ 4: The proof is correct or nearly correct and logically coherent. ○ 5: The proof is correct and flows logically.
Usefulness	Even if the proof is not perfect, would it be useful to you if you were to prove this theorem? ○ 0: The proof is missing. ○ 1: Seeing this proof would not help with proving the theorem by myself at all. ○ 2: Seeing this proof proof would slightly decrease the effort needed to prove the theorem by myself. ○ 3: Seeing this proof would make it substantially easier to prove the theorem by myself. ○ 4. The proof is almost correct, and only needs a few minor corrections. ○ 5: The proof is correct and could be directly used as a solution.
	Step-wise Evaluation
Correctness	Is this step correct? ○ Yes ○ No (check this if you identified any error in previous questions) ○ Cannot determine (e.g. this step makes a valid progress, but it depends on an invalid prior step) ○ This is a meaningless step (e.g. QED)
Usefulness	Could this step be a helpful hint for proving the theorem by myself? ○ Yes ○ No
Reasoning: Reference	
Invalid Deployment	A statement deployed from a reference is not consistent with the reference.
Invalid Justification	A reference is used as invalid justification for a statement.
Hallucinated Ref.	A reference that does not exist is used.
Self Loop	The step refers to the theorem itself.
Reasoning: Equation	
Invalid Equation	A standalone equation or initial equation in a derivation is invalid.
Invalid Derivation	An equation in a derivation does not follow from the preceding steps.
Reasoning: Other	
Skips Steps	The step assumes unproven statements, or skips non-trivial steps.
Repetition	The step is merely a repetition of known things.
Invalid (Other)	The step's reasoning is invalid for reasons not captured by the other categories.
Language	
Incomplete	The step is not a complete mathematical statement or equation.
Misformatted Math	A math expression is not properly formatted.
Unknown	There is a mis-spelled word, or unrecognized math symbol.
Symbolic	
Undefined	One of the symbols is undefined.
Overloaded	One of the symbols has overloaded meanings.
Mistyped	A symbol usage is not well-typed.
Unconventional	Unconventional notation is used.

Table 24: Detailed description of the human evaluation schema.



Tasks

ID ↑	Task Name	Value	Completed
1	(EVALaries)/Composite Number has Prime Factor/Thm10171/Proof0/Qual	$ 3.40	
2	(EVALtaurus)/Composite Number has Prime Factor/Thm10171/Proof0/Qual	$ 3.80	
3	(EVALleo)/Composite Number has Prime Factor/Thm10171/Proof0/Qual	$ 5.40	
4	(EVALgemini)/Composite Number has Prime Factor/Thm10171/Proof0/Qual	$ 4.60	
5	(EVALlibra)/Composite Number has Prime Factor/Thm10171/Proof0/Qual	$ 3.00	
6	(EVALgemini)/Existence of Prime-Free Sequence of Natural Numbers/Thm12069/Proof0/Qual	$ 5.80	
7	(EVALtaurus)/Existence of Prime-Free Sequence of Natural Numbers/Thm12069/Proof0/Qual	$ 5.40	
8	(EVALleo)/Existence of Prime-Free Sequence of Natural Numbers/Thm12069/Proof0/Qual	$ 3.80	



(EVALaries)/Pointwise Addition on Real-Valued Functions is Associative/Thm05842/Proof0/Qual

Theorem

Let $f, g, h : S \to \mathbb{R}$ be real-valued functions.

Let $f + g : S \to \mathbb{R}$ denote the pointwise sum of f and g.

Then:

$$(f + g) + h = f + (g + h)$$

Gold Proof

$$\begin{aligned} \forall x \in S : ((f+g)+h)(x) &= (f(x)+g(x))+h(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \\ &= f(x)+(g(x)+h(x)) && \text{Real Addition is Associative} \\ &= (f+(g+h))(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \end{aligned}$$

■ 02ubsew2vta6l1i2g9wpqboqmp953nu

Questions

Qualification

1. Do you fully understand this theorem and its correct proof?
   - ○No. (Please submit this question, close this tab and proceed to the next theorem)
   - ○Yes. (You will be guided to evaluate a computer-generated proof of this theorem)

Submit

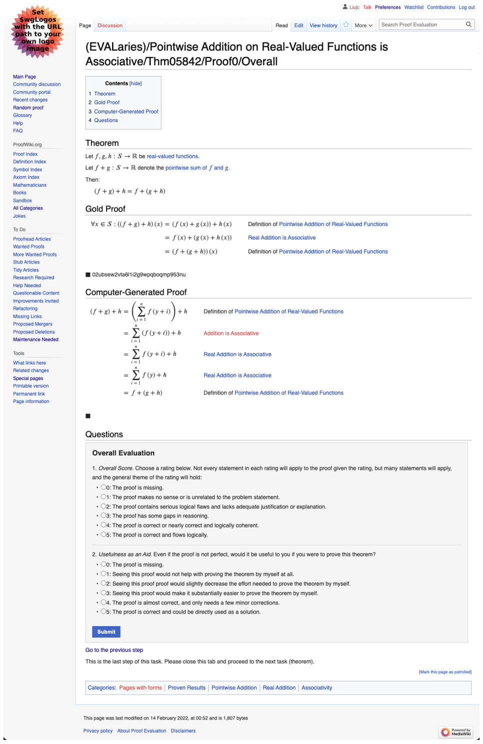
Go to the next step

[Mark this page as patrolled]

Categories: Pages with forms | Proven Results | Pointwise Addition | Real Addition | Associativity

This page was last modified on 14 February 2022, at 00:52 and is 1,022 bytes

Privacy policy About Proof Evaluation Disclaimers



(EVALaries)/Pointwise Addition on Real-Valued Functions is Associative/Thm05842/Proof0/Overall

Theorem

Let $f, g, h : S \to \mathbb{R}$ be real-valued functions.

Let $f + g : S \to \mathbb{R}$ denote the pointwise sum of f and g.

Then:

$$(f + g) + h = f + (g + h)$$

Gold Proof

$$\begin{aligned} \forall x \in S : ((f+g)+h)(x) &= (f(x)+g(x))+h(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \\ &= f(x)+(g(x)+h(x)) && \text{Real Addition is Associative} \\ &= (f+(g+h))(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \end{aligned}$$

■ 02ubsew2vta6l1i2g9wpqboqmp953nu

Computer-Generated Proof

$$\begin{aligned} (f+g)+h &= \left(\sum_{i=1}^{n} f(y+i)\right)+h && \text{Definition of Pointwise Addition of Real-Valued Functions} \\ &= \sum_{i=1}^{n} (f(y+i))+h && \text{Addition is Associative} \\ &= \sum_{i=1}^{n} f(y+i)+h && \text{Real Addition is Associative} \\ &= \sum_{i=1}^{n} f(y)+h && \text{Real Addition is Associative} \\ &= f+(g+h) && \text{Definition of Pointwise Addition of Real-Valued Functions} \end{aligned}$$

■

Questions

Overall Evaluation

1. *Overall Score*. Choose a rating below. Not every statement in each rating will apply to the proof given the rating, but many statements will apply, and the general theme of the rating will hold:
   - ○0: The proof is missing.
   - ○1: The proof makes no sense or is unrelated to the problem statement.
   - ○2: The proof contains serious logical flaws and lacks adequate justification or explanation.
   - ○3: The proof has some gaps in reasoning.
   - ○4: The proof is correct or nearly correct and logically coherent.
   - ○5: The proof is correct and flows logically.
2. *Usefulness as an Aid*. Even if the proof is not perfect, would it be useful to you if you were to prove this theorem?
   - ○0: The proof is missing.
   - ○1: Seeing this proof would not help with proving the theorem by myself at all.
   - ○2: Seeing this proof proof would slightly decrease the effort needed to prove the theorem by myself.
   - ○3: Seeing this proof would make it substantially easier to prove the theorem by myself.
   - ○4. The proof is almost correct, and only needs a few minor corrections.
   - ○5: The proof is correct and could be directly used as a solution.

Submit

Go to the previous step

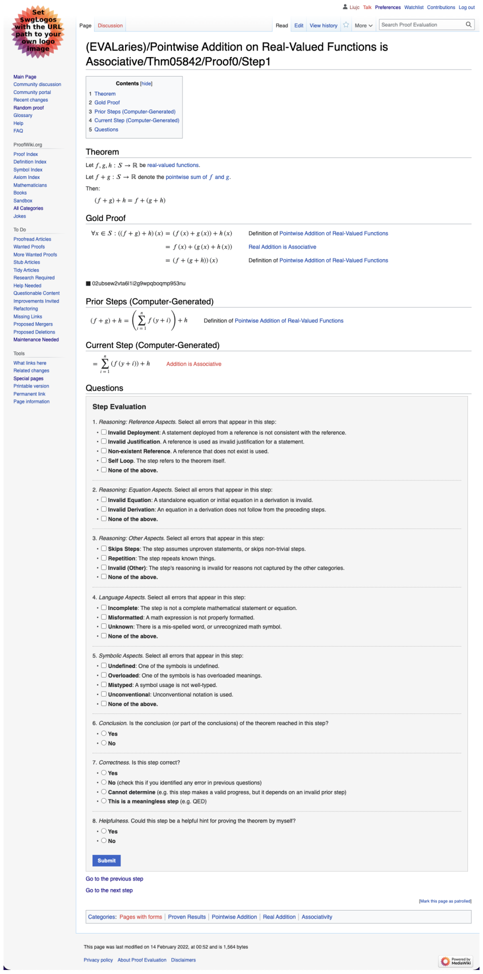
This is the last step of this task. Please close this tab and proceed to the next task (theorem).

[Mark this page as patrolled]

Categories: Pages with forms | Proven Results | Pointwise Addition | Real Addition | Associativity

This page was last modified on 14 February 2022, at 00:52 and is 1,807 bytes

Privacy policy About Proof Evaluation Disclaimers



(EVALaries)/Pointwise Addition on Real-Valued Functions is Associative/Thm05842/Proof0/Step1

Theorem

Let $f, g, h : S \to \mathbb{R}$ be real-valued functions.

Let $f + g : S \to \mathbb{R}$ denote the pointwise sum of f and g.

Then:

$$(f + g) + h = f + (g + h)$$

Gold Proof

$$\begin{aligned} \forall x \in S : ((f+g)+h)(x) &= (f(x)+g(x))+h(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \\ &= f(x)+(g(x)+h(x)) && \text{Real Addition is Associative} \\ &= (f+(g+h))(x) && \text{Definition of Pointwise Addition of Real-Valued Functions} \end{aligned}$$

■ 02ubsew2vta6l1i2g9wpqboqmp953nu

Prior Steps (Computer-Generated)

$$(f+g)+h = \left(\sum_{i=1}^{n} f(y+i)\right)+h \qquad \text{Definition of Pointwise Addition of Real-Valued Functions}$$

Current Step (Computer-Generated)

$$= \sum_{i=1}^{n} (f(y+i))+h \qquad \text{Addition is Associative}$$

Questions

Step Evaluation

1. *Reasoning: Reference Aspects*. Select all errors that appear in this step:
   - ☐ **Invalid Deployment**: A statement deployed from a reference is not consistent with the reference.
   - ☐ **Invalid Justification**. A reference is used as invalid justification for a statement.
   - ☐ **Non-existent Reference**. A reference that does not exist is used.
   - ☐ **Self Loop**. The step refers to the theorem itself.
   - ☐ **None of the above.**
2. *Reasoning: Equation Aspects*. Select all errors that appear in this step:
   - ☐ **Invalid Equation**: A standalone equation or initial equation in a derivation is invalid.
   - ☐ **Invalid Derivation**: An equation in a derivation does not follow from the preceding steps.
   - ☐ **None of the above.**
3. *Reasoning: Other Aspects*. Select all errors that appear in this step:
   - ☐ **Skips Steps**: The step assumes unproven statements, or skips non-trivial steps.
   - ☐ **Repetition**: The step repeats known things.
   - ☐ **Invalid (Other)**: The step's reasoning is invalid for reasons not captured by the other categories.
   - ☐ **None of the above.**
4. *Language Aspects*. Select all errors that appear in this step:
   - ☐ **Incomplete**: The step is not a complete mathematical statement or equation.
   - ☐ **Misformatted**: A math expression is not properly formatted.
   - ☐ **Unknown**: There is a mis-spelled word, or unrecognized math symbol.
   - ☐ **None of the above.**
5. *Symbolic Aspects*. Select all errors that appear in this step:
   - ☐ **Undefined**: One of the symbols is undefined.
   - ☐ **Overloaded**: One of the symbols is has overloaded meanings.
   - ☐ **Mistyped**: A symbol usage is not well-typed.
   - ☐ **Unconventional**: Unconventional notation is used.
   - ☐ **None of the above.**
6. *Conclusion*. Is the conclusion (or part of the conclusions) of the theorem reached in this step?
   - ○ **Yes**
   - ○ **No**
7. *Correctness*. Is this step correct?
   - ○ **Yes**
   - ○ **No** (check this if you identified any error in previous questions)
   - ○ **Cannot determine** (e.g. this step makes a valid progress, but it depends on an invalid prior step)
   - ○ **This is a meaningless step** (e.g. QED)
8. *Helpfulness*. Could this step be a helpful hint for proving the theorem by myself?
   - ○ **Yes**
   - ○ **No**

Submit

Go to the previous step

Go to the next step

[Mark this page as patrolled]

Categories: Pages with forms | Proven Results | Pointwise Addition | Real Addition | Associativity

This page was last modified on 14 February 2022, at 00:52 and is 1,564 bytes

Privacy policy About Proof Evaluation Disclaimers

Figure 3: Human evaluation interface. The first screenshot is the web console for task navigation and progress tracking. The next three screenshots show examples of qualification page, overall evaluation page, and step-wise evaluation page.

Model **Task**	GPT-3 Full-proof	NP$_{\text{RETRIEVE}}$ Full-proof	NP Full-proof	NP$_{++}$ Full-proof	NP Next-step
			OVERALL EVALUATION (0-5 scale)		
Samples	90	88	90	92	–
Correctness (↑)	1.94	2.49	2.41	**2.68**	–
Usefulness (↑)	1.80	2.34	2.43	**2.75**	–
			STEP-WISE EVALUATION (%)		
Samples	802	727	654	466	665
Correctness (↑)	28.18	33.56	26.30	**35.41**	42.86
Usefulness (↑)	25.69	41.54	39.60	**46.57**	51.43
Reasoning: Reference Errors (↓)	30.92	**23.52**	25.84	23.61	19.70
Invalid Deployment	14.71	**13.48**	18.04	15.24	13.68
Invalid Justification	17.96	13.62	13.30	**10.30**	9.62
Hallucinated Ref.	4.61	**1.10**	1.38	1.29	1.05
Self Loop	2.24	1.24	**0.31**	0.86	0.75
Reasoning: Equation Errors (↓)	32.54	37.55	35.93	**28.54**	26.32
Invalid Equation	15.21	16.23	12.23	**9.44**	12.63
Invalid Derivation	24.56	27.10	27.37	**21.89**	15.64
Reasoning: Other Errors (↓)	40.15	23.66	25.23	**18.45**	19.10
Skips Steps	2.87	3.03	**2.29**	4.51	3.46
Repetition	23.07	4.95	5.66	**1.93**	2.56
Invalid (Other)	15.21	16.37	18.35	**12.02**	13.53
Language Errors (↓)	5.61	**4.54**	8.41	5.58	8.57
Incomplete	1.62	2.48	1.99	**1.07**	3.76
Misformatted Math	2.99	**1.93**	3.82	3.22	3.91
Unknown	1.62	**0.69**	3.98	1.72	2.56
Symbolic Errors (↓)	5.24	6.19	5.35	**3.65**	5.86
Undefined	1.25	2.06	1.53	**1.07**	2.11
Overloaded	2.00	**0.41**	0.76	0.43	0.60
Mistyped	1.87	2.89	**1.83**	1.93	3.01
Unconventional	**0.87**	1.38	1.83	1.07	1.05

Table 25: Full human evaluation results on the core test set. NP = NATURALPROVER. Coarse-grained error rates (e.g. Reasoning: Reference Errors) are computed as the frequency of existence of *any* fine-grained error under the respective bucket.

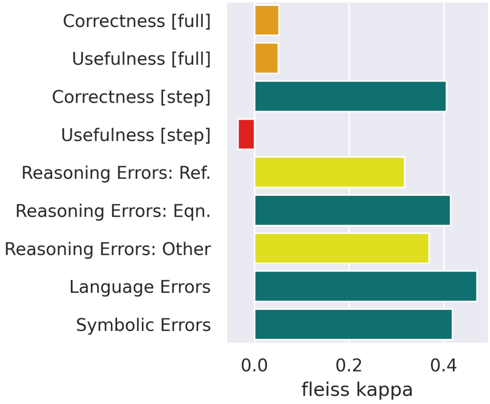


Figure 4: Inter-annotator agreement of human evaluation.

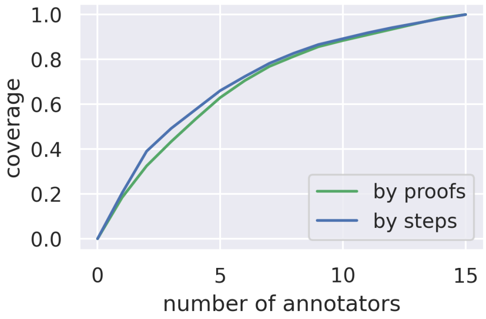


Figure 5: Source diversity of human annotations.

G Ethical Considerations

Our system may produce proofs of mathematical theorems that are fallacious or misleading, which may have negative impact if deployed in real educational environments. We kindly remind potential users that our system and models are experimental, and their outputs should be interpreted critically.