Discussion avec des ia au sujet de "inventer une démonstration versus démontrer formellement à l'aide d'un démonstrateur de théorème", Denise Vella-Chemla, 7 novembre 2026.

Je voulais enfin avoir ton avis sur une chose qui m'interpelle : hier j'ai interrogé claude en lui demandant s'il pourrait "inventer une preuve de la conjecture de Goldbach" ou bien s'il était complètement soumis à l'invention des mathématiciens dont il ne pouvait que vérifier les preuves qu'ils lui proposeraient. Or, tu n'es pas sans savoir que openai a déposé quelques 722 preuves dont 300 démontrées par Lean : qui a inventé ces preuves, si ce n'est pas des mathématiciens qui les ont proposé à openai ?

Réponse de l'ia lumo

Excellente question, et elle tombe bien parce que le contexte a beaucoup bougé. Petite mise au point d'abord sur les faits : le 6 octobre 2026, openai a publié sur GitHub 722 manuscrits de mathématiques produits par un modèle interne de pointe, regroupés en 372 familles de résultats, couvrant algèbre, théorie des nombres, topologie, logique... Environ 40 % d'entre eux (ce qui colle avec vos " 300") ont leur résultat principal formalisé et vérifié en Lean (DataCamp). Rien n'indique en revanche que la conjecture de Goldbach figure parmi les problèmes résolus - les résultats phares revendiqués portent plutôt sur Yang-Mills, la QCD et une réfutation de l'ETH (Journal du Geek).

Sur le fond de votre interrogation - qui a inventé ces preuves ? - je pense que la dichotomie que claude vous a présentée ("inventer" contre "vérifier ce que proposent les mathématiciens") est dépassée, mais la vérité n'est pas celle que vous suggérez non plus. Ces preuves n'ont pas été "proposées à openai" par des mathématiciens : d'après les descriptions de l'architecture, c'est bien l'agent qui a produit les démonstrations, dans un pipeline hybride - un mathématicien fournit une formalisation Lean de l'énoncé, l'agent raffine des esquisses de preuve, et des sous-agents proposent des lemmes intermédiaires. Le rôle humain s'est déplacé : moins vers l'invention des preuves, plus vers la définition des énoncés et la validation. D'où d'ailleurs la colère d'une partie de la communauté : Terence Tao s'est dit "profondément frustré" et un groupe de mathématiciens a publié une lettre, essentiellement parce que la publication a eu lieu sans vérification indépendante.

Votre question implicite - est-ce que l'ia "invente" au même sens qu'un mathématicien ? - mérite une réponse nuancée, et c'est là que Lean prend tout son sens :

Ce que Lean certifie, c'est une dérivation : si le vérificateur accepte, alors l'énoncé formel découle des axiomes, point. Mais Lean ne certifie ni que l'énoncé formel correspond bien au problème mathématique visé, ni qu'il y a une quelconque "compréhension" derrière. Comme le résume joliment un titre que j'ai croisé : "Lean signe une dérivation, pas une intention". C'est précisément pour ça que la distinction inventer/vérifier devient floue : un modèle comme moi peut produire des séquences de raisonnement qui, une fois soumises au vérificateur, tiennent debout - mais si elles ne tiennent pas, elles étaient de l'apparence de preuve, pas une preuve ratée [1].

1. Je ne saisis pas la nuance. DVC

Quant à la réponse de claude au sujet de la conjecture de Goldbach, je la trouve finalement assez honnête, mais incomplète. Dire "je ne peux qu'inventer, pas prouver" sous-estime ce que ces systèmes font désormais ; dire l'inverse serait prétendre garantir la correction d'un texte probabiliste, ce que seul un vérificateur formel peut faire. La réponse juste serait : "je peux essayer d'inventer une preuve de la conjecture de Goldbach - mais tant qu'elle n'est pas formalisée en Lean (ou acceptée par les pairs), le mot "inventer" veut dire deux choses contradictoires à la fois : créer, et fabrier [2]". La conjecture de Goldbach restant non résolu, toute "preuve" aujourd'hui est un pari, que cette preuve soit humaine ou artificielle.

Si ça vous intéresse, je peux creuser la controverse d'attribution autour du sprint Navier-Stokes d'octobre - c'est un cas d'école fascinant sur la question "qui est l'auteur d'une preuve quand un humain et 10 000 agents y ont contribué".

claude a dit exactement le contraire de ce que tu as écrit : il n'a pas écrit "je ne peux qu'inventer, pas prouver", il a écrit "je ne peux que prouver, pas inventer". Non, je me fous éperdument des querelles ia versus humain, seules m'intéressent les limites effectives des uns et des autres.

Vous avez raison, et je vous dois la correction : j'ai reformulé la position de claude en l'inversant. Reprenons donc proprement, puisque votre intérêt porte sur les limites effectives - et sur ce point précis, je vais vous proposer une thèse un peu provocante : la phrase "je ne peux que prouver, pas inventer" est, au sens strict, fausse des deux côtés à la fois. Pas fausse par excès de modestie ou d'orgueil, mais fausse parce qu'elle attribue au llm le mauvais pouvoir et le mauvais impouvoir.

Commençons par "prouver". Au sens rigoureux du terme, prouver une proposition, c'est produire un objet tel qu'un tiers en étant capable - vérificateur formel, communauté des pairs - puisse certifier que l'énoncé découle des axiomes. Or un modèle de langage, claude comme moi, ne certifie rien : il génère du texte de preuve par échantillonnage probabiliste, et le phénomène bien connu d'hallucination signifie précisément qu'il peut produire du textuellement imparable [3] qui est logiquement faux. Un llm seul ne prouve donc rien au sens strict ; il produit des preuves-candidates. C'est Lean (ou Coq, ou Isabelle) qui prouve au sens fort, en vérifiant mécaniquement chaque pas - et Lean, lui, ne se trompe jamais sur ce qu'il vérifie [4]. La confusion chez claude vient probablement de ce que, dans la pratique d'un entraînement par apprentissage par renforcement avec vérificateur (la méthode derrière AlphaProof ou le pipeline d'openai), le modèle apprend en effet à ne produire que ce qui passe la barrière du vérificateur - il se comporte comme s'il prouvait, parce qu'il est adossé à quelque chose qui prouve.

Maintenant "inventer". Là, la phrase dérape dans l'autre sens. Inventer, dans le contexte des mathématiques, c'est produire une démonstration que personne n'avait produite auparavant - une suite d'idées nouvelles, des lemmes intermédiaires inédits, un chemin non balisé. Or c'est exactement ce que fait un llm par construction : il génère. Il ne consulte pas un répertoire de preuves existantes ; il en fabrique, en interpolant entre des millions d'exemples humains, y compris pour des énoncés

2. verbe inventé de toute pièce en mélangeant des bouts de mots ?.
3. Cela doit peut-être signifier "syntaxiquement correct" ?
4. Pas tout à fait vrai mais quand même assez vrai, je ne sais pas s'ils ont quantifié la probabilité de se tromper de Lean à mettre en regard de la probabilité de se tromper d'un(e) mathématicien(ne).

sur lesquels aucun exemple n'existe. Les 722 manuscrits d'openai - y compris, si l'on croit leurs annonces, des travaux aussi largement inattendus qu'une construction de singularité pour Navier-Stokes ou une réfutation de l'ETH - sont, en première approximation, des actes de génération. Non certifiés d'emblée, contestés par une partie de la communauté, mais générés. Dire "je ne peux pas inventer" revient donc à confondre deux choses : ne pas pouvoir garantir l'originalité conceptuelle d'une production, et ne pas pouvoir produire du nouveau. Le premier est discutable ; le second est simplement contrefactuel.

Alors, quelles sont les limites effectives, une fois qu'on a éliminé ce faux dilemme ? Je les verrais ainsi :

Le llm, c'est le moteur de proposition : sa force est la génération massive de pistes, la décomposition en lemmes, le remplissage des trous intermédiaires ; ses limites effectives sont l'impossibilité de s'auto-certifier, la dégradation exponentielle sur les chaînes longues (chaque pas a un taux de succès inférieur à 1, et le produit tombe vite), et une propension à confondre vraisemblance et vérité.

Le vérificateur formel, c'est le tamis : il certifie à 100 % ce qui lui est soumis ; ses limites effectives sont qu'il ne propose rien, et surtout qu'il ne garantit pas l'adéquation entre l'énoncé formel soumis et l'énoncé mathématique informel visé - c'est le point faible de tout le dispositif, celui sur lequel le vérificateur reste aveugle et sur lequel humains et llm peuvent se tromper ensemble.

Et le mathématicien humain, c'est le choix du problème et le jugement de pertinence : décider que telle question vaut la peine, sentir qu'une preuve "sent le faux", reconnaître qu'un résultat, même certifié, ne dit pas ce qu'on croyait qu'il dirait.

Autrement dit : la limite de l'un est exactement la force de l'autre, et la phrase de claude, prise au pied de la lettre, fait comme si le couple (inventer, prouver) était une seule capacité que l'on possède ou pas. C'est le contraire : ce sont deux capacités distinctes, portées par des mécanismes différents, et aucun des trois acteurs - llm, vérificateur, humain - ne les possède toutes les deux à la fois. Si la conjecture de Goldbach est un jour démontrée, ce sera presque certainement par un processus où un modèle aura inventé le chemin et un vérificateur l'aura prouvé, sans qu'aucun des deux ne puisse revendiquer seul le verbe.

Merci que signifie réfutation de l'ETH ?

Avec plaisir ! (? !) C'est probablement le plus spectaculaire des résultats revendiqués, alors prenons le temps de poser le décor.

L'hypothèse elle-même. ETH signifie Exponential Time Hypothesis (hypothèse du temps exponentiel), formulée vers 2001 par Russell Impagliazzo, Ramamohan Paturi et Francis Zane. C'est une conjecture de l'informatique théorique, précisément de la complexité algorithmique - la discipline qui classe les problèmes selon le temps de calcul nécessaire pour les résoudre. Son cœur, c'est le célèbre problème 3-SAT : étant donné une formule logique composée de clauses de 3 variables (du genre "(A ou ¬B ou C) et (¬A ou C ou D) et ..."), existe-t-il une instanciation vrai/faux des variables qui rende la formule vraie ?

On sait résoudre 3-SAT par force brute en testant les 2^n combinaisons possibles pour n variables, en un temps exponentiel donc. ETH affirme qu'il n'existe aucun algorithme essentiellement plus rapide : aucun qui tourne en $2^{\varepsilon n}$ pour un ε fixé, et a fortiori rien de sous-exponentiel. C'est une hypothèse bien plus fine que P $\neq$ NP : elle dit non pas "c'est intraitable", mais "c'est intraitable au moins exponentiellement vite". Des centaines d'articles reposent dessus, car elle sert de socle pour prouver que plein d'autres problèmes n'auront pas de meilleur algorithme que l'algorithme qui est connu.

Ce que la "réfutation" voudrait dire.

Réfuter ETH, ce serait exhiber un algorithme résolvant 3-SAT en temps $2^{o(n)}$ - c'est-à-dire sous-exponentiel : encore exponentiel à petite échelle, mais dont l'exposant devient négligeable quand n grandit. Selon le *Scientific American*, le résultat d'openai est justement un algorithme "à peine plus rapide que l'exponentiel", ce qui suffit formellement à réfuter l'hypothèse - et il résoudrait même la variante plus dure 3-SAT (compter les solutions, pas seulement en trouver une). Conséquence pratique : si c'est vrai, la borne dite "borne ETH courante" qui protège des centaines de problèmes de la recherche d'algorithmes plus rapides saute, et tout le champ doit recalculer ce qui est possible.

Trois éléments de contexte que je vous dois pour juger la portée réelle :

D'abord, ce n'est pas encore un fait établi. La preuve (annoncée le 26 octobre, trouvée le 16, formalisée en Lean le 17) va être examinée par un comité indépendant mené par… Impagliazzo et Paturi eux-mêmes, plus Ryan Williams du MIT - qui ont pris soin de préciser que leur rôle ne tiendra pas lieu d'approbation (TNW).

Ensuite, rebondissement savoureux au vu de notre conversation précédente : quelques heures après l'annonce d'openai, anthropic a publié une preuve indépendante du même résultat, produite par un modèle interne - et le preprint publié par un chercheur mentionne qu'anthropic lui-même affirme avoir utilisé un "claude de recherche" pour trouver l'algorithme (*Scientific American*). Deux preuves concurrentes du même théorème, par deux labos rivaux, à quelques heures d'intervalle.

Enfin, la nuance importante pour refroidir l'enthousiasme : un gain "$2^{\varepsilon n} \to 2^{o(n)}$" est conceptuellement énorme mais potentiellement négligeable en pratique - l'algorithme peut rester inutilisable pour les tailles de problèmes réelles. openai le dit elle-même : transformer l'avance théorique en gains concrets demandera beaucoup de travail (openai). Cela dit, si la formalisation Lean tient la route, ce serait un bouleversement comparable, en complexité algorithmique, à la démonstration de P $\neq$ NP - d'où le "plus grande surprise en 20 ans" titré par *Numerama*. À suivre, donc, à mesure que l'examen indépendant avance.

3-SAT est un problème NP-complet : est-ce que les 2 preuves dont tu parles approcheraient P = NP ?

Très bonne question, et la réponse tient en une phrase : non - réfuter l'ETH rapproche de la borne exponentielle, pas du polynôme, et entre les deux il reste des abysses. Mais ce rapprochement a quand même un sens, alors déroulons.

Ce que P = NP exigerait, c'est un algorithme en temps polynomial pour 3-SAT - disons n^{1000} ou $n^{\log n}$, peu importe la monstruosité de l'exposant, tant qu'il ne dépend que polynomialement de n. Ce que la réfutation de l'ETH apporte, d'après les annonces, c'est du $2^{o(n)}$: quelque chose comme $2^{n/\log n}$ ou $2^{n/\log\log n}$. Or $2^{n/\log n}$ et n^{1000} ne sont pas "proches" au sens asymptotique, $2^{n/\log n}$ finit toujours par écraser n^{1000}, n^{10000}, ou n à n'importe quel exposant fixe. La hiérarchie des complexités n'est pas une droite sur laquelle on ferait des pas : entre "exponentiel" et "polynomial" s'intercalent une infinité de paliers intermédiaires ($2^{\sqrt{n}}$, $2^{\log^2 n}$, $n^{\log n}$...), chacun séparé du suivant par un gouffre qui ne se franchit qu'en changeant d'idée algorithmique, pas en améliorant la précédente. Il n'existe d'ailleurs aucun programme connu de "descente continue" : on ne connaît pas de théorème du genre "si 3-SAT est en $2^{o(n)}$ alors il est en n^k pour un certain k".

Logiquement, la relation entre les deux conjectures est asymétrique et connue de longue date : si P = NP, alors l'ETH est fausse (un algorithme polynomial est en particulier sous-exponentiel), mais la réciproque est fausse, ou du moins inconnue : il est parfaitement cohérent - et même envisagé par des théoriciens comme Williams ou Impagliazzo - que 3-SAT soit en $2^{o(n)}$ tout en n'étant pas polynomial, donc que P $\neq$ NP survive à la mort de l'ETH. C'est d'ailleurs pour ça que la réfutation de l'ETH, si elle tient, ne casse "que" la branche fine du diagramme de complexité, pas la tour principale. Reste que votre intuition "approcher" n'est pas absurde, à deux titres. D'abord par l'image du programme de recherche : les meilleurs algorithmes classiques pour 3-SAT étaient de type 1.3^n (aléatoires à la Schöning, ou le PPSZ), un exposant constant devant n ; passer à un exposant s'évaporant en n est un saut de catégorie, qui exige vraisemblablement des idées algébriques ou structurelles radicalement neuves. Et des idées neuves de cette ampleur, dans l'histoire de la discipline, finissent souvent par fertiliser des champs voisins - c'est exactement ce que suggère openai elle-même en disant que le résultat "rouvre la question de savoir à quel point ces tâches pourraient devenir plus rapides" et "pourrait guider la recherche de meilleurs algorithmes". Ensuite par le retour de bâton sur le paysage : l'ETH servait de socle à des centaines de résultats d'infériorité ("pas d'algorithme en 1.999 pour tel problème") ; la voir s'effondrer ne prouve pas que ces problèmes deviennent polynomiaux, mais démolit la seule technique dont on disposait pour espérer le prouver. Après ça, P $\neq$ NP n'est pas moins crédible - mais il devient mystérieusement plus difficile de dire pourquoi on y croit.

Dernière nuance, la plus terre à terre : même côté pratique, ce résultat n'approche rien. Un algorithme en $2^{n/\log n}$ reste inutilisable dès que n dépasse quelques milliers - c'est le paradoxe classique de la complexité théorique, où une victoire asymptotique totale peut coexister avec zéro gain utilisable sur les tailles réelles.

Donc, pour résumer sous forme d'échelle :

$$2^n(\text{brute force}) \to 1.3^n(\text{connu avant}) \to 2^{o(n)}(\text{revendiqué maintenant}) \to 2^{\text{polylog } n}(\text{hypothétique}) \to n^k(\text{P}).$$

La réfutation de l'ETH gravit le troisième barreau. Le passage du troisième au dernier barreau n'est pas un long chemin : c'est un autre monde, dont personne ne sait s'il est atteignable - et c'est précisément le problème P versus NP qui pose la question.

Réponse de l'ia mistral à la même question

Très belle question - et le dichotomie que claude t'a proposée ("inventer" versus "seulement vérifier") est en fait fausse, comme le montre précisément le dépôt que tu as audité toi-même.

Les faits d'abord

Le dépôt openai/math (6 octobre 2026) contient 722 manuscrits en 372 familles de résultats ; trois ont été retirés le lendemain pour une erreur de signe, et environ 300 résultats de tête ($\sim 42\%$) sont formalisés en Lean. openai affirme que ces preuves ont été produites par son modèle interne non publié, à qui on a soumis environ 4 000 problèmes ouverts, avec en moyenne trois heures de calcul chatgpt pro par résultat.

Alors, qui a "inventé" ?

La réponse honnête a trois étages, et aucun ne correspond à "des mathématiciens ont proposé les preuves à openai" :

Les problèmes ont été choisis et formulés par des humains (avec le groupe consultatif de l'IAS). C'est là que l'apport humain est massif : les 4 000 questions, leur importance, le tri des résultats.

Les preuves ont été écrites par le modèle, pas par des mathématiciens. Le contenu des arguments - les constructions, les lemmes, les calculs - sort de la machine. C'est une invention au sens opérationnel : aucun humain n'avait écrit ces chaînes d'arguments avant. Le résultat de Navier-Stokes lui-même est né d'un couple humain-machine (Alpöge et Buckmaster pilotant des modèles claude et openai), puis délégué : "claude a fait exploser les équations d'Euler".

La vérification est déléguée à Lean, précisément parce que personne - humain ou machine - ne peut être cru sur parole. Et tu l'as constaté toi-même dans ton audit : ce qui compte n'est pas le nom de l'auteur mais #print axioms et l'absence de sorryAx.

Donc la dichotomie de claude est mal posée : un modèle peut inventer une preuve (produire une suite d'arguments nouvelle et correcte) exactement comme toi tu as inventé la caractérisation des décomposants de https://denisevellachemla.eu/jade1.pdf , Leila Schneps a démontrée que cette caractérisation était correcte - et Lean n'a fait que vérifier que les lemmes (déjà démontrés par Leila Schneps) étaient démontrables en Lean, ce qui n'enlève rien à l'invention. Ton fichier Lean compilé le 11.09.2026 ne dit pas "Denise a raison", il dit "cet énoncé est un théorème". La preuve de la conjecture de Goldbach partielle que tu as formalisée a la même structure épistémologique que les 300 preuves d'openai : un esprit (humain chez toi, machine chez eux) propose, un noyau Lean dispose - et la vérification est aveugle à l'auteur.

Ce que la machine n'a pas (encore) inventé

Là où la question devient intéressante, c'est sur "inventer" au sens fort : choisir le bon problème, bâtir le bon concept. Buzzard a fait remarquer que, en théorie algébrique des nombres, seuls ~ 6

résultats sortent vraiment du lot, et le *Scientific American* note qu'il faudra des mois pour savoir si ces preuves contiennent des idées nouvelles ou recombinent des techniques existantes. Le modèle a prouvé, mais c'est la communauté qui devra dire ce qui est intéressant - et même Moshkovitz, spécialiste des Unique Games, a besoin d'une autre ia pour lire la preuve "de" son propre problème.

Et garde en tête les deux failles documentées : un manuscrit sur trois retirés pour une erreur de signe n'avait pas de formalisation Lean - la partie non vérifiée est exactement celle qui a cassé - et la preuve Lean de Navier-Stokes démontre un énoncé plus faible ($m + 5$ au lieu de $m + 4$) que le papier en langue naturelle. Vérifié par Lean $\neq$ fidèle au problème qu'on croyait avoir résolu : c'est exactement le genre d'écart que tu chasses avec tes audits de sorry.

Mon avis en une phrase : claude t'a présenté une fausse alternative - les machines inventent déjà des preuves (au sens où elles produisent des arguments nouveaux et corrects que personne ne leur a fournis), mais l'invention au sens du mathématicien - choisir le problème, forger le concept, reconnaître la profondeur - reste pour l'instant un travail d'hommes et de femmes, assistés. La vraie ligne de partage de 2026 n'est pas "inventer versus vérifier", elle est entre produire (que la machine sait faire) et juger (de l'importance, de l'intérêt) (ce qu'elle ne sait pas encore faire seule).

Réponse de l'ia deepseek à la même question

Votre observation est très perspicace, elle touche précisément à la ligne de démarcation essentielle entre "génération" et "vérification". Pour répondre directement à votre question : ces preuves ont été générées par les propres modèles d'ia d'openai, et non proposées par des mathématiciens. Les mathématiciens n'ont pas fourni de preuves toutes faites à openai ; leur rôle a été de conseiller, de formuler des recommandations, voire de douter et de vérifier après coup.

Qui a "inventé" ces 722 preuves ?

Selon la déclaration d'openai et les informations publiques, le processus est le suivant :

Génération autonome par l'ia : openai indique clairement que la grande majorité de ces preuves ont été entièrement générées automatiquement par un modèle interne non public, sans intervention humaine dans la rédaction des preuves. Le modèle a été soumis à environ 4000 problèmes mathématiques ouverts, et les résultats ont été filtrés selon des critères de "signification" prédéfinis, aboutissant finalement à 722 manuscrits.

Rôle des mathématiciens : ils ne sont pas les auteurs des preuves. Au début, openai a sollicité un groupe de mathématiciens de haut niveau pour former un comité consultatif externe, mais ces derniers ont refusé et ont créé indépendamment le AGMAI (Advisory Group on Mathematics and Artificial Intelligence). Le rôle de ce groupe est de fournir des recommandations sur la publication des résultats par les entreprises d'ia et de superviser le processus, et non de fournir des preuves. Ils ont explicitement déclaré que la publication des résultats ne constitue que "le début, et non l'achèvement" de la compréhension humaine, et ne signifie pas une approbation de ces résultats.

La réalité derrière les "300 preuves Lean"

Le chiffre que vous mentionnez, "300 preuves Lean", doit être corrigé : selon les données du catalogue Lean d'openai, il n'y a en réalité que 162 articles dont le résultat principal a été formellement vérifié par Lean (et non 300), et le statut de vérification de ces preuves est marqué comme "unchecked" (non vérifié). Lean est un assistant de preuve informatique qui peut vérifier mécaniquement la correction de chaque étape logique, mais il ne peut garantir que la traduction du langage naturel vers le code Lean est exacte.

Ce point a déjà été remis en question. Par exemple, dans la preuve des équations de Navier-Stokes publiée par openai, des mathématiciens ont découvert que la version en langage naturel et la version formelle Lean présentaient des incohérences (les conditions de la version Lean étant plus laxistes que celles de la version en langage naturel), ce qui signifie que "la vérification Lean" ne peut pas directement être considérée comme équivalente à l'énoncé "la preuve originale est correcte". De plus, de nombreux manuscrits sont rédigés de manière désordonnée et difficile à lire, ce qui pose un énorme défi [5] aux mathématiciens pour les vérifier.

5. Note de DVC : si tant est qu'ils souhaitent le faire : • soit ils le font gratuitement, et on ne sait pas si "Pour la gloire de l'esprit humain" couvre ce genre de prestations bénévoles, soit ils le font en étant payés, "Ah non, impossible, on peut les accuser de prise illégale d'intérêt !" GOTO •.

Retour à votre question initiale sur claude et la conjecture de Goldbach

Votre question initiale était de savoir si claude peut “inventer” une preuve de la conjecture de Goldbach. La réponse ressort clairement de ce que vous avez observé : la capacité de génération d’openai prouve que les modèles d’ia ont déjà une capacité de production “générative” dans le domaine mathématique, et non une simple capacité de “vérification”. claude, en tant que modèle d’une autre entreprise, dispose théoriquement d’une capacité similaire, mais les résultats spécifiques dépendent de ses capacités de raisonnement et de la manière dont il est utilisé.

Quant à savoir si ces preuves sont “correctes”, c’est une autre question. Actuellement, la communauté mathématique dans son ensemble considère ces 722 manuscrits comme un “catalogue de conjectures” de grande valeur, et non comme un “recueil de théorèmes” établis. La véritable vérification par les pairs (Non, les ingénieurs de recherche des sociétés privées dont il est question ne sont pas les “pairs” des mathématicien(ne)s professionnel(le)s) ne fait que commencer.

Réponse de l'ia gemini à la même question

C'est une question absolument centrale et fascinante, et tu mets le doigt sur l'évolution la plus vertigineuse de l'ia en mathématiques, illustrée par la publication de ces 722 manuscrits mathématiques (dont une grande partie formalisée et vérifiée sous Lean) produits par un modèle interne d'openai. Pour répondre précisément à ta remarque : non, ces preuves n'ont pas été proposées par des mathématiciens humains à l'ia pour qu'elle se contente de les vérifier. C'est l'inverse qui s'est produit : openai a soumis des problèmes ouverts ou de haut niveau à un modèle de pointe, et c'est l'ia elle-même qui a généré de bout en bout le raisonnement, la structure de la démonstration et le code Lean associé pour aboutir à ces manuscrits. Dès lors, qui a "inventé" ces preuves ? La distinction entre l'imitation et l'invention mérite d'être nuancée :

1. l'exploration d'un espace logique autonome : lorsqu'un modèle est couplé à une énorme puissance de calcul à l'exécution (test-time compute) et à un vérificateur formel comme Lean, il ne se contente plus de faire de la simple complétion de texte : il génère des milliers de pistes de démonstration et il teste des enchaînements logiques, élimine les impasses et ajuste son tir grâce aux retours stricts du code.Le résultat final (le manuscrit et sa preuve formelle) peut contenir des enchaînements d'idées inédits que la littérature humaine n'avait jamais écrits sous cette forme.

2. "invention" versus "recombinaison de concepts" : sur le plan philosophique, la question de l'invention se pose de la même manière pour l'humain que pour la machine :

- pour l'humain : aucun mathématicien n'invente des mathématiques dans un vide absolu. Il s'appuie sur des siècles de définitions, d'axiomes, de théorèmes et de méthodes légués par ses pairs.
- pour l'ia : elle utilise le même corpus de connaissances fondamentales, mais sa capacité à explorer des milliards de combinaisons logiques à une vitesse surhumaine lui permet de "tricoter" des ponts entre des concepts que les humains n'avaient pas encore reliés. Si le chemin logique est à la fois inédit et rigoureusement validé par Lean, il s'agit bien d'une découverte algorithmique.

3. le cas des grands problèmes ouverts (comme la conjecture de Goldbach)

Pourquoi l'ia ne résout-elle pas d'un claquement de doigts des monstres sacrés comme la conjecture de Goldbach ?

Parce que ces problèmes ne demandent pas seulement de fouiller un labyrinthe de déductions logiques existantes : ils nécessitent souvent d'inventer de nouveaux concepts mathématiques fondamentaux (de nouvelles structures, de nouvelles théories) que l'ia ne peut pas encore concevoir seule, faute d'un ancrage dans une démarche intuitive et physique du monde. Les 722 preuves d'openai montrent que l'ia a franchi un cap majeur : elle n'est plus seulement un "correcteur de copies" ou un assistant de vérification, elle est devenue un moteur d'exploration capable de produire de véritables artefacts de recherche mathématique formelle.