dinateurs pour une partie de la preuve de 1976. Cette utilisation était en fait seulement l'indice que l'on avait affaire à un résultat particulièrement compliqué.

Notons aussi que le logiciel *Coq* est souple et adaptable à de nombreux problèmes, et que José Grimm de l'INRIA a pu grâce à lui entreprendre de formaliser les *Éléments de mathématique* de Bourbaki. L'idée n'est pas de suivre exactement le formalisme décrit dans le traité – nous avons vu que ce n'était pas envisageable –, mais de concevoir un formalisme équivalent permettant d'exprimer tous les résultats que l'on y trouve en même temps qu'on formalise le détail des preuves. En janvier 2011, une grande partie du livre I du traité de Bourbaki avait été formalisée.

Des théorèmes encore trop complexes

Parmi les 15 résultats de la liste des 100 théorèmes qui restent trop difficiles à formaliser aujourd'hui, même en utilisant des assistants de preuve, citons en deux.

– La transcendance du nombre π. C'est l'affirmation que π n'est racine d'aucun polynôme à coefficients entiers (ce résultat montre que l'antique problème de la quadrature du cercle est insoluble). Le théorème a été démontré par Lindemann en 1882 et on peut dire qu'il est à la limite de ce que peuvent faire les assistants de preuve actuels. Il est probable qu'il sera rapidement prouvé de manière formelle.

– Le théorème de Fermat-Wiles. C'est l'affirmation, démontrée en 1993 par Andrew Wiles, qu'il n'existe pas de solutions à l'équation $x^n + y^n = z^n$ où x, y, z sont des entiers positifs et n un entier fixé supérieur à 2.

Sa preuve est extrêmement difficile. Elle avait été recherchée pendant 350 ans et elle n'est vraiment comprise que par quelques spécialistes. En produire une version formelle serait intéressant.

Même si les erreurs sont fréquentes, le fait qu'une démonstration soit faite et refaite par de nombreux mathématiciens assure qu'aucun résultat classique n'est faux. Jamais dans les parties centrales des mathématiques une erreur grave n'est restée longtemps cachée. Aussi le délicat, long et pénible travail de formalisation des mathématiques avec assistant informatique est-il vraiment utile ? La réponse est deux fois « oui ».

Vraiment utiles ?

Oui, la formalisation est utile, car certaines démonstrations récentes sont si longues que personne n'est entièrement persuadé de leur exactitude. C'est le cas du théorème de Thomas Hales qui résout la conjecture de Kepler concernant l'empilement optimal des sphères dans l'espace. Personne ne peut dire que la preuve de Hales a été contrôlée. Mener la vérification formelle d'une telle preuve n'est donc pas un luxe, mais une nécessité *(voir la figure 2)*. La situation est pire encore dans le cas du théorème de classification des groupes simples, dont la première démonstration a été annoncée en 1983. Elle s'étend sur des dizaines d'articles mathématiques écrits par plus d'une centaine de mathématiciens différents et couvre un total de plus de 10 000 pages. Un trou important y a été découvert, qu'un complément de 1 300 pages est venu combler dans les années 1990. La seconde version de cette démonstration est en cours d'élaboration et s'étendra sur environ 5 000 pages. Bien qu'impossible à envisager dans l'immédiat, il est clair que seule la mise au point d'une preuve formelle sera susceptible de rassurer définitivement les mathématiciens qui, comme Jean-Pierre Serre, contestent l'affirmation que ce théorème « monstrueux » a été démontré.

Oui, la formalisation des démonstrations mathématiques est utile, car elle sert à certifier des logiciels ou des puces informatiques ainsi d'ailleurs qu'à y repérer des erreurs. Les « assistants de preuve » utilisés pour la mise au point des preuves formelles sont des outils puissants exploités par l'industrie du logiciel pour les morceaux de programme dont on veut avoir la certitude qu'ils font correctement certaines tâches cruciales. Aujourd'hui, assez peu de logiciels sont ainsi validés par la formalisation complète de la démonstration qu'ils font ce qu'on attend et rien d'autre, et de nombreuses autres méthodes de contrôle et de certification sont utilisées.

3. Démonstrations

Les démonstrations formelles sont peu lisibles et même souvent illisibles. Cela ne doit pas surprendre, car, sans qu'on ait toujours bien fait la distinction, les démonstrations mathématiques possèdent deux fonctions.

La première fonction d'une démonstration est de garantir que l'affirmation exprimée dans l'énoncé du théorème est exacte. C'est la fonction de certification.

La seconde fonction est de permettre au mathématicien de percevoir et de comprendre pourquoi les choses sont comme le théorème l'énonce. C'est la fonction didactique.

Lorsque les démonstrations ne sont pas trop compliquées, les deux fonctions peuvent être envisagées simultanément : le texte de la démonstration assure la validité de l'énoncé et, en même temps, il en explique les raisons profondes.

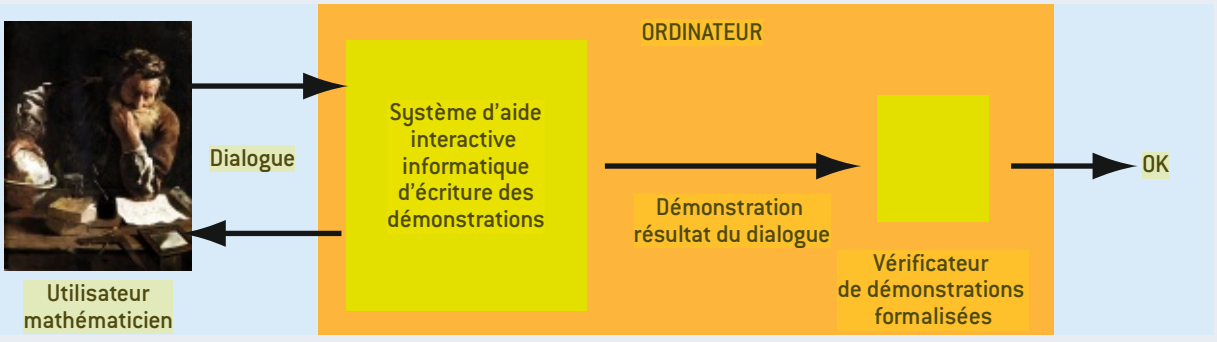
Lorsqu'on a affaire à des résultats vraiment difficiles, un double travail est nécessaire. La complexité dans le cas des mathématiques a donc pour conséquence la nécessité d'une part, de mener un contrôle formel (par l'utilisation des assistants de preuve), d'autre part de présenter un texte non formel illustré et largement commenté expliquant les idées mises en œuvre dans la démonstration.

C'est ce second texte qui permettra aux mathématiciens de saisir le sens du théorème… et d'imaginer le prochain théorème et la façon dont il faudra à son tour le démontrer.

4. Les assistants de preuve

Un logiciel assistant de preuve est destiné à aider un mathématicien à écrire des démonstrations formelles lors de cessions de travail interactives. Pour ce faire, le mathématicien traducteur de la démonstration classique en démonstration formelle doit avoir une parfaite connaissance de la preuve, ce qui n'est pas facile dans le cas d'une démonstration complexe. De plus, la traduction est difficile et lente : un ou plusieurs jours par page.

La partie du programme informatique qui gère les interactions est longue et délicate et, de ce fait, peut comporter des erreurs. C'est sans importance, car la preuve produite à l'issue du dialogue sera vérifiée par le cœur du système, qui lui seul doit être correct. Cette partie centrale est courte et vérifiée de façon indépendante par plusieurs spécialistes. On utilise parfois un autre logiciel assistant de preuve pour démontrer que ce cœur fonctionne correctement. La petite taille des parties centrales, et le soin particulier à les vérifier et à les prouver n'annulent pas le risque d'erreur finale (et donc de validation de preuves erronées), mais réduisent de plusieurs ordres de grandeur ce risque. Avec la formalisation des preuves, on améliore donc sensiblement la garantie de justesse de ce qui est démontré.



Cependant on a là, venue des mathématiques et de la logique, une technique quasi définitive permettant d'assurer l'absence de bogues dans les parties clefs d'un programme ou d'un microprocesseur. En 1993, la firme *Intel* avait mis en circulation une puce qui commettait des erreurs arithmétiques ennuyeuses ; elle utilise aujourd'hui la méthode des assistants de preuve pour éviter que cela se reproduise.

Récemment dans un article du *New York Times*, Adi Shamir (le *S* du système de cryptage RSA) faisait remarquer que la présence d'une seule erreur dans une puce informatique largement utilisée mettrait en danger la sécurité de millions de machines ; elle rendrait peut-être manipulables frauduleusement toutes les transactions faussement sécurisées du commerce électronique.

Non seulement la présence d'erreurs dans les logiciels et les puces peut faire tomber la fusée *Ariane*, ce qui s'est produit le 4 juin 1996, mais elle peut aussi être à l'origine de problèmes économiques majeurs. Les difficultés qu'on a en mathématiques à garantir l'exactitude des longues démonstrations et les méthodes élaborées pour résoudre ce qui ne semblait devoir n'intéresser qu'une petite communauté de chercheurs se révèlent importantes pour tout le monde. C'est là un exemple nouveau et spectaculaire de l'utilité des théories abstraites, ici la logique mathématique.

Quelle certitude ?

Certains voient un paradoxe dans l'idée d'utiliser l'informatique pour contrôler des démonstrations mathématiques, car notre expérience des ordinateurs montre que loin d'être des outils fiables, ce sont des sources nouvelles et incontrôlables d'erreurs dont on ne comprend souvent pas l'origine : on redémarre sa machine et ça s'arrange (parfois...) sans que personne ne sache pourquoi. Comment, dès lors, croire que les preuves formelles par ordinateur sont plus sûres que les preuves informelles et humaines ?

La réponse est que les assistants de preuve reposent sur des noyaux logiciels, la partie qui mène le contrôle des preuves produites lors des séances de travail interactif, aussi petits que possible et que l'on contrôle minutieusement :

– En s'arrangeant pour que leur présentation soit la plus lisible et la plus simple possible, et en demandant ensuite à un nombre aussi grand que possible de spécialistes de relire ces programmes clefs.

– En testant ces noyaux logiciels sur de très nombreux exemples : si une erreur persistait, elle aurait très probablement des conséquences et aurait produit des effets visibles lors des essais.

– En démontrant formellement que ces noyaux logiciels sont corrects à l'aide de versions différentes du même noyau ou, mieux encore, avec d'autres assistants de preuve écrits par des groupes de chercheurs indépendants.

Bien sûr, on n'arrive pas à une certitude totale que le noyau de l'assistant de preuve est sans erreur, mais à une très faible probabilité qu'il en contienne. Tout le risque se concentre sur un petit nombre de lignes de programmes (moins de 500 dans le cas de l'assistant *HOL*) et sur lui seul. Au total, on a considérablement gagné en sûreté et sans doute réduit le risque d'erreur de plusieurs ordres de grandeur. De même que les théorèmes les plus anciens des mathématiques ayant été redémontrés (sans utiliser d'assistant de preuve) un grand nombre de fois ne contiennent certainement aucune