tal pour les sociétés qui produisent des microprocesseurs ou dans le domaine de la fiabilité des programmes. Aujourd'hui, les industriels utilisent des logiciels dérivés de ces programmes de démonstration automatique. Le Centre national d'études spatiales, le Centre national d'études des télécommunications et l'Aérospatiale, en relation avec des universitaires et des chercheurs du CERT-ONERA de Toulouse, par exemple, mettent en œuvre de tels développements.

En intelligence artificielle, les logiciels systèmes experts qui tentent d'égaler les experts de domaines particuliers (médecine, détection et identification de pannes, conseil en placements boursiers, etc.) comportent des mécanismes de raisonnement appelés *moteurs d'inférences* qui proviennent en grande partie des recherches en démonstration automatique. De même, les langages d'interrogation des *bases de données* exploitent, pour répondre aux requêtes des utilisateurs, des mécanismes de calcul issus de la démonstration automatique.

Le langage de programmation *Prolog*, inventé à Marseille par Alain Colmerauer et Philippe Roussel en 1972, est très prisé pour le développement d'applications en intelligence artificielle. Il est conçu autour d'un mécanisme tiré d'un démonstrateur de théorèmes. La puissance de ce langage provient de cette origine et en explique la caractéristique principale : écrire un programme *Prolog* ne consiste pas à donner des ordres à l'ordinateur (principe de la programmation classique, dite *impérative)*, mais à décrire un problème que le langage se chargera de résoudre (principe de la programmation *déclarative)*.

De plus, en mathématiques, les démonstrateurs de théorèmes sont souvent les composants de base des vérificateurs de preuves.

VÉRIFIER SANS RÉPIT

Les erreurs en mathématiques sont fréquentes et, parfois, passent inaperçues de longues années. Dans le cas du théorème des quatre couleurs, par exemple, des mathématiciens ont publié à deux reprises des preuves incorrectes, et deux fois l'erreur est restée ignorée durant plusieurs années. Les exemples de démonstrations fausses sont innombrables en mathématiques. On peut penser que les parties centrales des mathématiques sont exemptes de telles erreurs, car leurs résultats ont été contrôlés par de nombreux mathématiciens, et ont été recoupés. En revanche, dans les domaines avancés de recherche où peu de mathématiciens travaillent, une erreur peut rester ignorée longtemps, voire indéfiniment si le résultat faux, trop en marge, n'est jamais utilisé. Les démonstrations fausses ou incomplètes de résultats justes, quoique moins ennuyeuses, sont aussi légion, et leur détection est encore plus délicate.

Il est naturel de vouloir être certain que les preuves mathématiques proposées par les mathématiciens sont bonnes. Cela pourrait être accompli en les soumettant au verdict impartial des machines. Cette idée a déjà conduit à la vérification de théorèmes mathématiques classiques.

Le projet AUTOMATH de N.G. de Bruijn, dès les années 1970, validait une partie du célèbre livre sur les fondements de l'analyse du mathématicien allemand Edmund Landau.

D'autres projets ont permis la vérification de preuves non triviales dont voici une liste très partielle :

– le théorème de Cantor sur le cardinal de l'ensemble des parties d'un ensemble (et qui indique qu'il y a plusieurs types d'infinis) ;

– le théorème des valeurs intermédiaires en analyse (qui dit qu'une fonction d'une variable continue qui prend les valeurs a et b prend aussi toutes les valeurs entre a et b) ;

– le théorème du point fixe de Banach (qui indique que, lorsque vous froissez une feuille de papier et la remettez dans le périmètre qu'elle occupait avant, alors, au moins un point est exactement au dessus de l'endroit qu'il occupait auparavant) ;

– le théorème de Wilson en arithmétique (qui stipule que p est un nombre premier, si et seulement si $(p-1)!+1$ est divisible par p) ;

– l'indécidabilité de l'arrêt d'un programme, démontrée en 1936 par Turing.

En France, le projet COQ propose ainsi un système puissant pour la formalisation et la vérification de démonstrations dans des domaines variés des mathématiques. Malgré ces succès obtenus et les incontestables progrès de ces dernières années, reconnaissons que peu de mathématiciens utilisent les programmes mis à leur disposition pour vérifier leurs démonstrations.

C'est que ceux-ci ne sont pas assez commodes, pour ce que le mathématicien ordinaire produit quotidiennement : les étapes de codage et de formalisation pour obtenir la vérification par l'ordinateur des résultats mathématiques nécessitent une patience et une familiarité qui, finalement, les réservent à quelques spécialistes. L'utilité de ces systèmes d'aide aux mathématiciens ne fait pourtant aucun doute, car ils permettraient de valider plus rapidement les preuves produites, d'en dériver des versions testables sur d'autres systèmes (vérifier une preuve plusieurs fois ne consisterait plus à la lire, mais à la faire passer dans des vérificateurs automatiques différents), et libérerait le mathématicien de certaines parties pénibles de son travail.

C'est pourquoi un projet international appelé QED (des trois mots *quod erat demonstrandum*, qui signifient «ce qu'il fallait démontrer») tente de se mettre en place depuis 1994. Son but est très ambitieux, et ses promoteurs le comparent à la rédaction du traité général de mathématiques de Bourbaki entreprise il y a quarante ans en France. Il s'agit de faire coopérer tous les spécialistes du domaine, de façon à accélérer les progrès.

La capacité à découvrir de nouvelles démonstrations non triviales et à vérifier celles produites par les mathématiciens a atteint un niveau intéressant ; d'après les tenants du projet, tous les mathématiciens l'utiliseront d'ici peu.

Avec le succès récent de *Deeper-Blue* sur Gary Kasparov, le 11 mai 1997, dans une compétition en bonne et due forme, et malgré les commentaires sceptiques qu'il a provoqués (voir *Pour la Science*, juin 1997), nous assistons à la naissance d'une intelligence artificielle aux capacités spécialisées, mais dont on ne peut plus dire qu'elle échoue dans tout ce qu'elle entreprend.

Jean-Paul DELAHAYE est directeur adjoint du Laboratoire d'informatique fondamentale de Lille du CNRS.
e-mail : delahaye@lifl.fr

W. MCCUNE et R. PADMANABHAN, *Automated Deduction in Equational Logic and Cubic Curves*, Lecture Notes in *Computer Science* n° 1095, Springer-Verlag, ISBN 3-540-61398-6, 1996.

C.W.H. LAM, *How Reliable Is a Computer-Based Proof?* in *The Mathematical Intelligencer*, vol. n° 12, pp.8-12, 1990.

M. RUSINOWITC, *Démonstration automatique. Techniques de Réécriture,* InterEditions, Paris, 1989

T. L. SAATY et P. C. KAITEN, *The Four-Color Problem, Assaults and Conquests,* Dover Publications, Inc. New York, 1977, 1986.

L. WOS, *The Automation of Reasoning : An Experimenter's Notebook with Otter Tutorial,* Academic Press, New York, 1996.