

et en a effectivement trouvé une que personne ne soupçonnait. Ici, les objets manipulés ne sont pas finis, puisqu'il s'agit de nombres réels ayant une infinité de décimales. Même si la méthode utilisée pour les conjectures sur les graphes ne s'applique pas, le procédé utilisé lui ressemble.

On se donne plusieurs nombres réels dont on connaît des évaluations aussi précises qu'on le veut ; c'est le cas pour π , et pour les sommes de séries dont le terme général est de la forme $1/[r^k(ak + b)]$. On recherche alors s'il existe une combinaison linéaire à coefficients entiers entre les nombres choisis qui soit nulle avec quelques décimales de précision. Quand une relation est trouvée, on teste si elle reste vraie avec un plus grand nombre de décimales. Si la relation est numériquement confirmée, l'algorithme propose la conjecture. En utilisant alors les méthodes classiques, le mathématicien reprend la main et tente de démontrer l'égalité.

C'est ainsi que la formule suivante fut découverte, puis rapidement démontrée :

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

L'intérêt de cette formule est qu'elle permet de calculer les chiffres binaires de π à un endroit donné du développement binaire, sans avoir à calculer les chiffres qui précèdent. Incontestablement, l'ordinateur avait découvert un résultat intéressant.

TROUVER DES DÉMONSTRATIONS

Un certain nombre de tentatives ont été faites pour introduire dans le domaine des mathématiques des méthodes de nature statistique et neuronale.

Kshitij Bansal et son équipe, au centre de recherche de Google à Mountain View, en Californie, ont développé un programme, intitulé *Holist*, pour démontrer des théorèmes en se fondant sur une méthode d'apprentissage. Partant du système d'aide à la formalisation et au contrôle des démonstrations *HOL-Light*, qui a été utilisé pour la mise au point de la démonstration de la conjecture de Kepler en 2014, les chercheurs ont programmé un outil automatique qui recherche des démonstrations en s'inspirant d'une base de démonstrations

82

LA MÉTHODE GRAFFITI

Le programme *Graffiti* énumère et sélectionne des conjectures portant sur les invariants des graphes.

- Le programme considère les invariants $inv_1, inv_2, \dots, inv_k$ et dispose pour chacun d'un sous-programme qui le calcule. Parmi les invariants possibles, il y a le nombre de nœuds du graphe, le nombre maximal de liens d'un nœud du graphe avec les autres nœuds, etc.
- Le programme dispose d'une base de test formée des graphes $G_1, G_2, \dots, G_n$. Cette base peut être complétée au fur et à mesure de l'utilisation du système pour le rendre plus efficace.
- Le programme dispose d'un système qui énumère les relations algébriques entre invariants. Ce système engendre, par exemple, $inv_1 + inv_2 \leq inv_3$, $inv_5 / inv_4 > (inv_1)^2$, etc. Le système donnera la priorité aux relations simples. La recherche de conjectures portant sur les invariants consiste alors à tester les relations algébriques énumérées par le système. Celui-ci élimine toute relation non vérifiée par l'un des graphes de la base de test.
- Les relations algébriques passant les tests sont soumises



Exemples de graphes. Le système *Graffiti* s'appuie sur une base de graphes pour tester des relations algébriques entre les « invariants » (nombre de nœuds, nombre moyen d'arêtes entre deux nœuds, etc.).

à une seconde série de tests pour déterminer si elles sont connues ou résultent de relations connues. Les relations non écartées sont proposées comme conjectures aux mathématiciens, qui tentent de les démontrer.