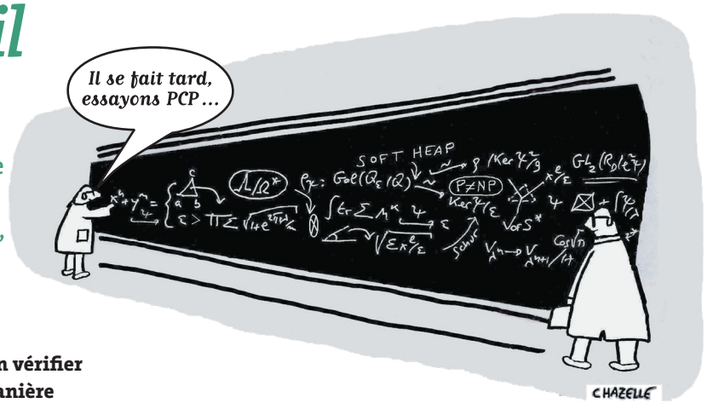
Vérifier une démonstration en un coup d'œil

Une méthode probabiliste de vérification de théorèmes permet de valider une démonstration en vérifiant seulement une petite partie. Irit Dinur, qui a produit une nouvelle démonstration de ce résultat, nous explique cette percée de l'informatique théorique.



© DR

Irit Dinur est professeur d'informatique à l'institut Weizmann, à Rehovot, en Israël. Ses travaux concernent les fondements de l'informatique et de la combinatoire, avec un intérêt particulier pour les preuves vérifiables de manière probabiliste et les algorithmes d'approximation.

Comment peut-on vérifier une preuve de manière probabiliste ?

I.D. D'habitude, pour vérifier une démonstration, il faut la lire entièrement. Mais grâce au théorème PCP (pour *Probabilistically Checkable Proof,* soit « preuve vérifiable de manière probabiliste »), on peut faire autrement. Dans un premier temps, il faut mettre la démonstration au format PCP à l'aide de graphes mathématiques. On aboutit à une preuve qui est une suite de bits. Ensuite, il suffit de vérifier au hasard quelques bits de cette démonstration. Si vous ne trouvez pas d'erreur dans ces quelques bits, vous êtes assuré que la démonstration initiale est juste avec une probabilité d'erreur de 1 sur un milliard par exemple. L'aspect le plus étonnant de cette méthode, c'est que la probabilité est indépendante de la longueur de la démonstration.

Quelle est l'astuce ?

I.D. Prenons une image. La preuve est comme un morceau de pain sur lequel il peut y avoir un peu de confiture localisée, qui représente les erreurs. Pour savoir s'il y a ou non de la confiture sur cette tartine, vous devez explorer toute la tartine, ce qui peut être long si la tartine est grande. Avec la méthode PCP, c'est inutile. C'est comme si vous étaliez d'abord votre confiture, s'il y a de la confiture. Puis, sur cette tartine (la nouvelle preuve), il vous suffit de regarder au hasard à quelques endroits. Si vous n'y trouvez pas de confiture, vous pouvez être assuré qu'il n'y en avait pas sur la tartine au départ, et donc que la preuve initiale est correcte. Le théorème PCP vous garantit qu'il existe toujours une autre tartine sur laquelle la confiture va s'étaler, c'est-à-dire transformer la preuve en une preuve où les éventuelles erreurs auront « diffusé » [1]. C'est la clé de la méthode : faire diffuser les erreurs dans toute la démonstration (voir encadré).

Quels types de problème peuvent être analysés par cette méthode probabiliste ?

I.D. La méthode PCP concerne les problèmes qui peuvent être vérifiés de manière efficace par un algorithme. Ce sont les problèmes de la classe NP (non déterministes polynomiaux). Intuitivement, cette classe de problèmes « raisonnables » : si je vous pose une question et que vous me donnez une réponse, il faut que je sois capable de vérifier que cette réponse est correcte. De plus, elle s'applique uniquement aux preuves formelles, c'est-à-dire à celles qui peuvent être mises sous forme compréhensible par un ordinateur. En pratique, c'est plus facile dans le cas de problèmes qui mettent en jeu des graphes et de la combinatoire. Ses domaines d'application privilégiés sont les problèmes d'approximation dans les équations linéaires, un domaine actif où il reste de nombreuses questions ouvertes.

Est-ce applicable à toutes les démonstrations ?

I.D. Uniquement aux preuves formelles. Dans d'autres domaines que les approximations linéaires, des formalisations de preuve ont été effectuées. Ainsi, Georges Gonthier, des laboratoires Microsoft, s'est attaqué à la démonstration par cette approche du théorème des 4 couleurs – le fait qu'il est possible de colorier une carte avec uniquement 4 couleurs sans que deux couleurs identiques se touchent. Il a déjà fait un énorme travail pour formaliser la preuve et la faire vérifier par un ordinateur, sans toutefois utiliser la méthode PCP [2]. Là encore, cette preuve ne fait intervenir que la combinatoire de graphes. Pour formaliser d'autres démonstrations, il faudrait formaliser des concepts non triviaux de topologie, d'algèbre et de géométrie différentielle par exemple. Cela reste hors de notre portée. ■

Propos recueillis par Philippe Pajot

[1] I. Dinur, *Journal of the ACM, 54,* n° 3, 12, 2007.

[2] G. Gonthier, *Notices of the American Mathematical Society, 55,* n° 11, 1382, 2008.

SIGNALEZ L'ERREUR

La méthode PCP est similaire par certains aspects aux codes correcteurs d'erreurs. Lorsqu'on envoie un message sur Internet, on ajoute des bits de contrôle qui permettent de repérer les erreurs. La transformation PCP est un codage qui ajoute des bits qui seront sensibles aux erreurs de la preuve. En pratique, la preuve est codée sous la forme d'un graphe mathématique (un réseau de nœuds reliés par des segments). Chaque nœud du graphe « demande » aux nœuds voisins s'il voit une erreur à son niveau. Le nœud rassemble ainsi toute l'information locale et la renvoie à ses voisins. Tout l'intérêt est là : même si l'information est locale – chaque nœud du graphe ne regarde que ce qui se passe autour de lui–, l'information se propage petit à petit. Un peu comme un commérage : chacun ne le dit qu'à son voisin, mais tout le monde finit par être au courant…

sur le web

http://bit.ly/chazelle

Une réflexion de Bernard Chazelle, de l'université de Princeton, sur le rôle des algorithmes en science (en anglais).

http://rjlipton.wordpress.com/

Un ton original sur la complexité algorithmique sur ce blog de Richard Lipton du Georgia Institute of Technology (en anglais).

Chronologie

Antiquité : Euclide produit le premier algorithme en donnant une méthode de calcul du PGCD de deux entiers.
1931 : Kurt Gödel énonce le théorème d'incomplétude. Il en découle que beaucoup de théorèmes ne peuvent être démontrés dans un système de logique formel.
1936 : Alan Turing invente le concept de calculateur universel.
1971 : Stephen Cook, de l'université de Toronto, et Leonid Levin, en URSS, formalisent la notion de complétude NP, essentielle en complexité algorithmique.
1990 : Laszlo Babai, de l'université de Chicago, Lance Fortnow, de la Northwestern University, et Carsten Lund, des laboratoires ATT, font le lien entre les preuves formelles standard et les preuves vérifiables de manière probabiliste.
1991-1992 : le théorème de vérification probabiliste de preuve (PCP) est démontré par une équipe internationale d'informaticiens.
2002 : Subash Khot, de l'université de New York, propose la conjecture des jeux uniques, qui renforce le théorème PCP. Elle reste à prouver.
2007 : Irit Dinur publie une preuve du théorème PCP utilisant la diffusion d'erreurs sur les graphes.