

discutées, car elles résultent d'une interprétation indirecte *a posteriori* des calculs d'AM.

Après ce premier essai, Douglas Lenat travailla sur un système nommé Eurisko, lui aussi destiné à produire des découvertes mathématiques. Son originalité est qu'il créait spontanément des règles heuristiques nouvelles à l'aide de métarègles du type : « Donner la priorité aux règles les plus simples ». Face aux difficultés, Douglas Lenat entreprit de concevoir et programmer un système dénommé Cyc, qui maîtrise des connaissances générales de tout humain, ce qu'il considérait comme indispensable pour aller plus loin en intelligence artificielle (IA). Pour lui, un système intelligent doit savoir que « si un humain marche, il ne reste pas à la même place » ou que « quand il pleut, les passants dans la rue sont mouillés », etc.

Le projet Cyc, démarré en 1983, a bénéficié depuis d'un travail équivalent à plus de mille hommes-années et se poursuit aujourd'hui encore. C'est le plus ambitieux et long projet d'IA jamais mené. Il a développé un savoir général et plusieurs savoirs particuliers liés à des sous-domaines, dont celui du terrorisme qui est probablement utilisé par l'agence américaine NSA (National Security Agency). Le système connaît plus d'un million de termes comme « la France », « le kilogramme », « New York », « les poissons », « la chasse »... et plus de 24 millions d'affirmations de base les concernant et les liant. Elles sont écrites dans un langage logique et disposent d'un système de déduction qui permet, à partir des affirmations de base, d'en produire de nouvelles.

79

QUELS TYPES DE RECHERCHES ?

L'intelligence artificielle en mathématiques a trois objectifs :

- L'aide à la démonstration et à la rédaction détaillée et contrôlée de démonstrations compliquées. Dans ce domaine, des succès ont été obtenus, comme la mise au point définitive de la démonstration de la conjecture de Kepler sur l'empilement le plus compact possible des sphères dans l'espace, démonstration due à Thomas Hales en 2014.
- La démonstration automatique : l'ordinateur recherche seul une démonstration. Quelques succès ont été obtenus.
- La recherche et la suggestion d'idées, de conjectures et de théorèmes nouveaux et intéressants. C'est ce dernier aspect qui est principalement envisagé dans cet article.

HR S'ATTAQUE AUX SUITES

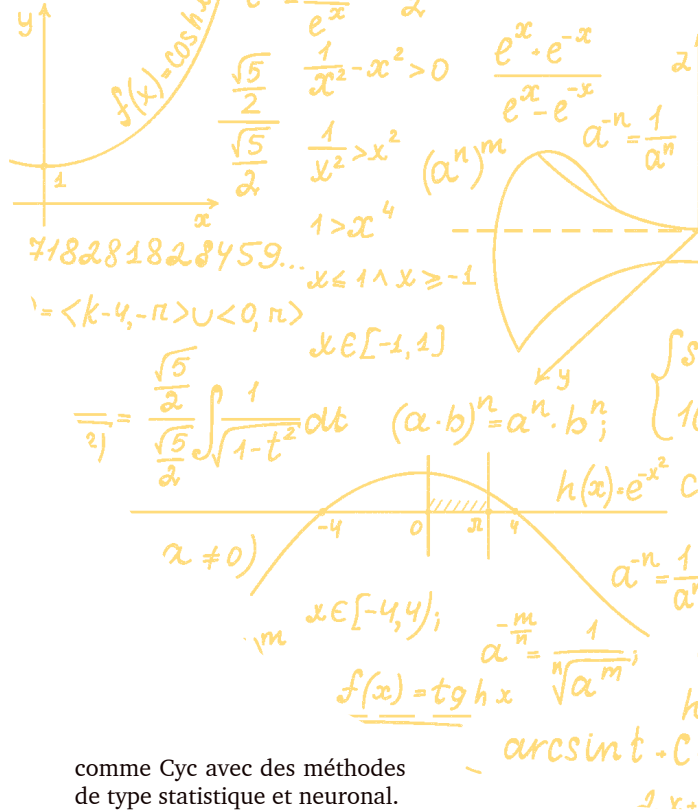
Cette recherche appartient à l'« IA symbolique » : les méthodes ne s'appuient pas sur des réseaux de neurones, ce sont des programmes manipulant des symboles, des mots, des concepts représentés par des éléments de programmes. Les humains qui les écrivent comprennent le détail de ce qu'ils y mettent et de ce qui se produit quand la machine fonctionne, contrairement aux réseaux de neurones de l'« apprentissage profond », où le système n'explique pas ses réponses. Le lien avec les mathématiques est devenu secondaire dans le projet Cyc, bien qu'un module ait été conçu pour aider les étudiants en mathématiques.

En tout cas, il semble certain que les progrès de l'IA devront associer la capacité de raisonnement et les connaissances générales de systèmes

FAIRE DES CONJECTURES INTÉRESSANTES

Pour concevoir des systèmes menant une recherche mathématique, il faut faire en sorte qu'ils effectuent des tris. Un tel système ne doit retenir que les propriétés et conjectures les plus intéressantes. Le Britannique Simon Colton, qui a développé le système *HR* (voir le texte principal), propose six critères pour repérer les conjectures intéressantes :

- 1) Les conjectures doivent être plausibles. Il faut éliminer celles qui sont fausses parce qu'il existe des exemples simples qui les contredisent.
- 2) Les conjectures doivent être nouvelles. Le système doit disposer des résultats connus sur le sujet étudié et savoir les exploiter pour ne pas proposer les conséquences immédiates de résultats connus.
- 3) Les résultats envisagés doivent être « étonnants », une notion qui échappe aux programmes. Le plus souvent, on se contente d'éliminer les énoncés qui sont des conséquences immédiates d'énoncés connus ou qui sont des cas particuliers de formules logiques toujours vraies, comme $\text{non}(\text{non-}A) \Leftrightarrow A$. Une propriété simple qui exige une longue démonstration est étonnante puisqu'on ne peut pas comprendre aisément ce qui la rend vraie. Malheureusement, la notion « exige une longue démonstration » est impossible à programmer en pratique, car, pour l'affirmer d'un énoncé, il faudrait explorer tous les chemins de démonstrations possibles pouvant y conduire.
- 4) Les conjectures doivent concerner de nombreux cas.
- 5) Les conjectures doivent être facilement compréhensibles et, donc, doivent avoir des énoncés assez courts et, si possible, ne faire intervenir que les concepts de base de la théorie étudiée.
- 6) Les conjectures doivent être fécondes et, par exemple, avoir beaucoup de conséquences simples. En mathématiques, il est difficile d'anticiper cette qualité d'une conjecture. Le grand théorème de Fermat n'est considéré comme fécond que parce qu'il a résisté pendant des siècles. Si Fermat l'avait démontré en quelques lignes, personne n'aurait estimé l'énoncé de ce théorème comme intéressant.



comme Cyc avec des méthodes de type statistique et neuronal.

Quelques années après Eurisko, Simon Colton, de l'université d'Édimbourg, a réussi à faire « imaginer » des notions mathématiques nouvelles à son programme, dénommé *HR* en l'honneur des mathématiciens Godfrey Hardy et Srinivasa Ramanujan, génie ayant découvert une multitude de formules extraordinaires.

Le programme *HR* s'est intéressé à l'arithmétique et a tenté à la fois de créer des concepts nouveaux – des définitions de suites numériques nouvelles –, de formuler des hypothèses à leur sujet et parfois de les démontrer à l'aide de programmes de démonstration automatique qu'il mettait à son service. Le programme disposait de critères pour ne se concentrer que sur des définitions potentiellement intéressantes. Il a pu ainsi identifier 14 suites numériques nouvelles, qui ont été acceptées en 2000 par l'Encyclopédie des suites numériques de Neil Sloane (<http://oeis.org>), où ne figurent que les suites ayant de l'intérêt aux yeux des mathématiciens.

L'une des suites proposées par *HR* est étonnamment simple et a conduit Simon Colton à publier un article dans le *Journal of Integer Sequences* en 1999. Il s'agit de la suite des « nombres refactorisables ». Un entier n est refactorisable s'il est divisible par le nombre de ses diviseurs. L'entier 9 est par exemple refactorisable, car il a 3 diviseurs (1, 3 et 9) et que justement il est divisible par 3. Plus généralement, si p est premier, p^{p-1} est refactorisable.

Notons que le programme de Simon Colton a découvert tout seul cette jolie propriété : « Si la somme des diviseurs d'un entier est un nombre premier, alors le nombre de diviseurs de ce