La preuve par ordinateur

Les ancêtres :

Leibnitz est sans doute le premier à avoir émis l'idée de la démonstration automatique.

Babbage dans "Calculating Engines" (1837) estime que si l'on utilise convenablement une machine à calculer mécanique, elle est plus fiable que l'homme ; il s'intéressait surtout à la confection des tables numériques.

Von Neumann qui s'est intéressé simultanément à la logique et à la conception des ordinateurs.

Shannon (un des fondateurs de la "théorie de l'information") a été le premier à concevoir des programmes pour le jeu d'échecs.

1. Découvrir une preuve ; en quoi l'ordinateur peut-il aider ?

a) La conception assistée :

Dans cette démarche, l'ordinateur est pourvu d'un programme qui permet au mathématicien un dialogue avec la machine sur un problème donné ; il fera donc une suite d'expériences qui peuvent éventuellement l'amener à découvrir des concepts, et des démarches utiles pour une démonstration.

L'exemple le plus fameux est la récente "démonstration" du théorème des quatre couleurs. La conception assistée a permis aux auteurs de dégager peu à peu les objets et les concepts clés, puis de las étudier exhaustivement.

b) L'heuristique (ou intelligence artificielle)

Il s'agit ici de tenter d'imiter les démarches humaines de démonstrations ou de résolutions des problèmes.

Exemples : 1) On a pu écrire des programmes permettant à un ordinateur de démontrer des théorèmes simples de géométrie (il s'agit de la géométrie des anciens programmes du secondaire) ; on en est resté à un stade très rudimentaire ; au mieux, la machine se révèle être un médiocre élève de seconde.

2) Confection d'emplois du temps : Les résultats sont peu concluant pour l'instant.

3) Jeu d'échecs : Les résultats sont plus impressionnants, puisqu'on trouve maintenant dans le commerce des robots-joueurs d'échecs, (dotés de plusieurs niveaux) qui sont déjà de très bon joueurs.

Dans tous ces exemples, la difficulté est de formaliser des démarches de tâtonnements.

2. Construire une preuve (programmation raisonnée); qu'est-ce qu'une preuve ?

2.1. Les ancêtres

Descartes, dans le discours de la Méthode; a été le premier à concevoir une "preuve" comme un enchaînement de pas élémentaire , dont chacun est suffisamment court pour emporter la conviction. La caractéristique des preuves mathématiques est qu'elle comporte en général un très grand nombre de ces pas élémentaires. D'où le conseil de Descartes : pour résoudre un problème, fractionner la difficulté ; le décomposer en sous-problèmes,...

Pascal, dans la logique de Port-Royal, a insisté sur la nécessité de l'emploi de définitions avec le conseil : dans l'étude d'un problème, remplacer le défini par sa définition.

Frege, à la fin du 19ème siècle, avec son idéographie, est le précurseur de la confection des organigrammes.

2.2. Conception formaliste moderne de la preuve (ou : la preuve idéale)

Ce qui est décrit ici est la conception de Gentzen.

a) Les ingrédients d'une preuve formelle :

Les séquents, ou assertions mathématiques

$\alpha \vdash \beta$ (si on accepte α , β est vraie)

$\vdash \alpha$ (α est vraie)

Les règles de déduction

par exemple le modus ponens $\dfrac{\vdash \alpha \ , \ \alpha \vdash \beta}{\vdash \beta}$

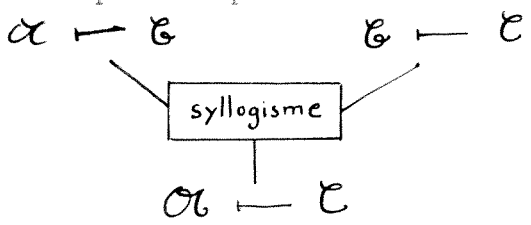
le syllogisme $\dfrac{\alpha \vdash \beta \ , \ \beta \vdash \gamma}{\alpha \vdash \gamma}$

(α implique β et β implique γ , donc α implique γ)

Un raisonnement est une accumulation de règles de déductions qui se termine par un séquent.

On peut ainsi représenter un raisonnement par un arbre, dont les ramifications sont des règles de déductions.

Ainsi le syllogisme se représente par :



b) Si l'on veut faire écrire des preuves par un ordinateur, il faut savoir coder les arbres précédents.

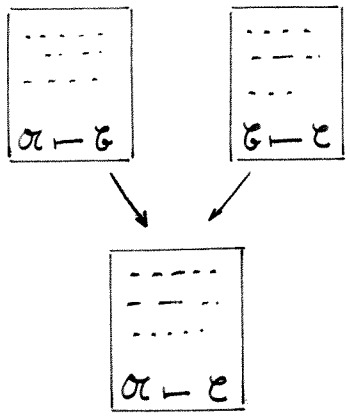
L'idée de base qui a permis ce codage est la référence à la théorie des catégories.

Dans ce qui précède, il y a deux sortes d'ingrédients :

les séquents, qu'on va considérer comme les objets d'une catégorie.

les démonstrations (ou raisonnements) qu'on va considérer comme les morphismes.

Alors une règle de déduction apparaît comme un opérateur sur les démonstrations. Par exemple, si l'on a deux démonstrations se terminant par les segments $\alpha \longrightarrow \beta$ et $\beta \longrightarrow \gamma$ le syllogisme permet de regrouper les deux pour constituer une démonstration se terminant par $\alpha \longrightarrow \gamma$; c'est donc un opérateur binaire sur les démonstrations.



Cette idée permet de considérer l'ensemble des démonstrations comme une algèbre , structurer par les règles de déduction. La théorie des monoïdes fournit alors un procédé de codage : en utilisant certains symboles de base, une démonstration peut être codée comme un mot formé avec ces symboles

c) L'écriture complète d'une preuve formelle requiert enfin des déclarations

Exemples : - Déclaration de variables : soit x un entier
soit x un réel ...

- Définitions : elles correspondent à un appel de sous-programme.

3. Vérifier une preuve

3.1. Aspect syntaxique

Il s'agit de savoir si une formule écrite avec le codage évoqué au paragraphe précédent représente bien une démonstration ; soit : se ramène-t-elle à un arbre dont les ramifications soient les règles de déduction ?

Cette démarche est analogue au travail du mathématicien qui vérifie un texte mathématique (sans nécessairement "comprendre" la démonstration). On a aujourd'hui de bons analyseurs syntaxiques par ordinateur.

3.2. Aspect sémantique

Il s'agit de savoir si un programme, par ailleurs syntaxiquement correct, réalise bien ce pourquoi il a été conçu (démontre effectivement ce que l'on attend).

Le problème est donc de s'assurer que la traduction d'une démarche exprimée avec la formulation mathématique usuelle dans le langage codé de la machine est fidèle.

On ne sait pas encore automatiser cette vérification ; elle se fait par des raisonnements mathématiques extérieurs.

4. Quel genre de preuves ?

4.1. Calcul numérique

Un calcul numérique (par exemple, le calcul des valeurs d'une fonction) est un exemple de démonstration.

4.2. Calcul Algébrico-arithmétique.

Ce sont les types de calculs numériques où l'on ne manipule que des nombres entiers (ou de rationnels), et où ne se posent pas les problèmes de précision.
Ce type de calcul permet par exemple des études sur les polynômes à coefficients entiers, sur les corps finis, etc...

4.3. Calcul "formel".

Ceci couvre essentiellement la partie calcul "formel" du calcul différenteil et intégral ; on peut ainsi écrire des programmes calculant des dérivées ou primitives des fonctions usuelles.

4.4. Construction d'objets algébriques.

On peut par exemple obtenir avec l'ordinateur des descriptions de groupes finis, par générateurs et relations. On a ainsi cherché à décrire certains groupes finis simples exceptionnels, ayant un nombre énorme d'éléments (le "monstre" de Fisher a environ 10^{90} éléments).

4.5. Algèbre "moderne".

Exemple d'un problème qu'étudie actuellement P. Cartier : La détermination, à l'ordinateur, du groupe de Galois d'une équation algébrique donnée. Cartier a pu, grâce à l'ordinateur, démontrer que l'équation $x^7 - 7x + 3 = 0$ a pour groupe de Galois le groupe simple à 168 éléments. C'est le premier exemple que l'on ait de ce type ; il était attendu depuis le mileiu du siècle dernier. On ignore encore si, étant donné un groupe fini G, il existe toujours une équation algébrique ayant G comme groupe de Galois.

5. Communiquer une preuve

Ce problème existe déjà dans la pratique humaine des mathématiques : lorsqu'on écrit une démonstration, jusqu'à quel point de détail ira-t-on pour emporter la conviction du lecteur ?

Avec l'ordinateur, c'est le problème entrée-sortie.
Si par exemple, on a écrit un programme qui fournit, pour un polynôme donné, la des-

cription de son groupe de Galois, il y a deux attitudes extrêmes :

1) Demander à l'ordinateur d'imprimer le détail de toutes les manipulations et calculs qui l'on conduit au résultat.

2) Lui demander seulement le résultat (la table de multiplication du groupe)

Dans le second cas, l'utilisateur n'aura aucune indication sur la procédure de démonstration. Dans le premier cas il aura des kilomètres de papier impossible à maîtriser.

On est donc amené à faire un compromis, et à demander la sortie d'informations intermédiaires qui pourront suffir à convaincre de la validité du résultat.

CONCLUSION : Il est probable que l'emploi de l'ordinateur dans la recherche mathématique se généralisera de plus en plus. Quel sera alors le degré de confiance que l'on pourra avoir en les résultats obtenus par ces démarches où interviennent l'homme et la machine ? La communauté devra se former une déontologie sur cette question. La mathématique deviendra sans doute de plus en plus semblable aux autres sciences expérimentales.

Pierre Cartier

d'après les notes prises par J. Martinet lors de la conférence prononcée par l'auteur au séminaire sur les fondements des sciences de l'U.L.P. le 11 janvier 1979.

Errata : Dans l'"Ouvert" nº 16, l'article de G. Glaeser "Pour une mathématique imaginative et joyeuse" s'est vu amputé de ses notes que l'on trouvera ci-dessous :

(*) - Le cours de mathématiques de Bezout peut être consulté à la bibliothèque universitaire de Strasbourg.

- Je ne suis pas arrivé à retrouver le manuel de Boisbertrand. Existe-t-il?

Par ailleurs, on voudra bien excuser toutes les fautes, lapsus et autres étourderies. Si un collègue veut bien accepter la tâche ingrate de relire les textes avant impression, il aura du travail immédiatement !