Tome 1, n° 3, Avril-Juin 1984

COMPTES RENDUS
DE L'ACADÉMIE DES SCIENCES

SÉRIE GÉNÉRALE

LA VIE DES SCIENCES

gauthier-villars

Informatique et mathématiques

Marcel SCHÜTZENBERGER
Correspondant de l'Académie

L'informatique théorique est une branche nouvelle des mathématiques appliquées dont l'ambition est de jouer un jour vis-à-vis des ordinateurs un rôle analogue à celui que joue la mécanique rationnelle vis-à-vis des machines.

Elle prolonge certains chapitres des mathématiques classiques qui semblaient bien loin de toute application à la vie courante. Peut-elle aider à préciser les limites de ce que peuvent faire les ordinateurs?

Le contenu de cet exposé ne correspond qu'à une fraction restreinte de ce que semble annoncer son titre. En effet, on ne traitera ici que d'informatique théorique et, encore, sous l'angle particulier de ses rapports avec certaines thèses philosophiques que la vogue des ordinateurs a contribué à diffuser dans tous les milieux.

Ce choix est délibéré : il faudrait une compétence que je n'ai pas pour prétendre évaluer même sommairement l'influence que l'informatique a pu avoir sur le développement des mathématiques dans les trente dernières années. Et puis même, un tel exposé risquerait d'être rebutant par la technicité qu'il exigerait.

Bornons-nous donc à rappeler que les ordinateurs sont nés des besoins des mathématiques appliquées et que ce sont bien évidemment les chapitres de l'analyse classique dont elles dépendent le plus qui ont bébéficié en retour de l'impulsion la plus manifeste. Mentionnons aussi que certains des chapitres les plus purs de la mathématique traditionnelle sont devenus l'objet de recherches quasiment industrielles : l'exemple type est la théorie des nombres premiers qui joue un rôle essentiel dans la cryptographie. Est-ce à cette cause que nous devons

La Vie des Sciences, *Comptes rendus*, série générale, tome 1, n° 3, p. 175-179

les théorèmes profonds grâce auxquels on pourra peut-être bientôt factoriser des entiers ayant près d'une centaine de chiffres ?

Une autre question qu'il vaut mieux laisser aux futurs historiens des sciences est la discussion de la part qui revient au mouvement propre des mathématiques ou à l'esprit du temps dans le développement de ce que l'on appelle la combinatoire.

Avant d'en venir à l'informatique théorique il nous faut rappeler le courant d'idées dans lequel elle est née et le lecteur voudra bien pardonner cette digression interdisciplinaire.

La réduction du monde à des schémas logiques et le thème de la naissance d'une complexité organisée par approximation successive et sélection à partir d'un chaos ont toujours fasciné les penseurs. Pour éviter toute querelle philosophique je désignerai cette thèse par le nom de *thèse cybernétique* puisque c'est celui de son dernier avatar avant sa forme ultimement démotique qu'est la théorie des systèmes. Avec l'avènement de l'électronique est apparue enfin comme une possibilité de mettre en œuvre la thèse cybernétique. Dès le début des années cinquante des équipes diverses s'y sont employées et les efforts principaux se sont portés dans deux directions désignées par des besoins sociaux : la traduction automatique des langues naturelles et la reconnaissance des formes. Comme vous le savez, ces besoins sociaux étaient surtout militaires et aucun crédit n'a manqué. Quelques centaines de millions de dollars, ont permis pendant vingt ans aux États-Unis, en U.R.S.S., en France, etc. à beaucoup de grands talents de se vouer à cette tâche en s'inspirant de la conception naïve de l'acte de traduction : chercher les mots dans un dictionnaire et appliquer les règles des grammaires pour passer mécaniquement d'une langue à une autre.

Devant la médiocrité des résultats, la commission Peirce, aux États-Unis a conclu en 1972 à l'arrêt de tout financement. De l'avis des experts rien n'a depuis fondamentalement modifié les bases techniques de cette décision, sinon que les progrès de la linguistique font mieux percevoir aujourd'hui toutes les difficultés réelles du problème.

La reconnaissance cybernétique des formes a été illustrée par un dispositif appelé « perceptron » qui a également suscité d'importantes recherches dans tous les pays. Laissez-moi rappeler que pour l'essentiel le perceptron n'était pas autre chose que la réalisation électronique du modèle pavlovien de l'apprentissage : un perceptron est un réseau d'élément logiques interconnectés de façon flexible que l'on « stimule » répétitivement par la forme à reconnaître dûment digitalisée. A chaque cycle, des connections sont automatiquement modifiées quand la « réponse » fournie (notez les guillemets) n'est pas la bonne. L'intuition cybernétique fondée sur la finitude de l'appareil laisse espérer qu'il finira par se stabiliser sur une configuration interne adéquate au but poursuivi. Un petit théorème amusant montre que sous des hypothèses extrêmement particulières il en sera ainsi. Malheureusement ce conditionnement qui réussit si bien avec les rats et les pigeons enfermés dans les boîtes de Skinner ne produit pas l'effet voulu sur les circuits électriques et un silence miséricordieux s'est abattu sur les perceptrons.

Ceci ne signifie pas bien sûr que des résultats impressionnants n'aient pas été obtenus dans la reconnaissance des formes. Simplement la démarche a été toute autre, commençant pour chaque domaine d'application par l'élaboration d'une théorie d'une classe bien définie d'objets à reconnaître et se terminant par la mise au point d'un dispositif préadapté à la fois à ces objets et aux moyens de calcul. C'est ce qu'avait bien vu mon maître et ami René de Possel dont le lecteur universel de caractères typographiques donnait, il y a dix ans, des

résultats voisins de ceux des machines qui commencent à apparaître sur le marché et qui ont d'ailleurs été conçues selon la même démarche.

Le sort du perceptron a été partagé par nombre d'autres tentatives pour mettre en pratique la thèse cybernétique, car bien d'autres systèmes « auto-adaptatifs » fondés sur la même philosophie ont été largement financés puis discrètement oubliés. L'exemple le plus curieux est la « *méthode delphique* » qui, à grand renfort de nombres aléatoires et de calcul linéaire, ne visait à rien moins qu'à automatiser le progrès technologique par un processus proprement darwinien de mutation au hasard et de sélection.

Notez bien que dans toutes ces tentatives on ne peut incriminer l'insuffisance du nombre des cycles. Car un ordinateur travaille au moins quelques millions de fois plus vite qu'un système nerveux, c'est-à-dire que pour lui une seconde est ce qui, pour un homme constituerait plusieurs mois de réflexion sans relâche. Bref, je l'affirme crûment : toutes les tentatives faites jusqu'ici pour appliquer la thèse cybernétique ont échoué.

Ce préalable achevé, nous revenons aux mathématiques et à leurs applications à l'informatique. Bien avant les ordinateurs, les logiciens et les mathématiciens avaient développé une théorie riche et profonde de ce qui est et de ce qui n'est pas calculable, ce dernier terme étant d'abord défini de façon rigoureuse ce qui n'est pas la moindre difficulté. A cette œuvre sont attachés les noms de Godel, de Post, de Church, de Turing et une partie majeure de l'informatique théorique est un prolongement de leurs travaux. Essayons d'illustrer en quelques mots la nature des notions de base.

On s'est aperçu très vite que beaucoup de situations pouvaient être ramenées au schéma suivant : S étant un ensemble donné d'entiers, on demande de vérifier si un entier arbitraire n appartient ou non à S. Par exemple si S est l'ensemble des nombres premiers, la question est répondue en vérifiant que n n'est pas divisible par un autre entier. A l'opposé, pour prendre un exemple simple, soient x et y deux matrices et S l'ensemble des entiers qui apparaissent comme entrée d'au moins une des matrices obtenues en faisant un produit (arbitrairement long) de x et de y. Un théorème de Markov montre que la question posée est indécidable, c'est-à-dire qu'il n'existe aucun algorithme permettant de répondre à la question « n est dans S ». Quelque soit la méthode employée, aucun calcul *fini* ne donne, dans le cas général la possibilité de conclure que n n'est pas, peut-être, une entrée d'un produit suffisamment long. La théorie des fonctions récursives a réussi à montrer l'indécidabilité de maints problèmes classiques mettant fin à l'espoir d'en trouver des algorithmes de résolution. La théorie de la complexité s'occupe de problèmes décidables mais essaye de fixer une borne supérieure $S(n)$ au nombre des opérations nécessaires pour déterminer si « $n \in S$ », ce qui fournit un moyen de classer les ensembles S par l'ordre de grandeur de $S(n)$. Elle est dominée par une conjecture concernant les S pour lesquels $S(n)$ croît plus vite que toute fonction polynomiale du nombre des chiffres de n.

La partie la plus concrète de ces recherches consiste, comme on s'en doute, à découvrir les meilleurs algorithmes pour accomplir ces tâches de routine que sont les produits de matrices, la transformation de Fourier ou, dans l'informatique de gestion, le tri et le réarrangement des listes.

A côté de ces recherches qui continuent d'autres plus traditionnelles, l'informatique théorique a suscité la naissance et le développement de nombreuses théories spéciales telle que celle des transformations de programme ou la sémantique formelle. Mais malgré leur importance pratique, elles sont trop techniques pour que l'on puisse ici faire autre chose que de

les évoquer. Nos collègues Arsac, Nolin et Nivat sauraient d'ailleurs bien mieux les présenter puisqu'ils en sont parmi les principaux artisans et on se limitera aux deux niveaux inférieurs de l'informatique parce que d'une part, ils suffisent pour discuter les difficultés de la thèse cybernétique et d'autre part, ils sont plus proches des mathématiques que de la logique.

Au premier niveau se trouvent les automates finis qui sont caractérisés par l'hypothèse que la quantité d'informations gardée en mémoire est *a priori* inexorablement bornée durant tout le calcul quelque soit le nombre n dont on veut vérifier l'appartenance à S. C'est une famille très générale et très élémentaire d'algorithmes à laquelle se résuisent la plupart des modèles cybernétiques que j'évoquais plus haut. Les intuitions qui motivent ces modèles et leur valeur explicative tirent précisément leur source de l'hypothèse de finitude. De plus on peut ramener à la théorie des automates finis maints résultats qualitatifs de comportement asymptotique dans des domaines tels que les chaînes de Markov en dimension finie ou certains modèles physiques régis par des équations élémentaires tels qu'on les rencontre en électronique ou en thermodynamique. Par contre, la plus modeste calculette programmable ne peut pas être envisagée de façon intéressante comme un automate fini : son usage normal suppose que les calculs restent en deçà des limites de la mémoire, ce qui implique que cette dernière soit tacitement considérée comme étant non bornée.

La théorie des automates finis a été fondée par Kleene en 1952 dans un mémoire consacré à un modèle cybernétique du système nerveux et a été presque aussitôt largement utilisée par les électroniciens. Vingt ans plus tard, S. Eilenberg en a donné une mise au point qui dominera pendant longtemps les nombreuses recherches qui continuent à se poursuivre. Ici encore, les questions majeures sont des questions de classification. Chaque automate fini se trouve être régi par un groupe fini et on peut préciser ce qui résulte de la combinaison de plusieurs automates en réseau par cascade et/ou mise en parallèle. Il découle de ces considérations qu'aucune complexité organisée ne peut apparaître qui ne soit pas comme donnée à l'avance par les groupes simples dont relèvent chacun des composants. La théorie mathématique des automates finis est une généralisation non commutative de celle des fonctions *rationnelles*. Le niveau supérieur, celui des « automates d'empilement », correspond de façon analogue à une généralisation des fonctions algébriques. Les algorithmes qui relèvent de ce niveau jouent un grand rôle en informatique puisqu'ils incluent la *compilation* c'est-à-dire le passage du programme écrit par l'utilisateur dans un langage humainement signifiant en une séquence de commandes élémentaires à l'usage de la machine. Dans le jargon informatique ont dit que les langages de programmation tels que *Fortran* ou *Algol* sont des « langages algébriques ». Le début de leur théorie remonte aux travaux d'un linguiste, N. Chomsky, qui, en 1955, avait cherché en partant des idées de Z. Harris à élucider le pourquoi de l'impossibilité d'approcher utilement les grammaires des langues naturelles avec des automates finis. Une des raisons en est devenue claire : les automates d'empilement relèvent du groupe libre de la même manière que les automates finis dépendent de groupes finis et il est hors de question d'utiliser les méthodes et les intuitions relatives à ces derniers pour effectuer la compilation d'un langage de programmation, donc, *a fortiori*, pour manipuler la syntaxe des langues naturelles. Bien sûr, ce raisonnement abstrait n'est pas indispensable pour expliquer et prévoir l'échec de la traduction automatique qui résulte plus naturellement des recherches proprement linguistiques telles que celles de M. Gross.

Ces exemples suffisent j'espère pour illustrer le propos de cet exposé qui était de montrer que la mathématique a gardé vis-à-vis de ces nouvelles machines que sont les ordinateurs les mêmes rapports que ceux qu'elle a traditionnellement entretenu avec le monde physique :

dans une direction elle fournit une aide précieuse au développement pratique et à l'enseignement des disciplines appliquées en construisant les structures abstraites et les modes de raisonnement efficaces sans lesquels il n'y aurait pas de progrès. Dans un autre sens, en retour, la mathématique tire de la réalité extérieure de nouvelles intuitions qui enrichissent son fond propre. Certes l'informatique théorique n'est qu'une très petite sœur de la mécanique rationnelle — mais de grâce, prenez en considération la différence d'âge.

BIBLIOGRAPHIE

H. DREYFUS, *What computers can't do,* Harper, 1979.
S. EILENBERG, *Automata Languages and Machines,* Academic Press, 1974.
M. GROSS et A. LENTIN, *Notions sur les grammaires formelles,* Gauthier-Villars, 1967.
S. L. JAKI, *Brain, Mind, and Computers,* Gateway Editions, 1969.
R. MCNAUGHTON, *Elementary Computability, formal languages and automata,* Prentice Hall, 1983.