

La démonstration automatique de théorèmes :

des méthodes et des résultats qui concernent l'enseignement des mathématiques

*par M. VIVET, Centre Universitaire du Mans
et J.P. LAURENT, Université de Caen.*

Ce papier concerne certains aspects de la démonstration automatique dans des domaines proches des mathématiques enseignées au niveau du premier cycle des universités. Une première partie insiste sur le besoin de trouver une bonne représentation du problème que l'on veut résoudre. La partie suivante montre que le cadre des systèmes logiques est généralement bien adapté et que la notion de graphe structurant un espace de recherche, notion effectivement utilisée en démonstration automatique, pourrait l'être aussi avec profit dans l'enseignement des mathématiques. La dernière partie fait le bilan des résultats et des techniques utilisées dans certains programmes. C'est une façon d'éclairer avec plus de précision ce qui précède. Globalement, le papier met l'accent sur deux types de tâches rencontrées en mathématiques : tâches de nature algorithmique et tâches de nature heuristique.

1. INTRODUCTION

Seul l'aspect pédagogique de la démonstration automatique est abordé ici. On ne parlera pas du tout de l'importance de la démonstration de théorèmes en intelligence artificielle dans des secteurs aussi variés que la conception de robots, l'analyse de

langues naturelles ou l'écriture automatique de programmes. Disons cependant que la démonstration automatique est un domaine de recherche appelé à jouer dans l'avenir un rôle fondamental dans l'utilisation des ordinateurs.

La résolution (méthode de J.A. Robinson) [13] ne sera pas évoquée ici. Cette méthode, quoique effectivement utilisée en démonstration automatique, est assez éloignée de notre façon de travailler et semble peu utilisable du point de vue pédagogique (Elle pourrait cependant être présentée dans le cadre d'une initiation à la logique). Nous nous intéressons surtout à un certain nombre de travaux dont le support de recherche a été choisi dans un domaine proche des mathématiques enseignées. Il était intéressant de faire le bilan de ces recherches de façon à se rendre compte globalement des résultats obtenus et à voir en quoi ce type de recherche peut contribuer à améliorer la manière d'enseigner les mathématiques.

2. PROBLEMES DES REPRESENTATIONS

Un premier choix important doit être opéré dès que l'on se propose de résoudre un problème : Comment le représenter ? Comment le formaliser ?

Généralement, une représentation est bonne si elle permet une manipulation souple, pratique, efficace des objets. Il faut qu'elle puisse permettre une perception maximum des choses essentielles. Le choix de la représentation est toujours fondamental et conditionne beaucoup la qualité (et souvent l'obtention) du résultat. La difficulté est que souvent nous n'avons une vision claire du problème qu'après l'avoir résolu. Ces idées sont très générales et, que ce soit l'ordinateur ou l'homme qui doit traiter le problème, il faut soigneusement choisir la représentation sur laquelle on va travailler. Pour certains objets, nous avons des représentations usuelles reconnues unanimement comme bonnes dans la plupart des cas. Par exemple, dans les calculs, les naturels sont toujours représentés relativement à une base donnée (la représentation par les chiffres romains est toujours écartée). Le choix de la base reste lui-même important. Les polynômes sont généralement ordonnés par puissances croissantes ou décroissantes de façon à avoir une représentation qui parle à l'esprit, qui nous permet une perception immédiate des faits importants (accès au degré par exemple). En géométrie, c'est pour avoir une perception globale

que l'on fait une figure ni trop grande ni trop petite et que l'on s'efforce de ne pas fabriquer un cas particulier. On peut aussi choisir de représenter le problème pour le traiter avec la géométrie analytique ou la géométrie descriptive. Il semble important que l'enseignant insiste sur ce problème des représentations en montrant qu'il ne s'agit pas seulement d'une convention pratique mais aussi d'une approche du problème fondamentale pour la suite. C'est peut-être une bonne façon d'amorcer la recherche de la solution et d'accélérer le processus qui fait que tout d'un coup un problème sort de son énoncé et devient clair.

Ce passage de la représentation du problème tel qu'il est posé à la représentation du problème sur laquelle on va travailler est très important. Faire prendre conscience de ce phénomène, habituer les élèves à travailler sur la représentation du problème avant d'aborder le problème lui-même, peut éviter de les voir s'embarquer dans la recherche de solutions à des problèmes mal posés pour lesquels une mauvaise représentation cache la voie du succès.

3. SYSTEMES LOGIQUES

Dans un certain nombre de cas, les problèmes sont représentés dans le modèle offert par les systèmes logiques. Nous disposons d'un jeu d'axiomes (énoncés vrais à priori) et de règles d'inférences qui nous permettent de déduire des énoncés vrais (théorèmes) à partir d'autres énoncés vrais.

Démontrer le théorème, c'est tracer le chemin logique qui part des énoncés vrais donnés au départ et aboutit à la conclusion en utilisant les règles d'inférence. Il peut être pratique pour schématiser l'obtention de la preuve d'utiliser une représentation par un graphe de l'espace de recherche. C'est une technique courante en démonstration automatique et qui peut être d'un grand intérêt didactique.

Exemple : Problème des groupes de Dickson.
Soit E un ensemble muni d'une opération \cdot associative.

Sachant que

$$\exists e \in E \quad \forall x \quad e \cdot x = x \quad (2)$$

et que

$$\forall x \in E \quad \exists \bar{x} \quad \bar{x} \cdot x = e \quad (3)$$

montrer que

$$\forall x \quad x \cdot e = x \quad \text{et} \quad x \cdot \bar{\bar{x}} = e$$

On peut faire apparaître la solution à l'aide du graphe suivant :

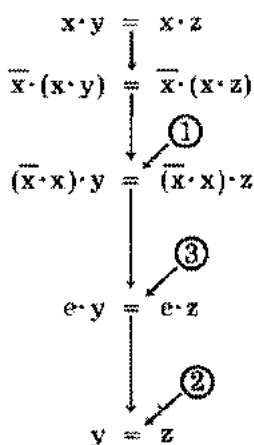


Fig. 1 : preuve du lemme L

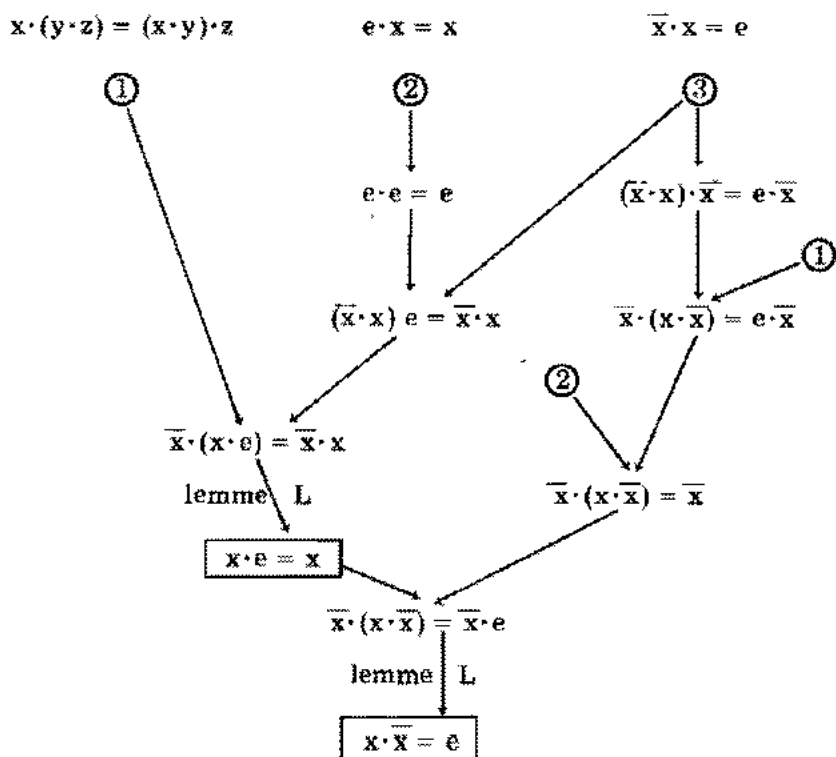


Fig. 2 : graphe montrant la preuve

On obtient ainsi un schéma qui parle à l'esprit, qui met en évidence les justifications de chaque résultat intermédiaire. La rédaction de certaines solutions sous cette forme est peut-être une façon de contraindre les élèves à justifier tout ce qu'ils écrivent. Une démonstration ainsi rédigée ne contient aucune littérature inutile (et se corrige très facilement). Elle ne contient que l'essentiel. A ce point, deux remarques s'imposent :

1. Le graphe peut devenir très grand. S'il n'est plus représentable sur le papier, la méthode est impraticable : les exercices à poser dans cet esprit doivent être bien choisis. Si l'on s'en tient à combiner sans discernement les énoncés vrais, il est fréquent, même avec des problèmes très simples, d'atteindre la limite de la capacité mémoire des ordinateurs.

2. Il y a manifestement deux types de tâches à effectuer :

α) Des manipulations : travaux de nature mécanique, algorithmique (par exemple : exécution de substitutions de termes à d'autres. On consultera avec intérêt l'algorithme d'unification [12] qui permet d'exécuter ce travail).

β) Des décisions : travaux plus subtils de nature heuristique (par exemple : choix des objets à manipuler, choix de la manipulation à effectuer, choix de l'endroit où l'on manipule dans l'objet...). Il est évident qu'à ces deux types d'activités très différentes doivent correspondre des méthodes pédagogiques différentes.

Tout ce qui est manipulation, mécanisme, doit être assimilé rapidement mais de façon définitive et parfaite (Pour l'informaticien, cela se traduit par des sous-programmes généraux qui doivent travailler vite et sans défaillance. Pour l'élève, c'est l'acquisition des mécanismes de calcul algébrique par exemple). En aucun cas cette partie ne doit être négligée.

Pour ce qui est de la partie heuristique, elle est beaucoup plus délicate à mettre en place en machine (ou à faire acquérir aux élèves). Les choix sont souvent nécessaires pour limiter l'espace et le temps de travail. Une méthode combinatoire qui essaierait toutes les combinaisons possibles d'axiomes et d'hypothèses pourrait être tentée. Elle n'est que très rarement possible, le graphe devenant trop volumineux (génération massive de noeuds n'apportant rien à la solution). Nous devons utiliser des heuristiques qui permettent d'éliminer des noeuds non prometteurs de succès. Toute la difficulté consiste à trouver des heuristiques assez

finies pour couper le maximum de noeuds inutiles tout en gardant tous ceux qui sont nécessaires à l'obtention du résultat. Les heuristiques servent également au niveau des choix des objets, des choix des opérateurs à appliquer à ces objets, etc... Une heuristique est bonne si d'une façon générale son utilisation fait progresser correctement le problème. Sa justification est d'ordre purement statistique.

Exemples

— heuristique générale souvent bonne en manipulation formelle : "Si une formule devient vraiment trop longue, essayer une autre voie"

— en calcul intégral : "Pour intégrer un produit, essayer l'intégration par parties"

— la complexité d'une formule est fonction croissante du nombre de connectives qui y figurent.

Le chercheur en démonstration automatique est amené à effectuer une formalisation des heuristiques afin de mieux comprendre comment on résout des problèmes. L'introspection ainsi réalisée doit permettre un meilleur enseignement : "les recettes" ne doivent pas être miraculeusement "bombardées" aux élèves mais justifiées statistiquement, puisque leur valeur dépend directement de la généralité et de la fréquence des succès qu'elles procurent.

4. BILAN DES RESULTATS ET DES TECHNIQUES DANS DES DOMAINES PROCHES DE L'ENSEIGNEMENT

4.1. Intégration

4.1.1. SAINT : (SLAGLE) 1961 [17]

Ce programme permet l'intégration formelle de fonctions usuelles (non compris les fractions rationnelles). Il permet de calculer des intégrales multiples lorsqu'il s'agit d'une extension triviale de l'intégrale simple. Le programme a effectivement calculé 86 primitives (moyenne : 2,4 minutes sur IBM 7090).

Exemples :

$$\int \frac{x^4}{(1-x^2)^{5/2}} dx \quad ; \quad \int (\sin x + \cos x)^2 dx$$

Il utilise trois modes de travail :

a) formes standards (20) permettant le calcul immédiat.

(exemple : $\int c^v dv = c^v / \text{Log } c$).

b) transformations algorithmiques : celles que l'on doit faire systématiquement lorsqu'elles sont possibles.

(exemple : $\int c f(x)dx = c \int f(x)dx$)

c) transformations heuristiques : transformations qui peuvent être ou ne pas être judicieuses.

Exemple : heuristique utilisée pour poser des changements de variable :

"Soit $g(v)$ une intégrande. Pour toute sous-expression $S(v)$ non constante, non linéaire, ayant sa première connective différente de $-$, qui n'est pas un produit avec un facteur constant et telle que le nombre de facteurs non constants de $g(v)/S'(v)$ est inférieur au nombre de facteurs de $g(v)$, essayer le changement de variable $U = S(v)$ ". Cette heuristique permet de poser $U = x^2$ dans

$\int x e^{x^2} dx$ (elle permet d'ailleurs également de poser $U = e^{x^2}$).

Le programme gère une arborescence ET — OU de buts. Cette arborescence est remise à jour dès qu'un résultat est obtenu afin de voir si ce dernier résultat achève la preuve. Une fonction permet la détection des noeuds les plus prometteurs.

4.1.2. SIN (Moses) 1971 [9] (Symbolic INtegrator)

Le programme travaille suivant trois modes de difficulté croissante.

α) Il cherche à voir si l'intégrande peut se mettre sous la forme

$$\int c \cdot op(u(x)) \cdot u'(x) dx$$

(c constante, op : fonction simple).

Pour un certain nombre de op, une table donne le résultat. A ce niveau, deux règles sont utilisées :

$$\int (A(x) + B(x)) dx = \int A(x) dx + \int B(x) dx$$

Développement des intégrandes à l'aide du binôme de Newton quand c'est possible.

β) Il permet d'utiliser 11 méthodes (choisies en fonction de l'apparition de connectives données dans l'intégrande). Ces méthodes permettent de faire subir des transformations à l'intégrande. Le problème transformé est alors renvoyé au mode α). Ici, une partie importante concerne l'intégration des fractions rationnelles.

γ) Ce mode concerne essentiellement la manipulation de l'intégration par parties et d'une heuristique issue de travaux de Liouville.

4.1.3. WANDERER (P.S. WANG) 1974 [20]

WANDERER est une partie du système MACSYMA conçu pour la manipulation symbolique. Il dispose ainsi de toutes les facilités offertes par le reste du système (en particulier manipulation des fractions rationnelles, résolution d'équations, factorisation d'expressions, différenciation, calcul de limites et programme d'intégration indéfinie). Aidé de cet important outillage de base, WANG a écrit un programme qui utilise des méthodes reliées directement à la théorie des résidus et à l'intégration sur des contours. Le contour est obtenu par examen de l'intégrande et des bornes d'intégration. Actuellement, WANG s'est intéressé à l'intégration de fractions rationnelles, de fonctions où interviennent certains types d'éléments irrationnels ($\sqrt{x^2 - a^2}$, x^k , ...) des lignes trigonométriques ou des logarithmes, exponentielles.

Exemples (entre crochets, temps de calcul sur PDP 10 - LISP) :

$$\int_{-\infty}^{+\infty} \frac{x^2 + Ax + B}{x^4 + 10x^2 + 9} dx = \frac{\pi B + 3\pi}{12} \quad [1978 \text{ ms}]$$

$$\int_0^{\infty} \frac{x^2 + Ax + B}{x^4 + 10x^2 + 9} dx = \frac{\pi B + 3\text{Log}(3) + 3\pi}{24} \quad [3582 \text{ ms}]$$

$$\int_0^{\infty} \frac{dx}{\sqrt{x}(x+1)} = \pi \quad [499 \text{ ms}]$$

$$\int_{-\infty}^{+\infty} \frac{x \sin x}{x^2 + 1} dx = \pi / E \quad [2635 \text{ ms}]$$

$$\int_0^{\infty} \frac{(\text{Log}(x))^2}{x^2 + 1} dx = \pi^3 / 8 \quad [1013 \text{ ms}]$$

L'obtention des pôles et le calcul des résidus se font à l'aide de l'outillage fourni par MACSYMA. Le choix du contour est de nature essentiellement heuristique.

4.2. Calcul de limites DALI (LAURENT) 1972 [7-8]

Le programme permet de déterminer la limite de fonctions numériques lorsque $x \rightarrow \alpha$.

Le programme manipule les développements limités et permet de lever les indéterminations. Le programme est composé de deux parties essentielles : l'une (POLLIM) essaie de trouver algorithmiquement la limite de l'expression qu'on lui fournit. L'autre (TRANSS) choisit heuristiquement des transformations sur l'expression ayant amené un échec de POLLIM. Ces deux parties conversent entre elles à l'aide d'un vecteur de renseignements (résultat d'une analyse de l'échec). Les transformations à faire subir à l'expression pour lever l'indétermination sont choisies à l'aide d'une évaluation calculée au moyen d'une matrice d'intérêt qui relie les propriétés des groupes de règles applicables et les propriétés de l'expression transmises par le vecteur de renseignements. Le programme a déterminé une centaine de limites choisies dans les manuels de mathématiques supérieures, proposées à l'oral des concours d'entrée aux grandes écoles.

Exemples :

$$\lim_{x \rightarrow \pi/4} (\operatorname{tg} x)^{\frac{1}{x - \pi/4}} = e^2$$

$$\lim_{x \rightarrow \infty} \frac{\operatorname{Log} x}{\operatorname{Log} \sqrt{1+x^2}} = 1 \quad , \quad \lim_{x \rightarrow \infty} e^{-x} e^{\sqrt{1+x^2}} = 1 \dots$$

4.3. Arithmétique. DALLARD (1974) [3]

Le programme reçoit en données des règles simplificatrices, des règles productrices et des axiomes. On lui fournit alors la conjecture à établir. La négation est rangée avec les axiomes. Le programme cherche en effet systématiquement à établir une preuve par l'absurde. Le but visé est, par implications successives, la génération d'un théorème qui, toutes simplifications faites, soit la constante FAUX (sans utiliser du tout le formalisme de la résolution, c'est la même idée).

Le programme travaille en trois phases :

1°) Dans la première, on essaie d'appliquer à tous les théorèmes toutes les règles de simplification qu'il est licite d'appliquer. C'est dans cette phase qu'on espère voir arriver la constante FAUX achevant la démonstration.

2°) Dans la deuxième phase, le programme tire parti des axiomes et théorèmes qu'il vient de simplifier et en déduit de nouvelles règles de réécritures simplifiantes ou productrices.

Par exemple :

— le théorème T peut donner naissance aux règles

$$T := \text{VRAI}$$

$$\text{NON } T := \text{FAUX}$$

— le théorème $A = B$ peut donner naissance aux règles

$$A := B \quad \text{et} \quad B := A$$

— le théorème $\forall x, y \ P(x, y)$ ou $Q(x, y)$ donnerait les règles

$$\text{NON } P(x, y) := Q(x, y)$$

$$\text{NON } Q(x, y) := P(x, y)$$

3^e) Dans la troisième phase, le programme utilisant des couples (théorèmes, réécriture productrice) non encore utilisés fabrique de nouveaux théorèmes. Le contrôle repasse alors à la première phase (c'est dans cette dernière phase que le programme, s'il ne peut fabriquer de nouveaux théorèmes, s'avouera vaincu).

Ce programme amène des idées intéressantes concernant la manipulation des quantificateurs et la façon d'écartier les problèmes liés à la présence d'opérations associatives et commutatives. Il a pu résoudre une cinquantaine de problèmes du niveau Terminale C.

Exemples :

- l'équation $x^2 + y^2 = z^2 + 1$ a des solutions avec $x \neq 1$ et $y \neq 1$
- $x^2 + 8z = 3 + 2y^2$ n'a pas de solution dans \mathbb{N}
- il existe une progression arithmétique de raison non nulle ne contenant aucun nombre triangulaire
- preuve par récurrence de 3^n divise $2^{(a^n)} + 1$
- si dans une division le quotient n'est pas nul, le dividende est supérieur au double du reste
- si a et b sont premiers entre eux, $a^2 \neq 2b^2$
- il existe des nombres premiers qui sont somme et différence de nombres premiers
- $3^{2n+1} + 2^{n+2}$ est divisible par 7, quel que soit n

4.4. Géométrie. GELERNTER (1959) [4-5]

Le programme écrit par Gelernter a résolu une cinquantaine de problèmes de géométrie plane. L'essentiel du matériel heuristique est obtenu en accompagnant les énoncés d'une figure (représentée sous forme analytique : chaque point étant donné par

son couple de coordonnées, des sous-programmes permettent de tester si deux segments sont égaux, parallèles, etc...). Cette figure n'est utilisée qu'en observation, les sous-programmes jouent le même rôle de reconnaissance que l'oeil. La preuve s'établit de façon formelle sur les énoncés. L'heuristique de base pour empêcher la prolifération d'énoncés stériles est la suivante : "Toutes les formules et buts intermédiaires qui sont engendrés à une étape donnée de la recherche de la preuve sont interprétés dans la figure. Si cette interprétation est valide dans la figure, alors l'objet généré est accepté comme étape possible de la preuve. Sinon il est rejeté". La figure joue ainsi le rôle d'un modèle sémantique permettant l'évaluation des énoncés formels.

Exemple de problème effectivement résolu :

Si dans un trapèze, on joint les milieux des diagonales, la droite obtenue coupe les côtés non parallèles du trapèze en leur milieu.

4.5. Dans des domaines encore proches des mathématiques enseignées, citons les travaux de Bledsoe [2] sur l'établissement des théorèmes sur les limites de fonctions numériques ; exemple :

$$\lim_{x \rightarrow x_0} f(x) \cdot g(x) = \lim_{x \rightarrow x_0} f(x) \cdot \lim_{x \rightarrow x_0} g(x)$$

Dans ses travaux Bledsoe utilise une heuristique qui est une mise en oeuvre du principe général suivant : "Pour établir une conclusion C de plusieurs hypothèses parmi lesquelles figure H, forcer H à contribuer au maximum à l'établissement de C et laisser un "reste" à établir à l'aide des seules autres hypothèses".

Les travaux de Madame GRANDBASTIEN [6] sur la résolution d'équations trigonométriques, de VIVET [19] sur la vérification d'identités à l'aide du raisonnement par récurrence, de SLAGLE [18] sur la manipulation d'inégalités, peuvent également être observés sous cet aspect pédagogique. Dans le domaine de la logique, les travaux de NEWELL [10] sont parmi les premiers en démonstration automatique et contiennent maintes idées reprises depuis. Ceux de PITRAT [12] ont montré l'intérêt de travailler au niveau des méta-théorèmes et méta-méta-théorèmes pour établir les théorèmes intéressants d'une théorie donnée par ses axiomes. L'heuristique de base est ici que "un théorème est intéressant s'il amène des théorèmes intéressants". Cette heuristique sert à gérer l'espace des théorèmes générés, en particulier à rejeter tous ceux qui ne présentent pas un intérêt suffisant.

Citons encore un travail original par la méthode qu'il utilise (en résolution de problèmes pour des robots) : SIKLOSSY [15-16] a écrit un système de deux programmes : l'un qui cherche à résoudre le problème posé, l'autre qui cherche à établir que le problème est impossible lorsque le premier a échoué. Ces deux programmes travaillent alternativement en s'échangeant un certain nombre d'informations sur la nature de l'échec.

CONCLUSION

La présentation de ces différents résultats avec une arrière-pensée didactique montre l'intérêt de la démonstration automatique et montre comment les recherches dans ce domaine peuvent apporter quelque chose à l'enseignement des mathématiques. L'introspection nécessaire pour dégager des heuristiques et formaliser des méthodes de travail amène une réflexion profonde sur la nature même de l'activité mathématique, et ne peut être que bénéfique à l'enseignement correspondant. Il semble en effet clair que si nous parvenons à faire en sorte que les machines fassent des mathématiques correctement, cela signifie que nous connaissons bien et maîtrisons bien un certain nombre de concepts réputés abusivement intuitifs. Les résultats déjà obtenus sont très encourageants.

REFERENCES

- [1] ARNOLD, A. "Les mathématiques à la portée de l'ordinateur", Dunod éd., Paris (1970).
- [2] BLEDSOE, W.W., BOYER, R.S., HENNEMAN, W.H. "Computer proofs of limit theorems", Second International Joint Conference on Artificial Intelligence, London, 1971.
- [3] DALLARD, R. "Présentation d'un programme de démonstration de théorèmes d'arithmétique", thèse de troisième cycle, Paris VI, avril 1974.
- [4] GELERTER, H. "Realization of a geometry-theorem proving machine", Computers and thought, (1963), p. 134-152.
- [5] GELERTER, H., HANSEN, J.R., LOVELAND, D.W. "Empirical explorations of the geometry-theorem proving machine", Computers and thought— Mac Graw Hill book company (1963), p. 153-163.

- [6] GRANDBASTIEN, M. "Un programme qui résout formellement des équations trigonométriques par des procédés heuristiques", thèse troisième cycle, Paris VI, juin 1974.
- [7] LAURENT, J.P. "Un programme qui calcule des limites en levant les indéterminations par des procédés heuristiques", thèse troisième cycle, Paris VI, juin 1972.
- [8] LAURENT, J.P. "A program that computes limits using heuristics to evaluate the indeterminate forms", *Artificial Intelligence Journal*, Vol. 4, N° 2, (1973).
- [9] MOSES, J. "Symbolic Integration: the stormy decade" *C. ACM*, August 1971, Vol. 14, N° 8, p. 548-560.
- [10] NEWELL, A., SHAW, J.C., SIMON, H.A. "Empirical explorations with the logic theory machine: a case study in heuristics", *Computers and thought*, (1963), p. 109-133.
- [11] NILSSON, N.J. "Artificial Intelligence", IFIP (Août 1974) Stockholm, p. 778-801.
- [12] PITRAT, J. "Un programme de démonstration de théorèmes" — Monographie d'informatique AFCET — Dunod (1970).
- [13] ROBINSON, J.A. "A machine oriented logic based on the resolution principle", *J. ACM*, Vol. 12, January 1965, p. 23-41.
- [14] SANDEWALL, E.J. "Heuristic search: concepts and methods", *Artificial Intelligence and heuristic programming*, University Press Edinburgh, 1971, p. 82-100.
- [15] SIKLOSSY, L., ROACH, J. "Proving the impossible is impossible is possible: disproofs based on hereditary partitions", Third International joint conference on artificial intelligence, Palo Alto, California, 1973.
- [16] SIKLOSSY, L., ROACH, J. "Collaborative problem solving between optimistic and pessimistic problem-solvers", IFIP, Stockholm, (Août 1974), p. 814-817.
- [17] SLAGLE, J.R. "A heuristic program that solves symbolic integration problems in Freshman calculus", *Computers and thought*, (1963), p. 191-203.
- [18] SLAGLE, J.R., NORTON, L.M. "Experiments with an automatic theorem prover having partial ordering Inference Rules", *C. ACM*, Vol. 16, N° 11, (novembre 1973).

- [19] VIVET, M. "Un programme qui vérifie des identités à l'aide du raisonnement par récurrence", thèse troisième cycle, Paris VI, novembre 1973.
- [20] WANG, P.S. "Symbolic evaluation of definite integrals by residue theory in MACSYMA", IFIP (Août 1974), Stockholm, p. 823-827.