A tout CRIN : de la naissance à la maturité (1963-1976)

Histoire et mémoire de l'informatique universitaire à Nancy, 14 janvier 2019

Claude Pair¹

Le mot « informatique » a été forgé en 1962 pour nommer une société de services, à un moment clé : le début du passage d'un outil – l'ordinateur – à une science qui sera un peu plus tard désignée en français par ce même nom. Pour moi, 1962 est l'année où je prends contact avec Jean Legras qui m'ouvre son Centre de Calcul et m'invite à suivre le cours de programmation qu'il y a confié à deux assistantes, Jacqueline Giannesini et Marion Créhange : Jacqueline enseigne le « langage-machine » de l'ordinateur IBM 650 et Marion son « code de programmation » [13] qui facilite quelque peu cet exercice.

Par rapport à un *outil* – un ordinateur programmé dans un dialecte qui lui est propre – une *science* fait preuve de généralité : à cette époque, il s'agit d'introduire des langages de programmation « évolués », plus proches du langage humain et indépendants de la machine utilisée. 1962 est ainsi l'année où apparaît FORTRAN IV, un langage destiné à tous les ordinateurs... pourvu qu'ils soient des IBM, une firme alors toute-puissante ! Mais depuis plusieurs années, un comité international travaille à définir ALGOL 60 [5] indépendant de tout constructeur et plus novateur que FORTRAN : un programme n'est plus une suite d'instructions, toutes au même niveau, mais structuré comme un texte en langue naturelle ; la récursivité apparaît à deux niveaux, pour accroître la capacité d'expression des programmes et pour décrire la syntaxe dans un métalangage dû à John Backus² et proche des grammaires du linguiste Chomsky [12].

ALGOL 60 sera fondamental dans la constitution d'une science informatique, par les notions qu'il introduit et les problèmes que pose sa « compilation », c'est-à-dire sa traduction dans le langage-machine de l'ordinateur dont on dispose. Je le découvre à travers une collection des Communications de l'ACM³ trouvée dans une armoire du Centre de Calcul, puis grâce à d'autres articles que je demande au centre de documentation du CNRS⁴. Sa définition finale est publiée en 1963 [39] et une rencontre se tient à Grenoble pour faire connaître sa version française ; Jean Legras en revient enthousiasmé.

Nous décidons donc de l'adopter comme support d'enseignement de la programmation, et d'en construire un « compilateur ». Jean Legras incite quelques étudiants à participer à l'aventure. Je me souviens de la réunion inaugurale à la rentrée 1963, où je trace les perspectives et organise une équipe de trois personnes pour construire ce compilateur, sur un ordinateur 1620 prêté par IBM dans l'attente qu'il puisse être acheté pour remplacer 650. Mais cette machine n'est accessible que certaines nuits et à Metz : ce n'est guère commode ! En 1965 pourtant, le compilateur est presque achevé, le groupe européen des utilisateurs de 1620 nous demande de le lui présenter à Mannheim [42], et l'université de Saint-Andrews (Ecosse) de le lui envoyer sur cartes perforées, l'outil de l'époque pour entrer des informations dans un ordinateur. Mais, patatras, plutôt que le 1620, le Centre de Calcul choisit, dans le cadre du Plan Calcul naissant, une machine française CAE 510 qui possède déjà un compilateur ALGOL. Notre motivation disparaît et IBM arrête son prêt, avant que le compilateur soit réellement utilisable par d'autres que nous.

¹ Voir [55, 15] et l'ouvrage de P-É Mounier-Kuhn sur l'émergence de la science informatique en France, PUPS.

² Auteur du langage FORTRAN, détaché par IBM auprès du Comité définissant Algol. Docteur honoris causa de l'université Henri Poincaré en 1989.

³ Association for Computing Machinery.

⁴ Ancêtre de l'INIST, aujourd'hui installé à Vandœuvre.

Mais l'expérience ne sera pas perdue. Des notions ont été dégagées, comme celle d'analyse syntaxique : pour traduire un texte, il faut d'abord en faire une analyse grammaticale, sous la forme d'un arbre ; en 1964, j'avais publié un article [41] introduisant la méthode d'analyse « ascendante » utilisée dans notre compilateur ALGOL⁵. Des étudiants ont été formés, en particulier à la théorie des langages, et cinq thèses de troisième cycle ont été soutenues [16, 67, 1, 6, 24, 27] dont trois sur ce compilateur. J'ai moi-même présenté fin 1965 une thèse d'Etat⁶ [43] qui récapitule les méthodes d'analyse syntaxique des langages de Chomsky. Deux sociétés de service nous ont demandé de les conseiller sur la compilation⁷. Surtout, une équipe de recherche en informatique a été créée, individualisée à l'intérieur du Centre de Calcul. La plupart des artisans de notre compilateur ont pris un emploi à l'extérieur, mais de jeunes étudiants la rejoignent : sa taille augmente rapidement – plus de 50 chercheurs en 1975 – grâce à la création de postes pour faire face à l'explosion et à la diversification de l'enseignement de l'informatique dans les facultés, écoles, et à l'IUT créé en 1966. En 1971, nous accueillerons un des quatre centres créés par le ministère de l'éducation nationale pour former des professeurs de lycée. Pendant longtemps, les chercheurs seront presque tous des enseignants et nous ne disposerons d'aucun technicien.

Cette pauvreté s'étend au matériel auquel nous avons accès, ce qui nous conduit à une recherche plutôt théorique. D'ailleurs, faire de l'informatique une science, c'est comprendre, formaliser, généraliser, ce que permet l'origine mathématique de la plupart de nos chercheurs. Mais nous nous refusons à faire des maths pour les maths. Depuis 1963, j'avais écrit plusieurs articles et communications dans le cadre de l'AFCALTI⁸. En les relisant aujourd'hui, je constate que leur caractère formel et pas toujours explicite les rend difficiles à lire et donc à utiliser pour fonder un programme. C'est encore vrai pour ma thèse, administrativement rangée dans la spécialité « mathématiques » où elle est la première en France traitant d'informatique, ce qui ne manque pas de susciter l'étonnement et le mépris de certains mathématiciens. L'année précédente, Jean-Claude Boussard avait soutenu à Grenoble une thèse d'Etat [8] sur la réalisation d'un compilateur ALGOL, qui, elle, avait dû être qualifiée de « science appliquée » en fonction de son contenu.

Il nous faudra un peu de temps pour inventer un style d'écriture à la fois rigoureux et moins uniquement mathématique, adapté à cette nouvelle science qu'est l'informatique [7] et à ses concepts fondamentaux : algorithmes [59] comme pour l'analyse syntaxique ; langages comme ALGOL, structures de données (ou d'information) [47, 50] comme les textes, les arbres, les piles (premier entré, dernier sorti). A partir de 1966, nos sujets de recherche – algorithmes de cheminement dans les graphes [44, 19, 20] dont les arbres sont un cas particulier, analyse syntaxique de figures à deux dimensions [38], puis langages d'arbres et définition des structures de données – dépassent la compilation, tout en réutilisant un certain nombre de notions qu'elle nous a fait rencontrer. En outre, nous voulons ouvrir la science informatique sur des disciplines variées (histoire [28], linguistique [45, 69, 48, 11], médecine [30], géologie [18]), comme vient de l'expliquer Marion Créhange. Nous nous construisons ainsi une place spécifique dans la recherche informatique française, moins appliquée qu'à Grenoble et Toulouse, moins théorique qu'à Paris. Nos relations nationales sont concrétisées

⁵ Cette méthode sera réinventée en 1966 par N. Wirth et H. Weber sous le nom d'analyse par précedence.

⁶ Au jury, présidé par Jean Legras, participaient Jacques Arzac (directeur de l'Institut de programmation à la faculté des sciences de Paris) et Jean-Claude Herz (IBM France et professeur associé à la faculté des sciences de Lille), ainsi que deux mathématiciens de Nancy.

⁷ pour PL/1 (lettre de la CERCI, 18/07/1966) et pour SNOBOL [31].

⁸ Association française de calcul et de traitement de l'information, qui a succédé à l'association française de calcul créée en 1956 et qui prend ensuite successivement les noms d'association française d'informatique et de recherche opérationnelle (AFIRO) et enfin de cybernétique économique et technique (AFCET, 1968-1994).

par l'école d'été d'informatique que nous avons fondée en 1971 sous l'égide de l'AFCEP, des liens étroits avec l'IRIA à partir de 1972, mon élection en 1973 comme président de la sous-section d'informatique créée au Comité consultatif des universités, et la même année notre reconnaissance par le CNRS comme équipe associée puis en 1976 comme laboratoire associé.

Nous obtenons aussi petit à petit une reconnaissance internationale, d'abord grâce à ma participation au groupe de travail de l'IFIP⁹ « Formalization of programming concepts », dès sa création en 1966. Elle me conduira à co-diriger le Ph. D. d'un étudiant de l'université de Pennsylvanie [2, 3] qui traduira en outre ma thèse en anglais. Alain Quéré et moi publions en 1968 dans la revue américaine *Information and Control* un article [57] qui introduit les « bilangages », langages d'arbres généralisant les langages linéaires et permettant de traiter de l'analyse syntaxique des langages de Chomsky [61, 52, 49]. Cet article recueille un écho international notable, en particulier une correspondance avec Rózsa Péter, célèbre logicienne hongroise (1905-1977) avec qui nous présentons deux communications croisées à une université hongroise et à l'Académie des Sciences de Budapest [60, 58]. Quant à l'école d'été, dès sa création elle attire des collègues Suisses et Espagnols, puis se tient dans divers pays francophones, en Europe, Afrique du nord, Amérique : Montréal en 1977.

A partir des bilangages, la voie est ouverte pour introduire et étudier des algèbres plus générales [32]. Elles conduiront à une formalisation des types de données comme systèmes formels à axiomes équationnels [65, 50, 51], dans le cadre de la première Action thématique du CNRS en 1972-74, proches de ce qui sera nommé plus tard « types abstraits algébriques ». Ceux-ci seront utilisés pour définir la sémantique des langages de programmation [34, 53, 54, 33, 10], une question qui nous fera revenir à la compilation¹⁰, ou plutôt la méta-compilation : décrire les compilateurs pour pouvoir les engendrer automatiquement [6, 37, 22].

Dans la même veine est proposée une formalisation de la sémantique d'Algol 68 [25], un nouveau langage [68] qui sera l'occasion de réinvestir beaucoup de nos réflexions. Comme son prédécesseur Algol 60, il fait avancer la science informatique. Le groupe Algol franco-belge, que nos collègues de Grenoble m'ont proposé d'animer, se fixe pour objectif de le faire connaître, notamment à travers plusieurs publications [9, 68, 4]. Un des progrès importants apporté par ce langage est que les types de données n'y sont plus prédéfinis comme en Algol 60 (nombres entiers, réels, valeurs booléennes, tableaux), mais peuvent être introduits en fonction des besoins. Le langage permet donc de définir, non seulement des programmes et des fonctions, mais aussi des types de données, et ceci de manière récursive. Trente ans après, j'ai eu la surprise de voir reconnue, grâce à Michel Sintzoff et Pierre Lescanne, mon antériorité sur la décidabilité de l'égalité de deux types récursifs, établie par un article [46] publié en 1970 à propos d'Algol 68.

Cependant, Algol 68 ne sera guère utilisé pour programmer. Ce n'était pas d'un langage à tout faire, donc complexe, qu'on ressentait alors le besoin pour améliorer l'efficacité de la programmation, mais plutôt d'une réflexion en profondeur sur cette activité. Le temps était venu de la programmation structurée, prônée dans les années 1972-76 par des leaders de la communauté informatique internationale [17]. A partir de 1973, nous introduisons et enseignons une voie différente : une méthode déclarative et modulaire, dite « déductive », qui, au lieu de décrire des calculs, définit des objets en partant du résultat à atteindre [56, 23]. L'idée est proche de ce qui sera nommé plus tard programmation fonctionnelle. En même

⁹ International Federation for Information Processing.

¹⁰ Ce sujet avait été effleuré dès 1964 à propos de la compilation [40].

temps est développé le projet Civa [21] adapté au développement de logiciels de taille importante, se référant à cinq bras d'un seul dieu pour cinq temps de la vie d'un programme.

Après 1968, la réorganisation universitaire créant à Nancy trois établissements d'enseignement supérieur, avec de l'informatique dans chacun d'eux, et le caractère « sans murs de notre labo, auraient pu mener à une dispersion des informaticiens. Leur volonté et l'existence d'un séminaire commun permettront de l'éviter. Au contraire, la transformation en 1971 du Centre de Calcul en service commun de ressources, le séparant de l'enseignement et de la recherche, conduit à l'indépendance du laboratoire, qui sera confortée par son association au CNRS. Il prend le nom de CRIN en 1975. Les thèmes de recherche se sont diversifiés. Jean-Paul Haton est venu du laboratoire d'automatique pour créer un pôle d'intelligence artificielle avec des chercheurs déjà présents. Deux thèses d'Etat ont été soutenues [21, 14] et d'autres s'annoncent [33, 26, 35, 66], permettant à leurs auteurs de diriger des recherches. Le directeur, élu président de l'Institut polytechnique et de la nouvelle section informatique-automatique du CNRS, est moins disponible. Il est temps de revoir l'organisation. Lorsqu'en 1976 le CRIN devient laboratoire associé, on y distingue cinq équipes : informatique théorique et théorie des langages ; outils et méthodes de développement de logiciels ; reconnaissance des formes et intelligence artificielle ; informatique d'organisation et bases de données ; informatique pour l'éducation et la formation [62, 63], en lien avec la formation des professeurs de lycée dirigée alors par Maryse Quéré.

Quelques années plus tard, en 1981, nommé directeur au Ministère de l'Education nationale, je passe le flambeau à Jean-Claude Derniame. Puis viendront Jean-Pierre Finance et Jean-Marie Pierrel, avant que le CRIN se transforme en LORIA en 1997 : il y a 20 ans, mais après 40 ans d'informatique universitaire à Nancy.

Références

1. André J. 1965. *Contribution à la construction d'un compilateur Algol pour IBM 1620*, Thèse 3^e cycle¹¹, Nancy.
2. Berger, J. 1977. *A study of inference for regular bilanguages*, Ph.D., University of Pennsylvania.
3. Berger, J. et Pair, C. 1978. Inference for Regular Bilanguages, *Journal of Computer and System Sciences* 16.
4. Bacchus, P., J. André, C. Pair (éds) 1975, *Manuel du langage algorithmique Algol 68*, Actualités scientifiques et industrielles, Hermann.
5. Backus, J. W. (éd.). 1960. Report on the algorithmic language Algol 60, *Numerische Mathematik*. 2 (Le comité de définition comprend un Français, B. Vauquois, professeur à Grenoble).
6. Bellegarde, F. 1972. *Face, langage d'écriture de compilateurs : définition et implémentation*, Thèse 3^e cycle, Nancy.
7. Borillo, M. et Pair, C. 1979, L'informatique une dimension nouvelle de la science contemporaine, in Mathieu, V. et Rossi, P. *La culture scientifique dans le monde contemporain*, Scientia, Milan.
8. Boussard, J. C. 1964. *Étude et réalisation d'un compilateur Algol 60 sur calculatrice du type IBM 7090/7094 et 7040/44*, Thèse d'État Science appliquée, Grenoble.
9. Boussard, J. C. et Duby J. J. (éds) 1971, Rapport d'évaluation Algol 68, *Revue française d'informatique et de recherche opérationnelle* R-1.
10. Broy, M., M. Wirsing, C. Pair. 1984. A systematic study of models of abstract data types, *Theoretical Computer Science* 33.
11. Carbonell, N. 1972. *Rôle des fonctions récursives primitives de ramifications dans la définition d'une langue naturelle : application à la syntaxe française*, Thèse 3^e cycle, Nancy.
12. Chomsky, N. 1959. On certain properties of grammars, *Information and Control* 2.
13. Créhange, M. 1961. *Structure du code de programmation*, Thèse 3^e cycle, Nancy.
14. Créhange, M. 1971. *Description formelle, représentation, interrogation des informations complexes : système Pivoines*, Thèse d'État, Nancy.
15. Créhange, M. et Haton M. C. 2014. L'informatique universitaire à Nancy : un demi-siècle de développement, *1024, Bulletin de la société informatique de France*, 3.
16. Cusey, M. 1964. *Construction d'un compilateur Algol pour IBM 1620*, Thèse 3^e cycle, Nancy.
17. Dahl O. J., E. W. Dijkstra, C. A. H. Hoare 1972. *Structured Programming*, Acad. Press.
18. De Bary, C. 1974. *Conception et réalisation d'un logiciel de gestion et d'interrogation d'une base de données géologiques*, Thèse 3^e cycle, Nancy.
19. Derniame, J. C. 1966. *Étude d'algorithmes pour les problèmes de cheminement dans les graphes finis*, Thèse 3^e cycle, Nancy.
20. Derniame, J. C. et Pair, C. 1971. *Problèmes de cheminement dans les graphes*, préface de Jean Kuntzmann, Monographies d'informatique, Dunod.
21. Derniame, J. C. 1974. *Le projet CIVA, un système de programmation modulaire*, Thèse d'État, Nancy.
22. Deschamp, P. 1980. *Production de compilateurs à partir d'une description sémantique des langages de programmation : le système Perluette*, Thèse docteur-ingénieur, Nancy.
23. Ducrin Amédée. 1984. (ouvrage collectif : M. Créhange, J. P. Finance, J. Guyard, N. Hertschuh, C. Pair, M. Quéré, J. Souquières), *Programmation*, tome 1, Dunod Informatique.

¹¹ A l'époque concernée, les universités françaises délivraient trois sortes de thèses : deux courtes, thèse de troisième cycle et thèse d'ingénieur pour les impétrants possédant ce titre ; une longue, la thèse d'État, aujourd'hui remplacée par l'habilitation à diriger des recherches.

24. Émond, A. 1965. *Application de la notion de pile à des problèmes portant sur les chemins des graphes*, Thèse 3^e cycle, Nancy.
25. Finance J. P. 1974. *Contribution à la formalisation de la sémantique d'un langage de programmation : application à Algol 68*, Thèse 3^e cycle, Nancy.
26. Finance J. P. 1979. *Étude de la construction des programmes : méthodes et langages de spécification et de réalisation de programmes*, Thèse d'État, Nancy.
27. Floc'h, A. 1966, *Achèvement d'un compilateur Algol, traitement des procédures*, Thèse 3^e cycle, Nancy.
28. Fossier, L. et Créhange, M. 1970. Un essai de traitement sur ordinateur des documents diplomatiques du Moyen Age, *Les Annales* 25.
29. Gaudel, M. C. 1980. *Génération et preuve de compilateurs basées sur une sémantique formelle des langages de programmation*, Thèse d'État, Nancy.
30. Germain, P. 1972. *Système de gestion et d'exploitation documentaire d'un corpus de dossiers médicaux*, Thèse 3^e cycle, Nancy.
31. Jaray, J. 1975. *Le langage SNOBOL4, ses applications, son implémentation*, Thèse 3^e cycle, Nancy.
32. Lescanne, P. 1971, *Étude de quelques théories des langages et généralisation du théorème de Kleene*, Thèse 3^e cycle, Nancy.
33. Lescanne, P. 1979, *Étude algébrique et relationnelle des types abstraits et de leurs représentations*, Thèse d'État, Nancy.
34. Livercy, 1978. (nom collectif : J.-P. Finance, M. Grandbastien, P. Lescanne, P. Marchand, R. Mohr, A. Quéré, J.-L. Rémy), préface de C. Pair, *Théorie des programmes : schémas, preuves, sémantique*, Dunod Informatique ; d'après un cours de l'école d'été francophone d'informatique, Tarbes, 1974.
35. Marchand, P. 1981. *Langages d'arbres, langages dans les algèbres libres*, Thèse d'État, Nancy.
36. Marin Navarro, J. 1979. *Un método de programación : presentación, implematación, transporte*, Universidad Complutense, Madrid.
37. Maroldt, J. 1972. *Définition de Face, langage pour l'écriture des compilateurs, implémentation d'un sous-ensemble*, Thèse docteur-ingénieur, Nancy.
38. Mohr, R. 1973. *Modèle algébrique pour l'analyse syntaxique de figures*, Thèse 3^e cycle, Nancy.
39. Naur, P. (éd.). 1963. Revised Report on the algorithmic language Algol 60, *Comm. ACM* .
40. Pair, C. 1964. Essai de description de la sémantique des langages de programmation, *Séminaire AFCALTI sur les langages symboliques* (animateur François Genuys).
41. Pair, C. 1964. Arbres, piles et compilation, *Revue française de traitement de l'information*.
42. Pair, C. 1965. Description d'un compilateur Algol, *European Region 1620 Users Group*, Mannheim.
43. Pair, C. 1965. *Étude de la notion de pile, application à l'analyse syntaxique*, Thèse d'État, Nancy.
44. Pair, C. 1966. Sur les algorithmes pour les problèmes de cheminement dans les graphes finis, *Théorie des graphes*, Centre international de Calcul, Rome, et Dunod, Gordon-Breach, 1967.
45. Pair, C. 1967. *La formalisation des grammaires*, Centre de Recherches et d'Applications Linguistiques, Faculté des lettres et sciences humaines, Nancy.
46. Pair, C. 1970. Concerning the Syntax of Algol 68, *Algol Bulletin* 31.
47. Pair, C. 1971 *Les structures de données et leur représentation en mémoire*, École d'été francophone d'informatique, publié avec la collaboration de M.-C. Gaudel, IRIA, 1977.

48. Pair, C. 1971. La formalisation des langages de programmation (introduction d'André Lentin), *Mathématiques et Sciences humaines* 34, École Pratique des Hautes Études.
49. Pair, C. 1971. Application de la théorie des ramifications au problème de l'équivalence structurale de deux C-grammaires, *Revue française d'informatique et de recherche opérationnelle* R-2.
50. Pair, C. 1974. *Formalization of the Notions of Data, Information and Information Structure*, IFIP Working Conference Data Base Management, North Holland.
51. Pair, C. (éd.) 1974. *Rapport de l'ATP Informatique 1972 : Informatique théorique, programmes et données*, Equipe associée 364, Université de Nancy II.
52. Pair, C. 1976. Les arbres en théorie des langages, *Colloque Les arbres en algèbre et en programmation*, Lille.
53. Pair, C. 1978. Some Theoretical Aspects of Program Construction, in F. L. Bauer and M. Broy (éds), *Program Construction*, International Summer School Marktoberdorf, Springer.
54. Pair, C. 1982. Abstract data types and algebraic semantics of programming languages, *Theoretical Computer Science* 18.
55. Pair, C. 1988. A tout CRIN : histoire d'un laboratoire, *Colloque sur l'Histoire de l'Informatique en France*, Grenoble. CRIN : The History of a Laboratory, *Annals of the History of Computing* 12 (1990).
56. Pair, C. et Maroldt, J. 1975. Introduction à une méthode de programmation déductive, in *L'enseignement de la programmation*, IRIA.
57. Pair, C. et Quéré, A. 1968. Définition et étude des bilangages réguliers, *Information and Control* 13.
58. Pair, C. et Quéré A. 1970. Sur les fonctions récursives primitives de ramifications, *Acta Mathematica Academiae Scientiarum Hungaricae* 21.
59. Pair, C., R. Mohr, R. Schott. 1988. *Construire les algorithmes*, Dunod Informatique.
60. Péter R. 1968. *Die Pairschen freien Binoïden als Spezialfälle des angeordneten freien holomorphen Mengen*, Eötvös Lorand Universität Budapest.
61. Quéré, A. 1969. *Étude des ramifications et des bilangages*, Thèse 3^e cycle, Nancy.
62. Quéré, M. 1975. Démarches algorithmique et déductive en enseignement assisté, rapports avec l'intelligence artificielle, *Colloque Analyse de la didactique des mathémat.*, Bordeaux.
63. Quéré, M. 1975. Modèle mathématique d'un système d'enseignement, *Computers in education*, North Holland.
64. Quéré, M. 1980. *Contribution à l'amélioration des processus d'enseignement, d'apprentissage et d'organisation de l'éducation : l'ordinateur outil et objet d'enseignement, application au projet SATIRE*, Thèse d'État, Nancy.
65. Rémy, J. L. 1974. *Structures d'information, formalisation des notions d'accès et de modification d'une donnée*, Thèse 3^e cycle, Nancy.
66. Rémy, J. L. 1982. *Étude des systèmes de réécriture conditionnels et application aux types abstraits algébriques*, Thèse d'État, Nancy.
67. Romac, R. 1964, *Étude des méthodes de tri*, Thèse 3^e cycle, Nancy.
68. Van Wijngaarden, A., B. J. Mailloux, J. E. L. Peck, C. H. A. Koster (éds) 1969. *Report on the Algorithmic Language Algol 68*, Springer. Traduit par le groupe franco-belge Algol (P. Arnal, J. Buffet, A. Quéré, éds), *Actualités scientifiques et industrielles*, Hermann, 1972.
69. Villard, J. 1969. *Emploi de PL/I pour les problèmes de linguistique*, Thèse 3^e cycle, Nancy.